Theoretical Computer Science 1 (1976) 237–267. © North-Holland Publishing Company

SOME SIMPLIFIED *NP*-COMPLETE GRAPH PROBLEMS

M. R. GAREY, D. S. JOHNSON

Bell Laboratories, Murray Hill, N. J., USA

and

L. STOCKMEYER*

Massachusetts Institute of Technology, Cambridge, Mass., USA

Communicated by A. Meyer
Received 1 September 1974

Abstract. It is widely believed that showing a problem to be *NP*-complete is tantamount to proving its computational intractability. In this paper we show that a number of *NP*-complete problems remain *NP*-complete even when their domains are substantially restricted. First we show the completeness of Simple Max Cut (Max Cut with edge weights restricted to value 1), and, as a corollary, the completeness of the Optimal Linear Arrangement problem. We then show that even if the domains of the Node Cover and Directed Hamiltonian Path problems are restricted to planar graphs, the two problems remain *NP*-complete, and that these and other graph problems remain *NP*-complete even when their domains are restricted to graphs with low node degrees. For Graph 3-Colorability, Node Cover, and Undirected Hamiltonian Circuit, we determine essentially the lowest possible upper bounds on node degree for which the problems remain *NP*-complete.

Introduction

Certain combinatorial problems, such as the traveling salesman problem and theorem proving in the propositional calculus, have long been notorious for their computational intractability, in that, despite the effort of many clever people, no algorithms have been found for them which can be guaranteed to require time bounded by a polynomial in the length of the input. The belief in the inherent difficulty of these problems has been strengthened by results of Cook and Karp [3, 13]. These show that simple forms of the above problems, together with a wide variety of other combinatorial problems, form a class, the *NP-complete*[1] problems, no member of which

* The work of this author was done, in part, at Project MAC, and supported by the National Science Foundation under research grant GJ-34671; present address: IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y.

[1] "Polynomial complete" problems, in the terminology of Karp [13].

is known to have a polynomial time algorithm, but such that if any one of the problems does have such an algorithm, then they all do.

These results have stimulated many researchers to examine other combinatorial problems for which no polynomial time algorithms are known, to determine whether they too are *NP*-complete, and their efforts have resulted in the discovery of additional members of this class [15, 17, 19]. Such results have considerable practical significance. If one knows that the problem he wishes to solve is *NP*-complete, and thus is unlikely to have any polynomial time algorithm, he may feel justified in concentrating on more hopeful alternative approaches.

He can look for algorithms which, although admittedly exponential in the worst case, seem to work quickly on most practical problems (e.g., the simplex method), or even which are just "less exponential" than previous algorithms, and hence extend somewhat the maximum size problem which can be solved within practical time limits [16]. Alternatively, he can look for fast algorithms which, although they do not actually find optimal solutions for the problem, are *guaranteed* to yield solutions which are "close" to optimal [6, 9, 11, 12].

An important motivation for this paper is that in many real-world applications the standard problem does not occur with full generality, but rather in a restricted form, due to additional constraints imposed on the input domain by the practical situation at hand. In some cases, such constraints may make the problem more amenable to efficient algorithmic solution, whereas, in other cases, the restricted problem may be essentially as difficult to solve as the original problem. In this paper we examine certain natural restrictions on the domains of a number of known *NP*-complete problems, to determine whether the resultant subproblems are still *NP*-complete, or if they do have polynomial time algorithms.

Our results show that many known *NP*-complete problems remain *NP*-complete, even when their domains are substantially restricted. In addition to the immediate significance of knowing that these restricted problems are *NP*-complete, the nature of the restrictions makes the completeness results useful in two other ways. First, they increase our knowledge of the essential elements which made the *original* problems *NP*-complete. Second, they give us valuable tools for proving other completeness results. For instance, by observing that Satisfiability With At Most 3 Literals Per Clause, a restricted form of Satisfiability, is still *NP*-complete, Karp [13] was able to derive the *NP*-completeness of Chromatic Number, Exact Cover, Max Cut, and a number of other problems.

In the first half of this paper, we show that an important restricted version of Max Cut is still *NP*-complete, and from that derive the completeness of the Optimal Linear Arrangement problem, as well as a number of more closely related problems. The second half of the paper considers the effect of restricting the allowable types of graphs for *NP*-complete graph problems such as Node Cover, Chromatic Number, and Hamiltonian Circuit, by either restricting the maximum degree of the nodes, or allowing only planar graphs, or both.

We summarize here the basic definitions, referring the reader to [13] for a more complete discussion. Let $B = \{0, 1\}$ and let B^* denote the set of all finite strings of elements from B. Any subset L of B^* is called a *language*. Let π be the class of functions $F: B^* \to B^*$ which are computable in polynomial time by one-tape deterministic Turing machines. If L and M are languages, we say that L is *polynomiael reducible* to M, written $L \alpha M$, when there is a function $f \in \pi$ such that $f(x) \in M$ if and only if $x \in L$. M is *NP*-complete if $M \in NP$ (the class of languages recognizable in polynomial time by one-tape *non*deterministic Turing machines) and *every* language in *NP* is polynomial reducible to M. In fact, if L is *NP*-complete and $L \alpha M$, then $M \in NP$ implies that M is *NP*-complete.

In accord with the above definitions, the "problems" we shall consider in this paper, although many are more naturally thought of as optimization problems, shall be presented as recognition problems (with the straightforward details of the encoding of entities such as graphs and integers into strings of 0's and 1's omitted). Our proofs can then consist of showing that known *NP*-complete languages reduce to the ones we are considering. (A list of the known *NP*-complete languages we shall use, together with their definitions, is given in the Appendix). In general, we leave to the reader the straightforward verification that (a) the language is in *NP* and (b) the described mapping can be performed in polynomial time.

1. Simple Max Cut And Related Problems

In [13], the following problem was shown to be *NP*-complete:

Max Cut

Input: Graph[2] $G = (N, A)$, weighting function $w: A \to \mathbf{Z}$ (the non-negative integers), positive integer W.

Property: There is a set $S \subseteq N$ such that

$$\sum_{\substack{\{u,v\} \in A \\ u \in S,\, v \in N-S}} w(\{u, v\}) \geqslant W.$$

Karp proved the *NP*-completeness of this problem by a reduction from the Partition problem. Thus, his proof relies on the fact that the edge weights can be represented in space proportional to the logarithm of their magnitudes, since there is a dynamic programming algorithm for Partition which runs in time polynomial in the input length, if those inputs are expressed in unary (i.e., in length proportional to their magnitudes).

One might conjecture, therefore, that if we restricted Max Cut by requiring each edge weight to be exactly 1, the new problem, which we call Simple Max Cut, might

[2] We use the ordered pair (N, A) to denote a graph G having node set N and edge set A.

become easy. As added support for this view, notice that if $W = |A|$, then this problem simply asks whether G is bipartite, which can be determined quite easily.

In fact, however, Simple Max Cut is NP-complete, as we show using a two-step reduction from Satisfiability With At Most 3 Literals Per Clause (Sat3 — for formal definition, see the Appendix). We first consider the following restricted version of the Maximum Satisfiability problem of [12]:

Maximum Satisfiability With At Most 2 Literals Per Clause

Input: Disjunctive clauses $C_1, C_2, \ldots, C_p$, each containing at most two literals, positive integer k.

Property: There is a truth assignment to the variables which satisfies k or more clauses.

We use the abbreviation Max Sat2 to denote this problem. Observe that, when $k = p$, this problem can be solved in polynomial time [3]. However, we now show that Sat3 can be reduced to Max Sat2, proving that Max Sat2 is NP-complete.

Theorem 1.1. *Sat*3 α *Max Sat*2.

Proof. Suppose we are given an input for Sat3, that is, a set S of disjunctive clauses, each containing at most 3 literals. If any clause has fewer than 3 literals, we may replace it by an equivalent clause which has exactly 3 literals, merely by repeating one of the literals which it contains. Hence, we may assume that each clause in S contains exactly 3 literals, and we label them $(a_1 \vee b_1 \vee c_1)$ through $(a_m \vee b_m \vee c_m)$, where each a_i, b_i, and c_i represents either a variable or its negation. The corresponding set S' of clauses and value k for Max Sat2 are given by:

$$S' = \bigcup_{i=1}^{m} \{(a_i), (b_i), (c_i), (d_i), (\bar{a}_i \vee \bar{b}_i), (\bar{a}_i \vee \bar{c}_i), (\bar{b}_i \vee \bar{c}_i),$$
$$(a_i \vee \bar{d}_i), (b_i \vee \bar{d}_i), (c_i \vee \bar{d}_i)\},$$
$$k = 7m.$$

$7m$ or more of the clauses in S' can be satisfied simultaneously if and only if the original set S is satisfiable. For note that, if we have any satisfying assignment for S, then either one, two, or three of a_i, b_i, c_i must be set "true" for each i. The reader may verify that, in all three cases, there is a truth setting for d_i causing precisely seven of the clauses in S' arising from clause i to be satisfied. Furthermore, no setting of d_i will permit more than seven of the ten clauses to be satisfied, and at most six of the clauses can be satisfied if all of a_i, b_i, and c_i are "false".

We now prove the completeness of Simple Max Cut by reducing Max Sat2 to it.

Theorem 1.2. *Max Sat2* α *Simple Max Cut.*

Proof. Let clauses $C_1, C_2, ..., C_p$ and integer k be given as input for Max Sat2. In analogy with the proof of Theorem 1.1, we may assume that each clause contains exactly two literals, not necessarily distinct, and label them as $(a_1 \vee b_1), (a_2 \vee b_2), ..., (a_p \vee b_p)$. Furthermore, we may assume that no two clauses are identical since, given not necessarily distinct clauses $C_1', C_2', ..., C_q'$ and integer k', an equivalent problem with all clauses distinct is obtained by replacing each clause $C_i' = (u_i \vee v_i)$ with the two clauses $(u_i \vee c_i)$ and $(v_i \vee \bar{c}_i)$ (where c_i is a new variable) and setting integer $k = k' + q$.

Corresponding to this input for Max Sat2, we shall construct a graph as input to Simple Max Cut in two steps, first giving the nodes and a basic framework of edges, and then adding in some additional problem-specific edges. Let $x_1, x_2, ..., x_n$ be the variables occurring (either complemented or uncomplemented) in the p clauses. The set N of nodes for the graph G is

$$\begin{aligned} N = &\{T_i: 0 \leqslant i \leqslant 3p\} \cup \{F_i: 0 \leqslant i \leqslant 3p\} \\ &\cup \{t_{ij}: 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant 3p\} \\ &\cup \{f_{ij}: 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant 3p\} \\ &\cup \{x_i: 1 \leqslant i \leqslant n\} \cup \{\bar{x}_i: 1 \leqslant i \leqslant n\}. \end{aligned}$$

The basic framework A_1 of edges is

$$\begin{aligned} A_1 = &\{\{T_i, F_j\}: 0 \leqslant i \leqslant 3p, 0 \leqslant j \leqslant 3p\} \\ &\cup \{\{t_{ij}, f_{ij}\}: 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant 3p\} \\ &\cup \{\{x_i, f_{ij}\}: 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant 3p\} \\ &\cup \{\{\bar{x}_i, t_{ij}\}: 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant 3p\}. \end{aligned}$$

For any given partition $N = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$, we will say that edge $\{u, v\}$ is "bad" if both u and v belong to the same set in the partition and is "good" otherwise. Notice that all edges in A_1 will be good for any partition $N = S_1 \cup S_2$ which obeys (a) all T_i belong to the same set in the partition and all F_i belong to the other set, and (b) for each i, x_i and all t_{ij} belong to the same set in the partition and $\bar{x}_i$ and all f_{ij} belong to the other set. Furthermore, if any pair F_i, F_j belong to different sets in the partition, then at least $3p+1$ edges from A_1 will be bad, since each such pair of nodes are mutually adjacent to $3p+1$ other nodes. Similarly, if any pair x_i, $\bar{x}_i$ belong to the same set in the partition, then at least $3p+1$ edges from A_1 will be bad, since there are $3p+1$ disjoint 3-edge paths between x_i and $\bar{x}_i$.

The following additional edges are included in G:

$$\begin{aligned} A_2 = &\{\{a_i, b_i\}: 1 \leqslant i \leqslant p \text{ and } a_i \neq b_i\} \\ &\cup \{\{a_i, F_{2i-1}\}: 1 \leqslant i \leqslant p\} \cup \{\{b_i, F_{2i}\}: 1 \leqslant i \leqslant p\}. \end{aligned}$$

The input for Simple Max Cut is the graph $G = (N, A_1 \cup A_2)$ and $W = |A_1| + 2k$.

Given a truth assignment for the n variables which satisfies k or more clauses, construct the partition $N = S_1 \cup S_2$ as follows:

$$S_1 = \{F_i : 0 \leqslant i \leqslant 3p\} \cup \{x_i : x_i \text{ is false}, 1 \leqslant i \leqslant n\}$$
$$\cup \{t_{ij} : x_i \text{ is false}, 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant 3p\}$$
$$\cup \{\bar{x}_i : x_i \text{ is true}, 1 \leqslant i \leqslant n\}$$
$$\cup \{f_{ij} : x_i \text{ is true}, 1 \leqslant i \leqslant n, 0 \leqslant j \leqslant 3p\},$$
$$S_2 = N - S_1.$$

Since, for each satisfied clause, one or both of a_i and b_i belong to S_2, exactly two edges in A_2 arising from that clause must be good. Furthermore, by our previous comments, every edge in A_1 is good. Thus we have at least $W = |A_1| + 2k$ good edges.

Now, suppose we have a partition $N = S_1 \cup S_2$ for which W or more edges are good. Since $k > 0$ and $|A_2| \leqslant 3p$, the number of bad edges cannot exceed $3p$. By our previous discussion, this implies that all the F_i must belong to the same set, say S_1. For the same reason, exactly one of each pair $x_i, \bar{x}_i$ must belong to S_1. Thus, a consistent truth assignment is obtained by setting x_i "true" if and only if x_i belongs to S_2. For this truth assignment, clause i is satisfied whenever a_i or b_i or both belong to S_2. However, it is not difficult to see that, of the edges in A_2 arising from clause i, exactly two are good if one or both of a_i and b_i belong to S_2 and none are good if a_i and b_i both belong to S_1. Therefore, since at least $2k$ edges from A_2 must be good, this truth assignment must satisfy at least k clauses. □

An easy corollary to the completeness of Simple Max Cut concerns the following problem:

Minimum Cut Into Equal-Sized Subsets

Input: Graph $G = (N, A)$, two distinguished nodes s and t, positive integer W.

Property: There is a partition $N = S_1 \cup S_2$ with $S_1 \cap S_2 = \emptyset$, $|S_1| = |S_2|$, $s \in S_1$, $t \in S_2$, and $|\{\{u, v\} \in A : u \in S_1, v \in S_2\}| \leqslant W$.

Observe that this problem can be solved in polynomial time if no restriction is made as to the sizes of the subsets [13]. However, as defined, the problem is NP-complete, as we can conclude from the completeness of Simple Max Cut and the following:

Theorem 1.3. *Simple Max Cut* $\propto$ *Minimum Cut Into Equal-Sized Subsets.*

Proof. Given a graph $G = (N, A)$ and positive integer W, as input for Simple Max Cut, let $n = |N|$ and $U = \{u_1, u_2, ..., u_n\}$ satisfy $U \cap N = \emptyset$. The corresponding input for Minimum Cut Into Equal-Sized Subsets is the graph $G' = (N', A')$, nodes u_1 and u_n, and positive integer W', defined as follows:

$$N' = N \cup U;$$
$$A' = \{\{u, v\} : u, v \in N' \text{ and } \{u, v\} \notin A\};$$
$$W' = n^2 - W.$$

Suppose there is a partition $N = S_1 \cup S_2$ such that $|\{\{u, v\} \in A : u \in S_1, v \in S_2\}| \geqslant W$. Since W is positive, both S_1 and S_2 are nonempty. Let $j = n - |S_1|$. Form $S_1' = S_1 \cup \{u_1, u_2, \ldots, u_j\}$ and $S_2' = N' - S_1'$. Then $N' = S_1' \cup S_2'$ is a partition for G' with $|S_1'| = |S_2'| = n$, $u_1 \in S_1'$, $u_n \in S_2'$, and

$$\begin{aligned}|\{\{u, v\} \in A' : u \in S_1', v \in S_2'\}| &= n^2 - |\{\{u, v\} \notin A' : u \in S_1', v \in S_2'\}| \\ &= n^2 - |\{\{u, v\} \in A : u \in S_1, v \in S_2\}| \\ &\leqslant n^2 - W = W'.\end{aligned}$$

Now, suppose there is a partition $N' = S_1' \cup S_2'$, with $u_1 \in S_1'$, $u_n \in S_2'$, and $|S_1'| = |S_2'| = n$ such that $|\{\{u, v\} \in A' : u \in S_1', v \in S_2'\}| \leqslant n^2 - W = W'$. Then $N = S_1 \cup S_2$, where $S_1 = S_1' \cap N$ and $S_2 = S_2' \cap N$, is a partition for G satisfying

$$\begin{aligned}|\{\{u, v\} \in A : u \in S_1, v \in S_2\}| &= |\{\{u, v\} \notin A' : u \in S_1', v \in S_2'\}| \\ &= n^2 - |\{\{u, v\} \in A' : u \in S_1', v \in S_2'\}| \\ &\geqslant n^2 - (n^2 - W) = W.\end{aligned}$$

Thus G has a cut of weight greater than or equal to W if and only if G' has a cut with weight not exceeding W', which separates u_1 and u_n and divides the nodes of the graph into two equal sized subsets. The reduction is proved. □

A useful restatement of Simple Max Cut is:

Minimum Edge-Deletion Bipartite Subgraph

Input: Graph $G = (N, A)$, positive integer k.

Property: G has a bipartite subgraph formed by deleting k or fewer edges.

That the following node-deletion version of this problem is also *NP*-complete follows from Theorem 1.4 below.

Minimum Node-Deletion Bipartite Subgraph

Input: Graph $G = (N, A)$, positive integer k.

Property: G has a bipartite subgraph formed by deleting k or fewer vertices.

Theorem 1.4. *Clique* α *Minimum Node-Deletion Bipartite Subgraph.*

Proof. Given a graph $G = (N, A)$ and positive integer j as input to Clique (for a formal definition of Clique see the Appendix), let $n = |N|$ and $U = \{u_1, u_2, \ldots, u_n\}$ where $U \cap N = \emptyset$. The corresponding input for Minimum Node-Deletion Bipartite

Subgraph is the graph $G' = (N', A')$ and integer k defined as follows:

$$N' = N \cup U;$$
$$A' = \{\{u, v\}: u, v \in N', \{u, v\} \notin A, \text{ and } |\{u, v\} \cap U| \leqslant 1\};$$
$$k = n-j.$$

The reader may verify that G contains a clique of j nodes if and only if G' has a bipartite subgraph formed by deleting $n-j$ or fewer nodes. □

The final result of this section concerns the Optimal Linear Arrangement problem [1], defined as follows:

Optimal Linear Arrangement

Input: Graph $G = (N, A)$, weighting function $w: A \to \mathbf{Z}$, positive integer W.

Property: There is a $1-1$ function $f: N \to \mathbf{Z}$ such that

$$\sum_{\{u, v\} \in A} w(\{u, v\}) \cdot |f(u)-f(v)| \leqslant W.$$

This problem is a special case of the well-known quadratic assignment problem and a number of related facility location and component placement problems. We use a reduction from Simple Max Cut to show that this problem is *NP*-complete, *even* in the restricted case where all edge weights are required to be 1 (which we call Simple Optimal Linear Arrangement).

Theorem 1.5. *Simple Max Cut $\propto$ Simple Optimal Linear Arrangement*

Proof. Given a graph $G = (N, A)$ and positive integer k as input for Simple Max Cut, let $n = |N|$, $r = n^4$, and $U = \{u_1, u_2, \ldots, u_r\}$ where $U \cap N = \emptyset$. The corresponding input for Simple Optimal Linear Arrangement is the graph $G' = (N', A')$ and positive integer W defined as follows:

$$N' = N \cup U;$$
$$A' = \{\{u, v\}: u, v \in N' \text{ and } \{u, v\} \notin A\};$$
$$W = \binom{n^4+n+1}{3} - kn^4.$$

(Notice that $\binom{t+1}{3} = \sum_{1 \leqslant u < v \leqslant t} (v-u)$ which is the minimum W achievable for a complete graph on t nodes.)

Suppose we have a partition $N = S_1 \cup S_2$ which satisfies $|\{\{u, v\} \in A: u \in S_1, v \in S_2\}| \geqslant k$. Let $S_1 = \{a_1, a_2, \ldots, a_t\}$ and $S_2 = \{b_1, b_2, \ldots, b_{n-t}\}$. Define f as follows:

$$f(a_i) = i, \qquad 1 \leqslant i \leqslant t;$$

$$f(u_i) = t+i,\ 1 \leqslant i \leqslant n^4;$$
$$f(b_i) = n^4+t+i,\ 1 \leqslant i \leqslant n-t.$$

Then

$$\sum_{\{u,v\}\in A'} |f(u)-f(v)| = \binom{n^4+n+1}{3} - \sum_{\{u,v\}\notin A'} |f(u)-f(v)|$$
$$= \binom{n^4+n+1}{3} - \sum_{\{u,v\}\in A} |f(u)-f(v)|$$
$$\leqslant \binom{n^4+n+1}{3} - kn^4 = W.$$

Now suppose there exists a $1-1$ function $f\colon N' \to \mathbf{Z}$ such that

$$\sum_{\{u,v\}\in A'} |f(u)-f(v)| \leqslant W.$$

Then there exists such an f having range $\{1, 2, ..., n^4+n\}$. Let F denote the set of $1-1$ functions $f\colon N' \to \{1, 2, ..., n^4+n\}$. Observe that for any $f\in F$

$$\sum_{\{u,v\}\in A'} |f(u)-f(v)| + \sum_{\{u,v\}\in A} |f(u)-f(v)| = \sum_{1\leq i<j\leq n^4+n} |j-i| = \binom{n^4+n+1}{3}.$$

Therefore, there exists an $f\in F$ such that

$$\sum_{\{u,v\}\in A} |f(u)-f(v)| \geqslant kn^4.$$

Define

$$W^* = \max_{f\in F} \sum_{\{u,v\}\in A} |f(u)-f(v)|$$

and

$$F^* = \Big\{f\in F\colon \sum_{\{u,v\}\in A} |f(u)-f(v)| = W^*\Big\}.$$

Clearly $W^* \geqslant kn^4$ and F^* is nonempty. We shall now show that there is at least one $f\in F^*$ which maps the elements of U into a set of r consecutive integers, thereby partitioning N into those vertices that go before and those that come afterwards. For each $f\in F^*$, define the set

$$S(f) = \{v\in N\colon \exists u_i, u_j \in U \quad \text{with } f(u_i) < f(v) < f(u_j)\}$$

and let $m(f) = |S(f)|$. Then there exists a function $g\in F^*$ such that $m(g) \leqslant m(f)$ for all $f\in F^*$. We show that $m(g) = 0$, and hence g is our desired mapping.

Suppose $m(g) > 0$. Let $v_0 \in S(g)$ be such that $g(v_0) \geqslant g(v)$ for all $v\in S(g)$. For each $v\in N'$, define

$$L(v) = |\{u\in N'\colon \{u, v\}\in A \quad \text{and } g(u) < g(v)\}|$$

and

$$R(v) = |\{u\in N'\colon \{u, v\}\in A \quad \text{and } g(u) > g(v)\}|.$$

Note that $v \in U$ implies $L(v) = R(v) = 0$. Suppose $L(v_0) \geqslant R(v_0)$. Let $u_0 \in U$ be such that $g(u_0) \geqslant g(u)$ for all $u \in U$. Then by definition of v_0, $g(v_0) < g(v) \leqslant g(u_0)$ implies that $v \in U$. Consider the function $\bar{g} \in F$ which is identical to g except that $\bar{g}(v_0) = g(u_0)$ and $\bar{g}(u_0) = g(v_0)$. It is not difficult to see that

$$\sum_{\{u,v\}\in A} |\bar{g}(u) - \bar{g}(v)| \geqslant W^* \quad \text{and } m(\bar{g}) < m(g)$$

which contradicts either the definition of W^* or the choice of g. Thus, $L(v_0) < R(v_0)$. Let

$$t = \max\{g(v) : v \in N', g(v) < g(v_0) \quad \text{and } L(v) \geqslant R(v)\}.$$

The value of t is well-defined since there exists a $u \in U$ with $g(u) < g(v_0)$ and $L(u) = R(u) = 0$. Thus, if $g(v_1) = t$ and $g(v_2) = t+1$, we must have $L(v_2) < R(v_2)$ The function $\bar{g} \in F$, which is identical to g except that $\bar{g}(v_1) = g(v_2)$ and $\bar{g}(v_2) = g(v_1)$, satisfies

$$\sum_{\{u,v\}\in A} |\bar{g}(u) - \bar{g}(v)| > \sum_{\{u,v\}\in A} |g(u) - g(v)| = W^*,$$

contradicting the definition of W^*. Therefore, we must have $m(g) = 0$.

Since $m(g) = 0$, the elements of U are mapped by g to a set of consecutive integers. Define a partition $N = S_1 \cup S_2$ by

$$S_1 = \{v \in N : g(v) < g(u) \quad \text{for all } u \in U\},$$

$$S_2 = \{v \in N : g(v) > g(u) \quad \text{for all } u \in U\}.$$

We now have

$$\begin{aligned} kn^4 &\leqslant \sum_{\{u,v\}\in A} |g(u) - g(v)| \\ &= \sum_{\substack{\{u,v\}\in A \\ u,v\in S_1}} |g(u) - g(v)| + \sum_{\substack{\{u,v\}\in A \\ u,v\in S_2}} |g(u) - g(v)| + \sum_{\substack{\{u,v\}\in A \\ u\in S_1, v\in S_2}} |g(u) - g(v)| \\ &\leqslant \binom{n+1}{3} + \binom{n+1}{3} + (n^4+n)|\{\{u,v\}\in A : u \in S_1, v \in S_2\}| \\ &\leqslant \frac{n^3}{6} + \frac{n^3}{6} + \frac{n^3}{2} + n^4 \cdot |\{\{u,v\}\in A : u \in S_1, v \in S_2\}| \end{aligned}$$

which, since k is an integer, implies

$$|\{\{u,v\}\in A : u \in S_1, v \in S_2\}| \geqslant k.$$

This completes the proof of Theorem 1.5.

2. Restricted Graph Problems

Many of the reductions which were first used to show certain graph theoretic problems to be *NP*-complete involved the construction of rather complicated graphs, highly non-planar and with nodes having arbitrarily high degree. Since in many practical problems node degree may be bounded (e.g., fan-in, fan-out restrictions on circuit elements), or graphs may be planar, it is worthwhile to determine whether the complexity of the graphs involved in these reductions was necessary.

In certain cases, we can observe trivially that it is. For example, consider the problem Clique [13]. Since the largest clique possible in a planar graph has size 4, and the largest clique possible in a graph with maximum node degree k has size $k+1$, we can find the largest clique in either case in polynomial time by examining all subsets of 4 or fewer ($k+1$ or fewer) nodes, in time proportional to at most n^4 or n^{k+1}, respectively.

More interesting are the cases where the answer is not readily apparent. For instance, it is implicit in the literature that Max Cut, when restricted to planar graphs, can be solved in polynomial time. [14] presents a polynomial time procedure for reducing the problem of finding the maximum cut in a weighted planar graph to that of finding a minimum weighted matching in a complete graph derived from the dual of the original graph. Although [14] then resorts to a non-polynomial branch and bound technique, the weighted matching can be found in polynomial time using a method of Edmonds [4].

On the other hand, we have found that a number of graph problems remain *NP*-complete even when restricted to planar graphs and graphs with limited node degree. In this section, we shall present these completeness results, which concern Graph k-Colorability, Node Cover, and Hamiltonian Circuit. The formal definitions of these problems appear in the appendix.

The following table gives the principal restricted versions of these problems which we prove to be *NP*-complete:

Problem	Node degree at most
1. Planar Graph 3-Colorability	4
2. Undirected Hamiltonian Circuit	3
3. Planar Directed Hamiltonian Path	4-Out, 3-In
4. Node Cover	3
5. Planar Node Cover	6

For results 1, 3 and 5, it was not previously known if the planar problems were complete, even if *no* restrictions were placed on node degrees. In fact, concerning result 1, it was previously known only that Graph k-Colorability, with k an input variable, was *NP*-complete.

The degree constraints in 1, 2, and 4 are all best possible, in that each of the problems becomes easy if the restriction on node degree is reduced by 1. Node Cover and Undirected Hamiltonian Circuit are clearly trivial for graphs with maximum degree 2, and a wellknown result of Brooks [2] implies that a connected graph with maximum degree 3 is 3-colorable if and only if it differs from K_4, the complete graph on four nodes, which is easy to determine.

In addition to the above results, there are a number of more or less immediate corollaries. Result 2 implies that Directed Hamiltonian Circuit with node degree bounded by 3-Out, 3-In is *NP*-complete; however, the largest degree bounds for which we know this problem to be easy are 2-Out, 1-In or 2-In, 1-Out. Also, we may substitute Path for Circuit in result 2. However, we do not know whether Planar Directed Hamiltonian Circuit, Planar Undirected Hamiltonian Circuit, or Planar Undirected Hamiltonian Path are *NP*-complete, and these remain significant open problems.

The proofs of the results given in the table follow. For each problem, we show that there is a known *NP*-complete problem which reduces to it.

Theorem 2.1. *Sat3 α Graph 3-Colorability.*

Proof. The key construct in our proof is the graph H shown in Fig. 1. The graph H has two important properties which are straightforward to verify.

(2.1A) Any coloring of the nodes a, b, and c such that $1 \in \{f(a), f(b), f(c)\}$ can be extended to a valid 3-coloring f for H which has $f(y_6) = 1$.

(2.1B) If f is a valid 3-coloring of H with $f(a) = f(b) = f(c) = i$, then $f(y_6) = i$.

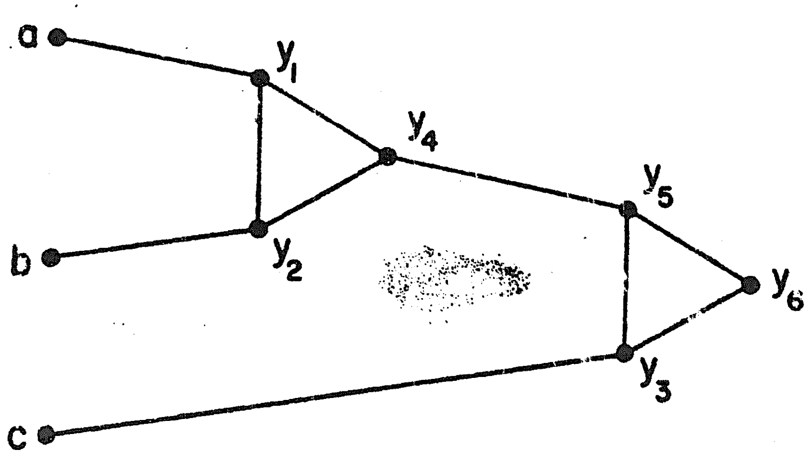


Fig. 1. Graph H for Theorem 2.1.

Let $C = \{C_1, C_2, \ldots, C_p\}$ be any set of clauses, in variables $x_1, x_2, \ldots, x_n$, given as input for Sat3. As in the proof of Theorem 1.1, we may assume that each clause contains exactly 3 literals and label them by $C_i = (a_i \vee b_i \vee c_i)$. We shall construct a graph G which is 3-colorable if and only if C is satisfiable.

The set N of nodes for G is given by

$$N = \{v_1, v_2, v_3\} \cup \{x_i, \bar{x}_i : 1 \leqslant i \leqslant n\} \cup \{y_{ij} : 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant 6\}.$$

The set A of edges for G is given by

$$\begin{aligned}A = &\{\{v_1, v_2\}, \{v_2, v_3\}, \{v_1, v_3\}\}\\ &\cup \{\{x_i, \bar{x}_i\}: 1 \leqslant i \leqslant n\}\\ &\cup \{\{v_3, x_i\}, \{v_3, \bar{x}_i\}: 1 \leqslant i \leqslant n\}\\ &\cup \{\{a_i, y_{i1}\}, \{b_i, y_{i2}\}, \{c_i, y_{i3}\}: 1 \leqslant i \leqslant p\}\\ &\cup \{\{v_2, y_{i6}\}, \{v_3, y_{i6}\}: 1 \leqslant i \leqslant p\}\\ &\cup \{\{y_{i1}, y_{i2}\}, \{y_{i1}, y_{i4}\}, \{y_{i2}, y_{i4}\}: 1 \leqslant i \leqslant p\}\\ &\cup \{\{y_{i3}, y_{i5}\}, \{y_{i3}, y_{i6}\}, \{y_{i5}, y_{i6}\}: 1 \leqslant i \leqslant p\}\\ &\cup \{\{y_{i4}, y_{i5}\}: 1 \leqslant i \leqslant p\}.\end{aligned}$$

Observe that for each clause C_i in the original input, the subgraph consisting of $y_{i1}, y_{i2}, y_{i3}, y_{i4}, y_{i5}, y_{i6}$ and the variable nodes corresponding to a_i, b_i, and c_i is just a copy of our graph H.

Now consider any satisfying truth assignment for C. Define $f: N \to \{y_{ij}: 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant 6\}$ by setting $f(v_1) = 1$, $f(v_2) = 2$, $f(v_3) = 3, f(x_i) = 1$ and $f(\bar{x}_i) = 2$ for x_i true, and $f(x_i) = 2$ and $f(\bar{x}_i) = 1$ for x_i false. Clearly f assigns different values to adjacent nodes. Furthermore, since the truth assignment satisfies C, $1 = f(v_1) \in \{f(a_i), f(b_i), f(c_i)\}$ for each i, $1 \leqslant i \leqslant p$. Therefore, by (2.1A), f can be extended to a 3-coloring $f: N \to \{1, 2, 3\}$ for G.

Conversely, suppose $f: N \to \{1, 2, 3\}$ is any 3-coloring of G. Since the edges in A force $\{f(x_i), f(\bar{x}_i): 1 \leqslant i \leqslant n\} = \{f(v_1), f(v_2)\}$ and $\{f(y_{i6}): 1 \leqslant i \leqslant p\} = \{f(v_1)\}$, it follows from (2.1B) that $f(v_1) \in \{f(a_i), f(b_i), f(c_i)\}$ for each i, $1 \leqslant i \leqslant p$. Since we also must have $f(x_i) \neq f(\bar{x}_i)$, $1 \leqslant i \leqslant n$, it follows immediately that setting x_i true if and only if $f(x_i) = f(v_1)$ gives a truth assignment which satisfies C.

Thus C is satisfiable if and only if G is 3-colorable, and the reduction is proved. □

Theorem 2.2. *Graph 3-Colorability α Planar Graph 3-colorability.*

Proof. The key structure used in this proof is the graph H pictured in Fig. 2, which will be called a *crossover* with *outlets* u, u', v, and v' as labelled. This crossover, simpler than our original, was provided by Michael J. Fischer. H has 13 nodes and obeys the following properties, as the reader may readily verify.

(2.2A) Any valid 3-coloring of H gives the same color to u and u', and the same color to v and v'.

(2.2B) For any $(i, j) \in \{1, 2, 3\} \times \{1, 2, 3\}$, there exists a 3-coloring of H using colors 1, 2, and 3 such that u and u' receive color i, and v and v' receive color j.

Given a graph $G = (N, A)$, we construct a planar graph $G' = (N', A')$ as follows (see Fig. 3):

(i) Embed G in the plane, allowing edges to cross each other, but such that no more than two edges meet at any one point (other than their mutual endpoint) and no edge touches a node other than its own endpoint. (This can be done in any number of standard ways in polynomial time).

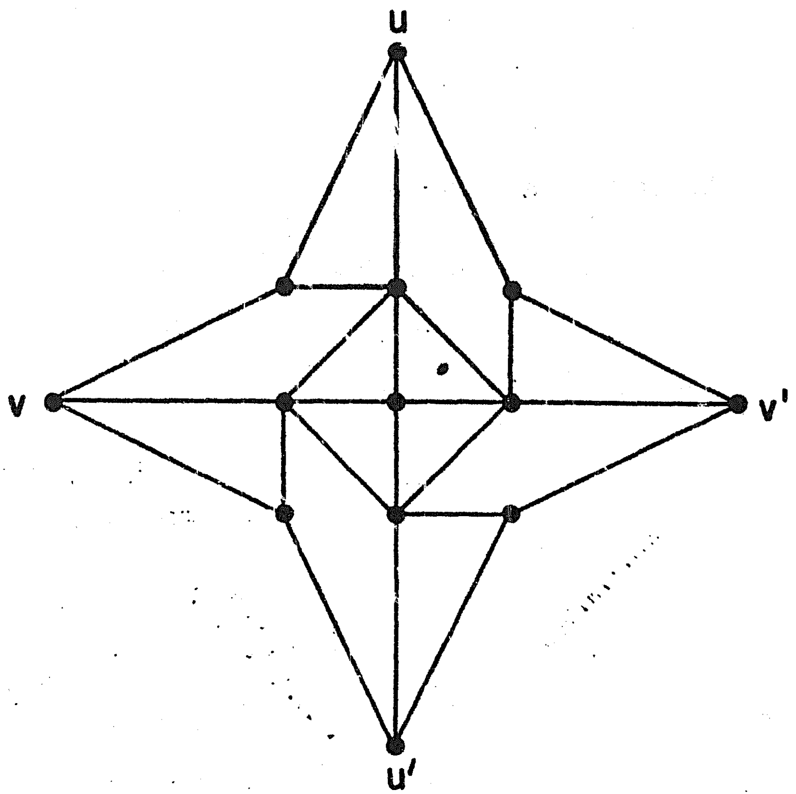


Fig. 2. Crossover H for Theorem 2.2.

(ii) For each edge $\{x, y\} \in A$, call its representation in the plane the $\{x, y\}$-*line*. To each such line which is "crossed" by other lines, add new points, one between each endpoint and the nearest crossing to it, and one between each pair of adjacent crossings.

(iii) Replace each crossing in the graph by a copy of graph H, identifying the outlets u and u' with the nearest new points on either side of the crossing on one of the lines involved, and identifying v and v' with the nearest new points on the other line.

(iv) For each $\{x, y\} \in A$, choose one endpoint as the *distinguished endpoint* and coalesce it with the nearest new point on the $\{x, y\}$-line. The edge between the other endpoint and *its* nearest new point on the $\{x, y\}$-line will be called the *operant edge* of the $\{x, y\}$-line.

This completes the construction of G'.

Suppose G' is 3-colorable and let $f: N' \to \{1, 2, 3\}$ be a valid 3-coloring. Then f restricted to $N \subseteq N'$ will be a valid 3-coloring of G. For suppose not. Then there would exist an $\{x, y\} \in A$ such that $f(x) = f(y)$. Consider the $\{x, y\}$-line in G', and assume without loss of generality that x is the distinguished endpoint for this line chosen in Step (iv). Then by (2.2A) all the new points on the $\{x, y\}$-line must have the same color as x. Thus both endpoints of the operant edge for that line have the same color, a contradiction.

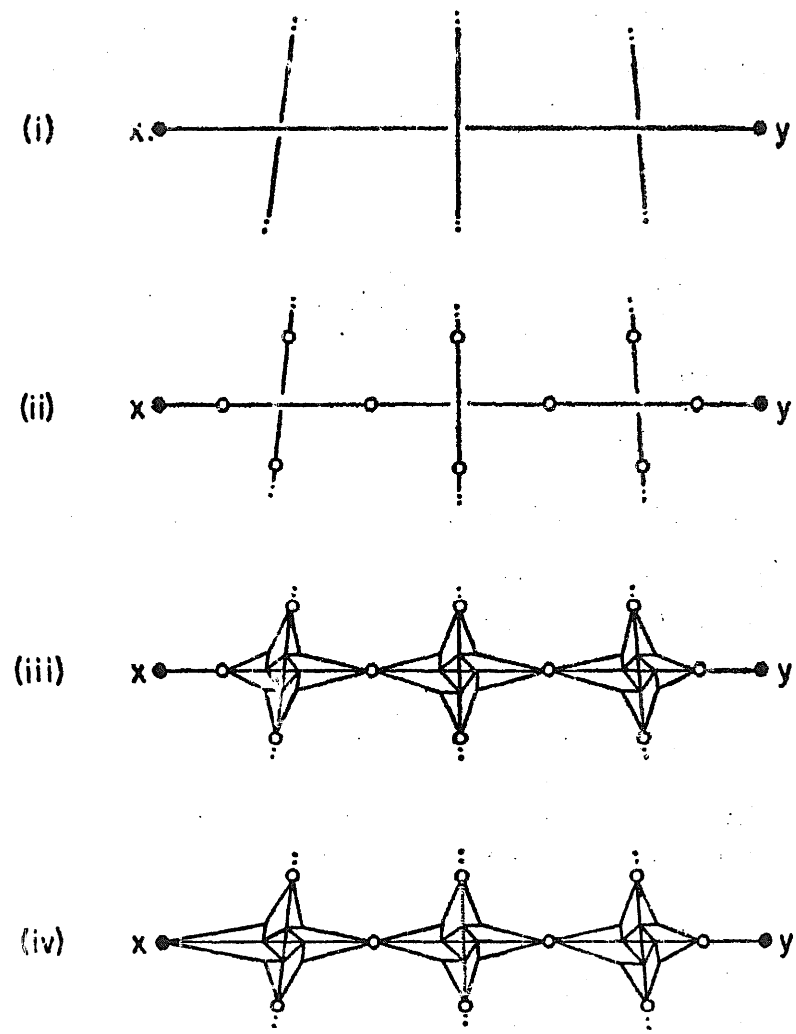


Fig. 3. Construction of G', as it affects the $\{x, y\}$-line.

Conversely, if $f: N \to \{1, 2, 3\}$ is a valid 3-coloring for G, it can be extended to a 3-coloring for G' as follows: For each $\{x, y\} \in A$, color each new point on the $\{x, y\}$-line with color $f(x)$, where x is the distinguished endpoint of the line. This insures that all the operant edges of G' are validly colored. By (2.2B) this 3-coloring can be extended to the interior nodes of the crossovers, thus yielding a valid 3-coloring of G'.

Thus G' is 3-colorable if and only if G is, and the reduction is proved. □

Theorem 2.3. *Planar Graph* 3-*Colorability* α *Planar Graph* 3-*Colorability With Node Degree At Most* 4.

Proof. The key to our construction will be the use of "node substitutes". Fig. 4(a) shows the 3-outlet node substitute H_3, with its first, second, and third outlet nodes labelled. For $k \geqslant 4$, the k-outlet node substitute H_k is formed by adjoining to H_{k-1} a copy of H_3 having its first outlet coinciding with outlet $k-1$ of H_{k-1}. The outlet nodes of H_k are the nodes having degree 2. The outlets which originally belonged to H_{k-1} retain the same labels, with the second outlet of the adjoined H_3 becoming outlet $k-1$ and its third outlet becoming outlet k. Fig. 4(b) shows H_5.

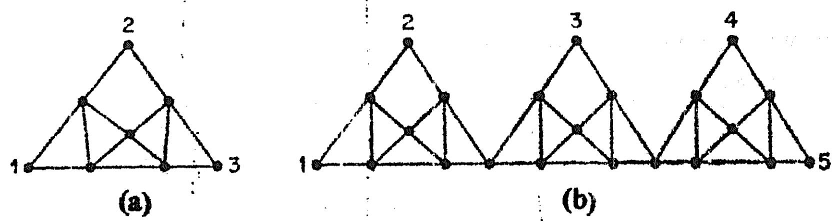


Fig. 4. Node substitutes H_3 and H_5.

It is easy to prove by induction that, for all $k \geq 3$, the following facts hold:

(2.3A) H_k has $7(k-2)+1$ nodes, including k outlets.
(2.3B) No node of H_k has degree exceeding 4.
(2.3C) H_k is planar.
(2.3D) H_k is 3-colorable, but not 2-colorable, and every valid 3-coloring of H_k assigns the same color to every outlet node.

Given any planar graph G, we show how to construct a planar graph G', using node substitutes, which has maximum degree 4 and which is 3-colorable if and only if G is 3-colorable.

Fix a planar embedding of G and arbitrarily designate the r nodes of G which have degree exceeding 4 as $v_1, v_2, ..., v_r$. We construct a sequence of graphs $G = G_0, G_1, ..., G_r = G'$ as follows: G_i is constructed from G_{i-1}. Let d be the degree of v_i in G_{i-1} and let $\{u_1, v_i\}, \{u_2, v_i\}, ..., \{u_d, v_i\}$ be the edges incident with v_i, taken in clockwise order. To form G_i, delete node v_i from G_{i-1}, replacing it with a copy of H_d, and replace each edge $\{u_j, v_i\}$ by an edge joining u_j to outlet j of the node substitute.

It follows from the construction and previously stated facts that, for $0 \leq k \leq r$, G_k is planar, G_k has $r-k$ nodes with degree exceeding 4, and G_k is 3-colorable if and only if G is 3-colorable. Thus, $G' = G_r$ satisfies all the required properties, completing the proof. □

Theorem 2.4. *Undirected Hamiltonian Circuit* α *Undirected Hamiltonian Circuit With Node Degree At Most* 3.

Proof. This construction will also use a "node substitute", which is formed from a special graph, called a *fan*. The one-outlet fan F_1 consists simply of a single node. The single node, labelled $U_{1,1}$, is both the *inlet* and the *outlet* of F_1. Inductively, assume we have defined the k-outlet fan F_k, $k \geq 1$, with inlet $U_{1,1}$ and outlets $U_{k,1}$ through $U_{k,k}$. The $(k+1)$-outlet fan F_{k+1} is formed from F_k by adding the following nodes and edges:

Nodes: $U_{k+1,i}$, $1 \leqslant i \leqslant k+1$; $S_{k+1,i}$, $1 \leqslant i \leqslant k+1$.

Edges: $\{U_{k,i}, S_{k+1,i}\}$, $1 \leqslant i \leqslant k$;
$\{U_{k+1,i}, S_{k+1,i}\}$, $1 \leqslant i \leqslant k+1$;
$\{U_{k+1,i+1}, S_{k+1,i}\}$, $1 \leqslant i \leqslant k$;
$\{S_{k+1,k+1}, U_{k+1,1}\}$.

The inlet of F_{k+1} is $U_{1,1}$ and its outlets are $U_{k+1,1}$ through $U_{k+1,k+1}$. Fig. 5 shows F_1, F_2, and F_3.

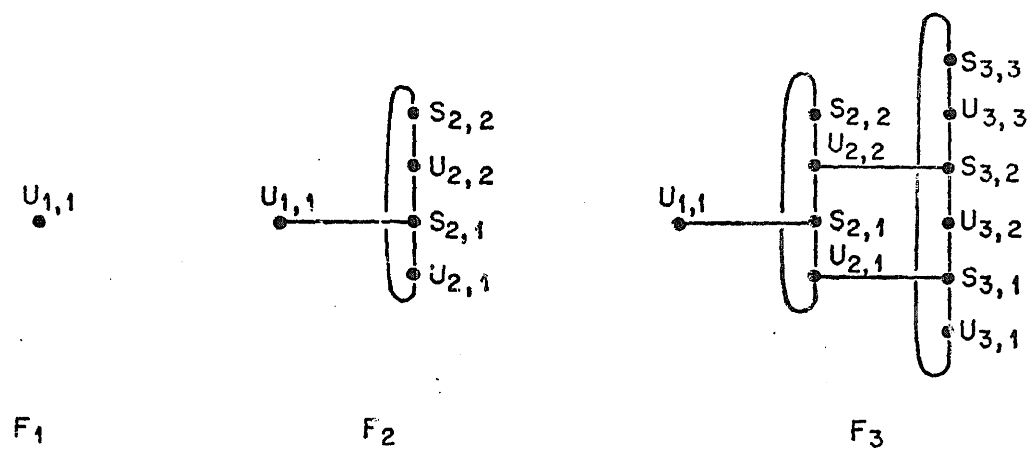


Fig. 5. Fans F_1, F_2, and F_3.

It is easy to prove by induction that the following facts hold for all $k \geqslant 1$:

(2.4A) F_k contains k^2+k-1 nodes, none with degree exceeding 3.

(2.4B) F_k has one inlet node of degree 1 and k outlet nodes, none with degree exceeding 2.

(2.4C) For any outlet node of F_k, there exists a path from the inlet to that outlet which includes each node of F_k exactly once.

One more property of F_k will be required and, since its proof is not quite as straightforward, we present it as a lemma.

Lemma 2.4.1. *Suppose a graph G contains a subgraph H isomorphic to F_k, $k \geqslant 1$, in such a way that*

(i) *no two nodes of H are adjacent in G unless the corresponding nodes of F_k are adjacent, and*

(ii) *any node of H which is adjacent to a node of G not belonging to H corresponds to either an inlet or outlet node of F_k.*

Then, any Hamiltonian circuit of G contains a path from the "inlet" of H to some "outlet" of H, consisting precisely of all the nodes of H.

Proof of Lemma. The Lemma holds trivially for $k = 1$. Suppose it holds for F_{k-1}, $k > 1$, and consider a graph G which contains a subgraph H isomorphic to F_k in the specified manner, and which contains a Hamiltonian circuit C. We consider the nodes of H as being labelled identically with the corresponding nodes of F_k. Observe that H, and hence G, contains a subgraph H' isomorphic to F_{k-1} which has inlet node $U_{1,1}$ and which satisfies the two conditions of the Lemma. By the induction hypothesis, C contains a path from $U_{1,1}$ to some $U_{k-1,j}$ which includes precisely the nodes belonging to H'. The node $U_{k-1,j}$ and the remaining nodes of H are shown in Fig. 6.

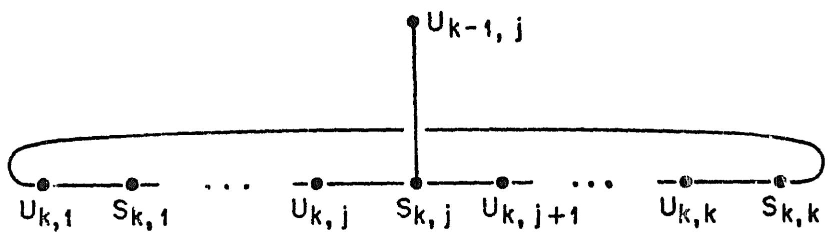


Fig. 6. Remaining nodes of H.

Consider the set T of nodes of the form $U_{k,i}$ and $S_{k,i}$, $1 \leqslant i \leqslant k$. By the construction of F_k and the assumptions on H, there are only two ways by which the nodes of T can be accessed by circuit C:

(a) From nodes of G not in H via an outlet $U_{k,i}$.

(b) From an outlet $U_{k-1,i}$ of H', via the corresponding $S_{k,i}$.

Since C contains a path from $U_{1,1}$ to $U_{k-1,j}$ which uses all the nodes of H', the only way that (b) can occur is via the edge $\{U_{k-1,j}, S_{k,j}\}$. Using this edge, the path from $U_{1,1}$ to $U_{k-1,j}$ can be extended to a path from $U_{1,1}$ to either $U_{k,j}$ or $U_{k,j+1}$ consisting precisely of the nodes belonging to H. If this is *not* what occurs in C, then either that edge is not used, or it is used and C exits from the set of nodes T before all nodes of T have been covered. In either case, a non-zero and equal number of U-type and S-type nodes from T will remain to be covered by subsequent visits of the Hamiltonian circuit to the set T. However each such visit of T by C must both enter and leave via a U-type node, and hence must use one more U-type node than S-type node. Thus, not all the S-type nodes could be covered by C, contradicting the fact that C is a Hamiltonian circuit. Therefore, C must contain a path from $U_{1,1}$ to some outlet $U_{k,i}$ consisting precisely of all the nodes of H, as claimed. The Lemma follows by induction. □

Given a graph $G = (N, A)$, we now show how to construct a graph $G' = (N', A')$, having maximum node degree 3, such that G' contains a Hamiltonian circuit if and only if G does.

Let $N = \{v_1, v_2, \ldots, v_n\}$ where $n = |N|$. Let D_n denote the double-fan formed by joining two copies of F_n with an edge connecting their inlet nodes. The outlet nodes of one of the copies of F_n are the inlet nodes for D_n, and the outlet nodes of the other copy are the outlet nodes of D_n. For each node v_i in N, the graph G' contains a copy $D_n(i)$ of D_n. The inlets of $D_n(i)$ will be denoted by $v_1(i), v_2(i), ..., v_n(i)$ and its outlets by $u_1(i), u_2(i), ..., u_n(i)$. The specification of G' is completed by including in G', for each edge $\{v_i, v_j\} \in A$, the two edges $\{u_j(i), v_i(j)\}$ and $\{u_i(j), v_j(i)\}$.

The following useful properties of double-fans D_n are immediate consequences of the corresponding properties for F_n:

(2.4a) D_n contains $2(n^2+n-1)$ nodes, none with degree exceeding 3.

(2.4b) The n inlet nodes and n outlet nodes of D_n each have degree not exceeding 2.

(2.4c) For each outlet and each inlet of D_n, there is a path between them which includes every node of D_n exactly once.

(2.4d) Suppose an undirected graph H contains a subgraph D isomorphic to D_n in the manner specified in Lemma 2.4.1. Then every Hamiltonian circuit of H contains a path from an "inlet" of D to an "outlet" of D, containing precisely all the nodes of D.

Observe that G' has maximum degree 3, since each inlet or outlet of a $D_n(i)$ has at most one edge joining it to a node not belonging to $D_n(i)$, using properties (2.4a) and (2.4b).

Suppose G' has a Hamiltonian circuit C, i.e., an ordering of the nodes as $y_1, y_2, ..., y_m$, where $m = |N'|$, such that for all j, $1 \leq j \leq m$, $\{y_j, y_{j+1}\} \in A'$ and such that $\{y_m, y_1\} \in A'$. We may assume without loss of generality that y_1 and y_m do not belong to the same double-fan $D_n(i)$. Thus, by construction of G', one of y_1, y_m must be an inlet of some fan $D_n(i)$ and the other must be an outlet of another fan $D_n(j)$. Since G' is an undirected graph, we may assume that y_1 is an inlet of some fan $D_n(i)$. Then, by (2.4d), C determines an ordering $D_n(i_1), D_n(i_2), ..., D_n(i_n)$ of the double-fans such that all the nodes of $D_n(i_k)$ precede all the nodes of $D_n(i_j)$ in C whenever $1 \leq k < j \leq n$. But the construction of G' implies that $v_{i_1}, v_{i_2}, ..., v_{i_n}$ must then be a Hamiltonian circuit of G. (Notice that this argument fails when $n = 2$, however, in this case we may let $G' = G$.)

Conversely, suppose $v_{i_1}, v_{i_2}, ..., v_{i_n}$ is a Hamiltonian circuit C of G. By construction of G some outlet of $D_n(i_j)$ must be connected to some inlet of $D_n(i_{j+1})$ in G', for $1 \leq j < n$, and similarly for $D_n(i_n)$ and $D_n(i_1)$. It is then a simple matter to construct a Hamiltonian circuit in G' using (2.4c).

Therefore, we have shown that G' contains a Hamiltonian circuit if and only if G does, completing the proof of Theorem 2.4. □

Theorem 2.5. *Exact Cover* α *Planar Directed Hamiltonian Path.*

Proof. Given any collection S of sets, we must construct a planar directed graph G which has a Hamiltonian path if and only if S contains an exact cover. We first introduce some terminology.

Let $G = (N, A)$ be a directed graph with nodes $p_1, p_2, ..., p_m$. The edges in A are ordered pairs $\langle p_i, p_j\rangle$. We will call p_i and p_j *adjacent* whenever either $\langle p_i, p_j\rangle \in A$

or $\langle p_j, p_i \rangle \in A$. Any Hamiltonian path (*H-path*) of G can be identified with a string $p_{i(1)}p_{i(2)} \cdots p_{i(m)}$ of nodes such that $N = \bigcup_{j=1}^{m} \{p_{i(j)}\}$ and $\langle p_{i(j)}, p_{i(j+1)} \rangle \in A$ for all $1 \leqslant j < m$.

Let N^* denote the set of all finite strings of elements from N. A *partial H-path* ω for G is an element of N^* which is a prefix of some H-path for G. For any partial

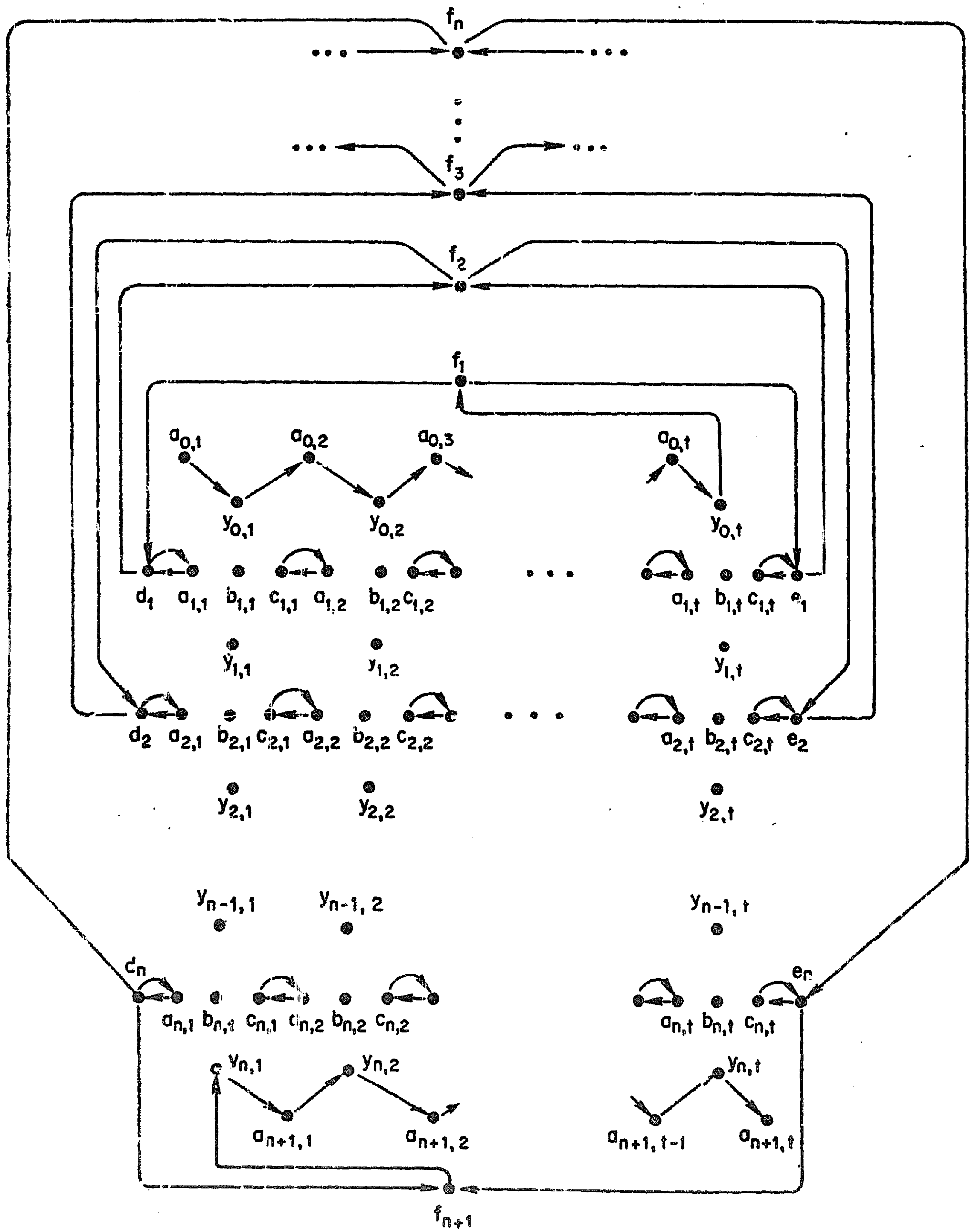


Fig. 7. The skeleton of G.

H-path ω for G, $\tau \in N^*$ is a *k-extension* of ω if the length of τ is k and $\omega\tau$ is a partial *H*-path for G.

For any string $\beta = b_1 b_2 \dots b_k \in N^*$, we say that $v \in N$ *belongs to* β if and only if $v = b_i$ for some i, $1 \leqslant i \leqslant k$, and $\langle u, v\rangle \in A$ *belongs to* β if and only if $u = b_i$ and $v = b_{i+1}$ for some i, $1 \leqslant i < k$.

Now suppose we are given a collection $S = \{S_1, S_2, \dots, S_n\}$ of sets with $\bigcup_{i=1}^{n} S_i = U = \{u_1, u_2, \dots, u_t\}$. The planar directed graph $G = (N, A)$ is specified as follows: The set N of nodes, which depends only on n and t, is

$$\begin{aligned} N = &\{b_{i,j}, c_{i,j}\colon 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant t\} \\ &\cup \{a_{i,j}\colon 0 \leqslant i \leqslant n+1, 1 \leqslant j \leqslant t\} \\ &\cup \{y_{i,j}\colon 0 \leqslant i \leqslant n, 1 \leqslant j \leqslant t\} \\ &\cup \{d_i, e_i, f_i\colon 1 \leqslant i \leqslant n\} \cup \{f_{n+1}\}. \end{aligned}$$

The set of edges is made up of two parts, $A = A_1 \cup A_2$, the first of which depends only on n and t and forms a skeleton for G (see Fig. 7):

$$\begin{aligned} A_1 = &\{\langle a_{0,j}, y_{0,j}\rangle, \langle y_{n,j}, a_{n+1,j}\rangle\colon 1 \leqslant j \leqslant t\} \\ &\cup \{\langle y_{0,j}, a_{0,j+1}\rangle, \langle a_{n+1,j}, y_{n+1,j+1}\rangle\colon 1 \leqslant j \leqslant t-1\} \\ &\cup \{\langle y_{0,t}, f_1\rangle, \langle f_{n+1}, y_{n,1}\rangle\} \\ &\cup \{\langle f_i, d_i\rangle, \langle f_i, e_i\rangle, \langle d_i, f_{i+1}\rangle, \langle e_i, f_{i+1}\rangle, \\ &\quad \langle d_i, a_{i,1}\rangle, \langle a_{i,1}, d_i\rangle, \langle e_i, c_{i,t}\rangle, \langle c_{i,t}, e_i\rangle \colon 1 \leqslant i \leqslant n\} \\ &\cup \{\langle c_{i,j}, a_{i,j+1}\rangle, \langle a_{i,j+1}, c_{i,j}\rangle\colon 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant t-1\}. \end{aligned}$$

The remaining edges in $A = A_1 \cup A_2$ flesh out the skeleton provided by A_1. For each i and j, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant t$, they connect certain nodes from $\{a_{i,j}, b_{i,j}, c_{i,j}, y_{i-1,j}, y_{i,j}\}$, depending on whether or not u_j belongs to S_i. If $u_j \notin S_i$, then A_2 contains the eight edges shown in Fig. 8A. If $u_j \in S_i$, then A_2 contains the seven edges shown in Fig. 8B.

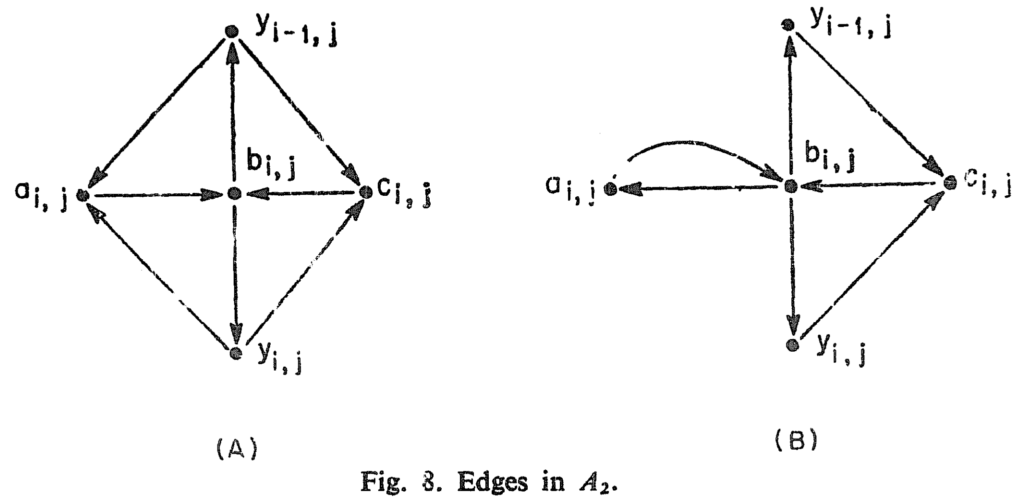


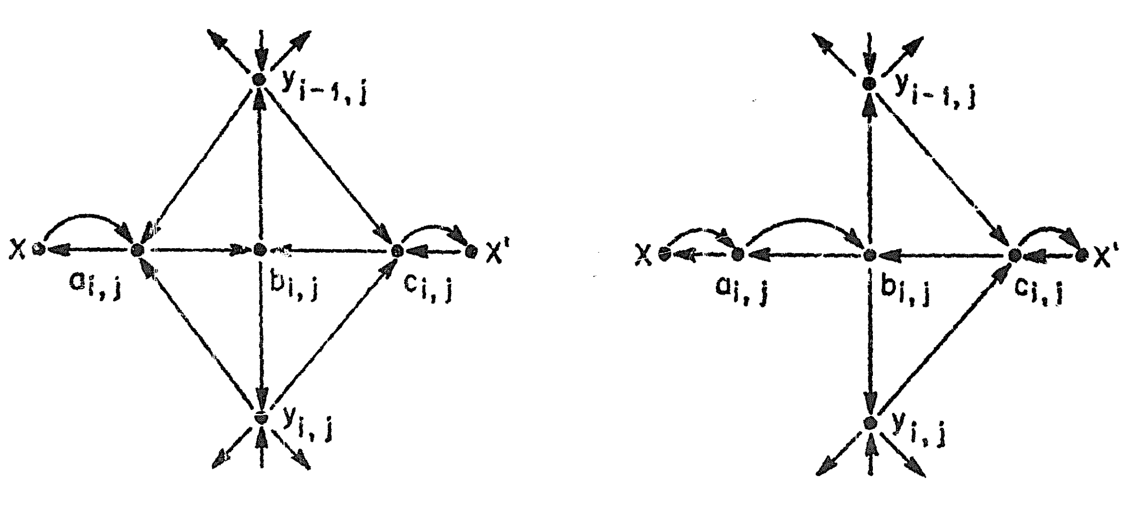
Fig. 8. Edges in A_2.

This completes the description of G. The reader should observe from Figs. 7 and 8 that G is planar.

Informally, the relationship between G and the covering problem is as follows. Suppose that we are building an H-path for G, in step-by-step fashion, and, at the same time, generating an exact cover for U. Clearly, the H-path must begin with $a_{0,1}, y_{0,1}, a_{0,2}, y_{0,2}, \ldots, a_{0,t}, y_{0,t}, f_1$. For each k, $1 \leqslant k \leqslant n$, at the step the H-path reaches f_k, we choose edge $\langle f_k, e_k \rangle$ if S_k is to be included in the cover, and edge $\langle f_k, d_k \rangle$ if it is not to be included. We then proceed left (right) along the $a_{k,j}, b_{k,j}, c_{k,j}$ line of the graph, exiting from $d_k(e_k)$, and finally arriving at f_{k+1}. Helping us in the choice of which direction to travel that line is the fact that the nodes $y_{k-1,j}$, $1 \leqslant j \leqslant t$, "remember" which elements of U have been covered by previously selected sets. Specifically, $y_{k-1,j}$ has already been visited by the partial H-path if and only if element v_j has *not* been covered yet. Note that, when we are about to make our first choice, at f_1, all nodes $y_{0,j}$ are already in the path and no element from U has been covered. When we reach f_{n+1}, after having made a decision for each set S_k none of the $y_{n,j}$ can have been visited by the partial H-path (or it could not be extended to an H-path), so all elements of U have been covered by the selected sets. The edges of A_2 force the transmission of information from one row of $y_{k,j}$ nodes to the next, and also prevent set S_k from being chosen whenever any of its members has already been covered. This latter property insures that the sets in the cover are disjoint, as required by Exact Cover.

The following Lemma shows how the edges in A_2 force the desired paths, by giving the relevant properties of Figs. 8A and 8B. Figs. 9A and 9B show all edges of G which are incident with nodes of interest.

Lemma 2.5.1. *Fix i and j, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant t$. Let $x = d_i$ if $j = 1$ and $x = c_{i,j-1}$ otherwise; $x' = e_i$ if $j = t$ and $x' = a_{i,j+1}$ otherwise. Let ω be any partial H-path*



(A) (B)

Fig. 9. Cases for Lemma 2.5.1.

of G satisfying:

(i) *every* $v \in N - \{a_{i,j}, b_{i,j}, c_{i,j}\}$ *adjacent to* $y_{i-1,j}$ *belongs to* ω, *and*

(ii) *None of* $a_{i,j}, b_{i,j}, c_{i,j}$ *belongs to* ω.

Then the following hold:

Case 1. $u_j \notin S_i$ (Fig. 9A)

(1L) *If* $\tau = \omega x a_{i,j}$ *is a partial H-path of G, the only possible* 4*-extensions of* τ *are* $b_{i,j}\, y_{i-1,j}\, c_{i,j}\, x'$ *and* $b_{i,j}\, y_{i,j}\, c_{i,j}\, x'$.

(1R) *If* $\tau = \omega x' c_{i,j}$ *is a partial H-path of G, the only possible* 4*-extensions of* τ *are* $b_{i,j}\, y_{i-1,j}\, a_{i,j}\, x$ *and* $b_{i,j}\, y_{i,j}\, a_{i,j}\, x$.

Case 2. $u_i \in S_i$ (Fig. 9B)

(2L) *Same as* (1L).

(2*R*) *If* $\tau = \omega x' c_{i,j}$ *is a partial H-path of G, the only possible* 3*-extension of* τ *is* $b_{i,j}\, a_{i,j}\, x$.

Proof of Lemma. (Case 1L) Let γ be an *H*-path with prefix τ. The last node of γ must be $a_{n+1,t}$. The only possible 4-extension of τ which has been excluded is $b_{i,j}\, y_{i,j}\, vv'$ where $v \neq c_{i,j}$. But then $c_{i,j}$ would have to be the last node of γ, since properties (i) and (ii) of ω insure that $\langle x', c_{i,j}\rangle$ belongs to γ; that is, $c_{i,j}$ does not yet belong to the *H*-path and the only way that path can reach $c_{i,j}$ is through x'. This contradiction proves (1L).

The other cases follow similarly. □

Using Lemma 2.5.1 and our informal description of the correspondence between *H*-paths and exact covers, the reader should have no difficulty in verifying that *G* has an *H*-path whenever *S* contains an exact cover. To complete the proof of Theorem 2.5, we must show the converse.

Suppose that *G* has an *H*-path γ. Let $T = \{k \mid \langle f_k, e_k\rangle$ belongs to $\gamma\}$. We shall show that $S' = \{S_k \mid k \in T\}$ forms an exact cover for *U*. Define the partial unions $U_0 = \emptyset$ and $U_k = \bigcup_{\substack{1 \leqslant i \leqslant k \\ i \in T}} S_i$, $1 \leqslant k \leqslant n$.

Then we have the following:

Lemma 2.5.2. *For each* k, $1 \leqslant k \leqslant n+1$, *if* $\omega \in N^*$ *and* ωf_k *is a partial H-path of G, then*

(i) *None of* $\{a_{i,j}, b_{i,j}, c_{i,j}, d_i, e_i\colon k \leqslant i \leqslant n,\ 1 \leqslant j \leqslant t\}$ *belong to* ω.

(ii) *All of* $\{a_{i,j}, b_{i,j}, c_{i,j}\colon 1 \leqslant i \leqslant k-1,\ 1 \leqslant j \leqslant t\}$ *belong to* ω.

(iii) *For each* j, $1 \leqslant j \leqslant t$, $y_{k-1,j}$ *belongs to* ω *if and only if* $u_j \notin U_{k-1}$.

(iv) *If there exists* j *such that* $u_j \in U_{k-1} \cap S_k$, *then* d_k *is the only* 1*-extension of* ωf_k.

Proof of Lemma. The proof is by induction. The basis $k = 1$ is immediate since any *H*-path must begin with $a_{0,1}\, y_{0,1}\, a_{0,2}\, y_{0,2} \cdots a_{0,t}\, y_{0,t}\, f_1$. The induction step follows almost entirely by Lemma 2.5.1. The only other possibility is that ωf_k might be extended by $d_k f_{k+1}$ (or by $e_k f_{k+1}$). But then an argument similar to the proof of Lemma 2.5.1 shows that e_k (respectively d_k) must be the last node of the H-path, which is impossible. □

It is now clear that $\{S_k | k \in T\}$ is an exact cover for U. Since any H-path must have the form $\omega f_{n+1}\, y_{n,1}\, a_{n+1,1}\, y_{n,2}\, a_{n+1,2} \cdots y_{n,t}\, a_{n+1,t}$, Lemma 2.5.2 (iii) for $k = n+1$ insures that $\bigcup_{k \in T} S_k = U_n = U$, and (iv) insures the disjointness of the sets in the cover. This completes the proof of Theorem 2.5. □

Observe that the directed graph G constructed in the proof need not have arbitrarily large degree, in fact, no node has in-degree exceeding 3 or out-degree exceeding 4. It is not known, however, whether these are the *strongest* possible degree constraints for which Planar Directed Hamiltonian Path remains NP-complete.

Theorem 2.6. Sat3 α *Node Cover With Node Degree At Most* 3.

Proof. Suppose we are given a set $C = \{C_1, C_2, \ldots, C_p\}$ of disjunctive clauses, each containing no more than 3 literals. As in the proof of Theorem 1.1, we may assume that each clause contains exactly 3 literals, possibly with duplication. Index the literals occurring in clause C_h as $a_{h,1}$, $a_{h,2}$, and $a_{h,3}$, $1 \leqslant h \leqslant p$. Let $x_1, x_2, \ldots, x_n$ denote the variables occurring in the p clauses, and for each i, $1 \leqslant i \leqslant n$, let $m(i)$ denote the number of occurrences of variable x_i (as literal x_i or literal $\bar{x}_i$) in the clauses. Arbitrarily index the $m(i)$ occurrences of variable x_i as occurrence 1, occurrence 2, ..., occurrence $m(i)$. We shall construct a graph G, having node degree at most 3, and give an integer $k > 0$, such that C is satisfiable if and only if G has a node cover of size k.

We describe the graph $G = (N, A)$ in several steps. First, for each variable x_i, we have a subgraph $H_i = (N_i, A_i)$, a simple circuit with $|N_i| = |A_i| = 2m(i)$, as shown in Fig. 10. Observe the alternate labelling of the nodes.

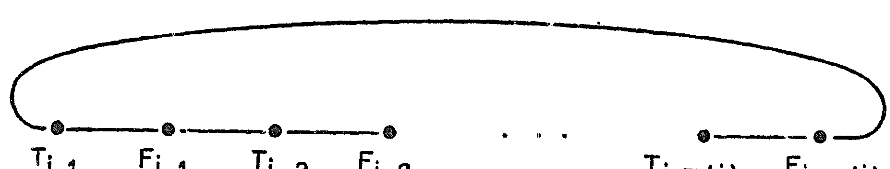


Fig. 10. A subgraph H_i.

For each clause C_h, we have a subgraph $H'_h = (N'_h, A'_h)$ where $N'_h = \{V_{h,1}, V_{h,2}, V_{h,3}\}$ and A'_h consists of an edge joining each pair of nodes belonging to N'_h. The remaining edges, which each join a node from some set N_i to a node from some set N'_h, are as follows:

$$B_1 = \{\{T_{i,j}, V_{s,t}\} : a_{s,t} = x_i \text{ is the } j^{\text{th}} \text{ occurrence of variable } x_i \text{ in } C\}$$

and

$$B_2 = \{\{F_{i,j}, V_{s,t}\}: a_{s,t} = \bar{x}_i \text{ is the } j^{\text{th}} \text{ occurrence of variable } x_i \text{ in } C\}.$$

The graph $G = (N, A)$ is defined by:

$$N = \bigcup_{i=1}^{n} N_i \cup \bigcup_{h=1}^{p} N'_h,$$

$$A = \bigcup_{i=1}^{n} A_i \cup \bigcup_{h=1}^{p} A'_h \cup B_1 \cup B_2.$$

Observe that every node of G has degree at most 3. We show that the set C of clauses is satisfiable if and only if G has a node cover of size $5p$.

The following properties of the subgraphs H_i are easy to verify:

(2.6A) There exists a node cover for H_i containing $m(i)$ nodes, including all nodes $T_{i,j}$ and no nodes $F_{i,j}$. There also exists such a node cover which includes all nodes $F_{i,j}$ and no nodes $T_{i,j}$.

(2.6B) No node cover for H_i contains fewer than $m(i)$ nodes, and every node cover for H_i which includes both a node $F_{i,j}$ and a node $T_{i,l}$ must contain more than $m(i)$ nodes.

Now, suppose we are given a truth assignment to the n variables which satisfies the set C of clauses. The corresponding node cover S contains the following nodes:

(i) For each variable x_i which is set "true", the cover of $m(i)$ nodes for H_i which includes all $T_{i,j}$.

(ii) For each variable x_i which is set "false", the cover of $m(i)$ nodes for H_i which includes all $F_{i,j}$.

(iii) For each clause C_h, all the nodes of N'_h except some one of them $V_{h,j}$ such that literal $a_{h,j}$ is true for this truth assignment (at least one such literal exists since the clause is satisfied).

Clearly, these nodes cover all of the edges belonging to the sets A_i, $1 \leqslant i \leqslant n$, and A'_h, $1 \leqslant h \leqslant p$. Each edge $\{T_{i,j}, V_{s,t}\}$ in B_1 is also covered since either $V_{s,t}$ belongs to S or $a_{s,t} = x_i$ is true and $T_{i,j}$ belongs to S. Similarly, each edge in B_2 is covered by S. Thus, S is a node cover. Furthermore, the number of nodes in S is

$$2p + \sum_{i=1}^{n} (m(i)) = 2p + 3p = 5p,$$

as required.

Conversely, suppose we have a node cover S for C such that $|S| = 5p$. S must contain at least two nodes from each N'_h in order to cover the edges in A'_h, for a total of at least $2p$ such nodes. Similarly, by (2.6B) S must contain at least $\sum_{i=1}^{n} (m(i)) = 3p$ nodes from the N_i. Hence S must contain *exactly* 2 nodes from each N'_h, and *exactly* $m(i)$ nodes from each N_i. Thus by (2.6A) and (2.6B), S must contain, from each N_i, either all the $T_{i,j}$ nodes and none of the $F_{i,j}$ nodes, or all the $F_{i,j}$ nodes and none of the $T_{i,j}$ nodes. A consistent truth assignment can be obtained by setting x_i "true" if S contains all the $T_{i,j}$ nodes, and setting x_i "false" otherwise. The reader may verify that, because of the edges in B_1 and B_2, this truth setting satisfies the set C of clauses. □

Remark. Theorem 2.6 was obtained independently by Peter Herrmann [10].

Theorem 2.7. *Node Cover* α *Planar Node Cover.*

Proof. The key structure used in this proof is the graph H pictured in Fig. 11, which analogously to Theorem 2.2 will be called a *crossover*, with *outlets* v_1, v_1', v_2, and v_2', as labelled.

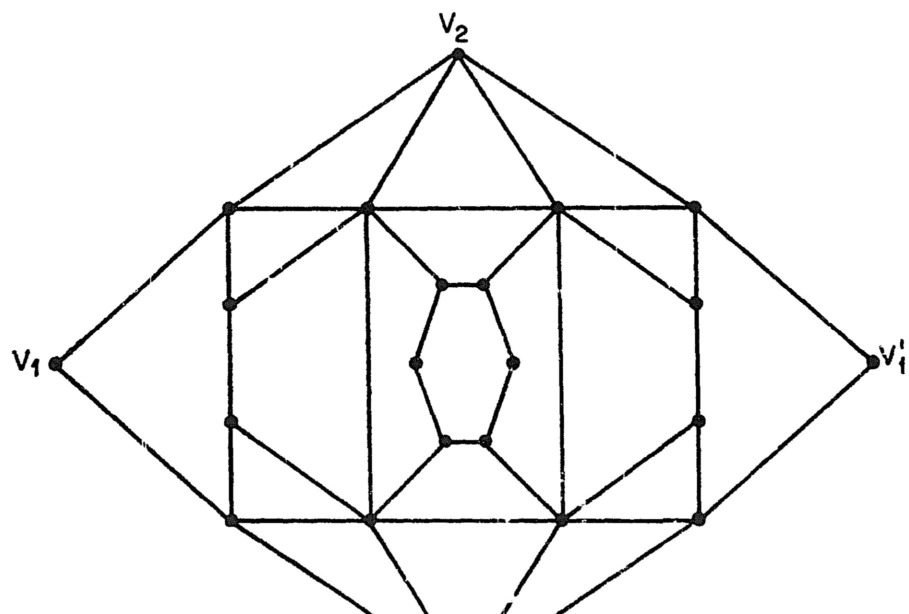


Fig. 11. Crossover H for Theorem 2.7.

Now for each i, j, $0 \leq i, j \leq 2$, let $c[i,j]$ be the minimum cardinality for all node covers C of H obeying

$$|\{V_1, V_1'\} \cap C| = i \quad \text{and} \quad |\{V_2, V_2'\} \cap C| = j.$$

Observe that, by symmetry, when i or j equals 1, the value of $c[i,j]$ is independent of *which* element of the corresponding pair is in C. Table 1 gives the values of $c[i,j]$. We leave to the reader the straightforward but tedious verification of the entries.

From Table 1, we observe that the following properties hold:

(2.7A) For $0 \leq l \leq 2$, $c[1,l]-c[0,l] \leq 1$ and $c[l,1]-c[l,0] \leq 0$.

(2.7B) For $0 \leq l \leq 2$, $c[2,l]-c[1,l] = c[l,2]-c[l,1] = 1$.

Table 1. Values of $c[i, j]$

j \ i	0	1	2
0	13	14	15
1	13	13	14
2	14	14	15

Given a graph $G = (N, A)$ we construct a planar graph $G' = (N', A')$ using these crossovers as follows:

(i) Embed G in the plane, allowing edges to cross each other as in Theorem 2.2.

(ii) Replace each crossing by a copy of H, as shown in Fig. 12.

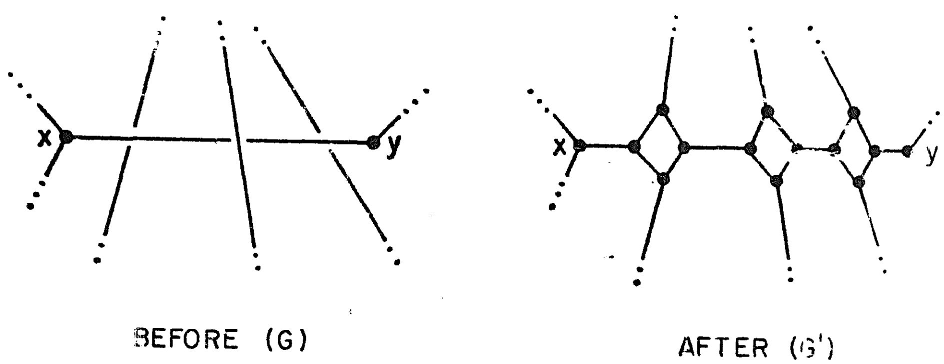


Fig. 12. Construction of G' (only outlets of crossovers shown).

The crossovers which replace crossings on the edge $\{x, y\}$ will be called *crossovers on the* $\{x, y\}$*-line*. The edges connecting these crossovers to each other and to x and y will be called *edges on the* $\{x, y\}$*-line*. The endpoints of these edges will be called the *nodes on the* $\{x, y\}$*-line*. Such nodes which are also crossover outlets will be called the $\{x, y\}$*-outlets* of their crossover. The one which is nearest x will be the crossover's y *outlet*, the one nearest y its x *outlet*, for each crossover on the $\{x, y\}$-line.

Let d be the number of copies of H used in constructing G'. Observe that the edges of G' can be partitioned into two sets: *line edges*, those which are on the $\{x, y\}$-line for some $\{x, y\} \in A$, and *crossover edges*, those which are part of one of the d crossovers. All the edges on the $\{x, y\}$-line can be covered by taking either x and all the x-outlets of crossovers on the line, or y and all the y outlets. The edges in the crossovers can only be covered by crossover nodes.

Now, since in G' each edge-crossing of the planar representation of G has been replaced by a planar graph which itself contains no crossings, G' is planar. Moreover, the size of G' is clearly at most a polynomial in the size of G. The proof of the theorem will thus be concluded by showing that, for any k, G has a node cover of size k if and only if G' has a node cover of size $k+13d$.

Suppose there is a node cover S of $G = (N, A)$ with $|S| = k$. We construct a node cover of G' from S as follows. For each edge $\{x, y\} \in A$, let $f(x, y)$ be an endpoint of that edge which is in S. Then define

$$S' = \{v\colon v \text{ is the } f(x, y) \text{ outlet for a crossover on the } \{x, y\}\text{-line for some } \{x, y\} \in A\}.$$

Since S is a node cover for G, f is defined for all edges in A, and so $S \cup S'$ covers all the line edges of G'. Moreover, since each crossover is on two lines in G', S' contains exactly two outlets for each crossover, one from each outlet pair, and so $|S'| = 2d$. All that remains is to cover the as yet uncovered crossover edges. For these, observe from Table 1 that any set which contains two nodes from a crossover, one from each outlet pair, can be extended, by adding 11 of the crossover's internal nodes, to form a node cover of the crossover made up of $c[1, 1] = 13$ nodes. Let S'' be the set containing, for each crossover, the 11 additional nodes needed to extend S' to a node cover for that crossover. Thus $S \cup S' \cup S''$ is a node cover of G' having $k+2d+11d = k+13d$ nodes.

Conversely, suppose there is a node cover of G' having $k+13d$ nodes. Let

$$k^* = \min\{|S|\colon S \text{ is a node cover of } G'\}, \text{ and}$$

$$M = \{S\colon S \text{ is a node cover of } G' \text{ and } |S| = k^*\}.$$

For each $S \in M$, define

$$m(S) = |\{x \in S\colon x \text{ is an outlet node for some crossover in } G'\}|,$$

$$m^* = \min\{m(S)\colon S \in M\},$$

and let $S^* \in M$ be some node cover with $m(S^*) = m^*$. Since S^* must contain 13 nodes from each of the crossovers in order for it to cover all the crossover edges (see Table 1), we know that $|S^* \cap N| \leqslant k$. We conclude our proof by showing that $S' = S^* \cap N$ is a node cover of G.

Suppose it is not. Then there exists some $\{x, y\} \in A$ such that $S' \cap \{x, y\} = \emptyset$ and hence $S^* \cap \{x, y\} = \emptyset$. Let the number of crossovers on the $\{x, y\}$-line be l. Then there are $l+1$ edges on the $\{x, y\}$-line, and hence at least $l+1$ of the nodes on the line must be in S^*, and since neither x nor y is, all $l+1$ must be outlets. If we let $n(i)$ be the number of crossovers on the $\{x, y\}$-line with i of their $\{x, y\}$-outlets in S^*, we thus have $n(2)-n(0) \geqslant 1$. We shall show that this leads to a contradiction.

Let X_i be the set of nodes in the ith crossover on the $\{x, y\}$-line, $1 \leqslant i \leqslant l$, and let $S_i = X_i \cap S^*$. Let $T_i \subseteq X_i$ be a node cover of the crossover containing its x outlet (but not the y outlet) and the same *non* $\{x, y\}$-outlets as does S_i, and having minimum cardinality for such node covers. For each i, let $r(i)$ be the number of $\{x, y\}$-outlets of the ith crossover which are in S^*. Then we have, by (2.7A) and (2.7B)

$$r(i) = 0 \text{ implies } |T_i| \leqslant |S_i|+1,$$

$$r(i) = 1 \text{ implies } |T_i| \leqslant |S_i|,$$

$$r(i) = 2 \text{ implies } |T_i| \leqslant |S_i|-1.$$

Let $T = \bigcup_{i=1}^{l} T_i$, $S = \bigcup_{i=1}^{l} S_i$. Since $n(2) - n(0) \geq 1$ we have by the above that

$$|T| \leqslant |S| - 1.$$

Moreover, T contains at least one fewer $\{x, y\}$-outlet than does S, and exactly the same number of non $\{x, y\}$-outlets. Furthermore, $T \cup \{x\}$ will cover all the line edges that S did, and so $T^* = (S^* - S) \cup T \cup \{x\}$ is a node cover of G' with

$$|T^*| = |S^*| - |S| + |T| + 1 \leqslant |S^*| = k^*, \text{ and}$$

$$m(T^*) = m(S^*) - 1 = m^* - 1,$$

contradicting the definition of m^*. Thus $S^* \cup N$ *is* a node cover of G and the theorem is proved. □

Notice that, if the graph G given as input for Node Cover has no node degree exceeding 3, the graph G' constructed in the proof as input for Planar Node Cover will have no node degree exceeding 6. This implies that Planar Node Cover With Node Degree At Most 6 is *NP*-complete. It is not known whether this degree bound is best possible.

3. Concluding remarks

We have seen that a number of graph-theoretic *NP*-complete problems remain *NP*-complete when the structure of the allowed inputs is substantially restricted. Similar questions can be asked for restricted versions of other *NP*-complete problems. For example, it is not yet known whether Steiner Tree [13] for planar graphs or multiprocessor scheduling with 3 processors, unit time tasks, and an arbitrary partial order [19] are *NP*-complete. The open status of Undirected Hamiltonian Circuit for planar graphs has been mentioned previously. The question of whether Max Cut with restricted node degree is *NP*-complete also remains open.

In examining such problems, it is important to keep in mind that two types of results are possible. Not only is it important to find simple subcases which are still *NP*-complete, but it is also important to find large subdomains for which the problem can be solved in polynomial time. We have given one example in this latter direction, the case of Max Cut for planar graphs. Other recent papers [5, 7, 8] have shown that Clique and Chromatic Number can be solved in polynomial time for "transitively orientable" graphs, "chordal" graphs, and "circle" graphs.

Both types of results should prove useful to designers of practical combinatorial algorithms.

Acknowledgment

The authors are pleased to thank Donald Knuth for pointing out a correction to our original proof of Theorem 1.2, Shimon Even for raising the question of the status of Optimal Linear Arrangement, and David Weinberger for helpful comments on our original manuscript.

Note added in proof. The Hamiltonian path and circuit problems mentioned as open in Section 2 have recently been proven *NP*-complete by R. E. Tarjan and the first two authors.

APPENDIX

Definitions of *NP*-Complete Problems

Satisfiability With At Most 3 Literals Per Clause (Sat3) [13]

Input: Set of clauses $C = \{C_1, C_2, ..., C_p\}$ in variables $x_1, x_2, ..., x_n$, each clause being the disjunction of 3 or fewer literals, where a literal is either a variable x_i or its negation $\bar{x}_i$.

Property: There is a truth assignment to the variables which simultaneously satisfies all the clauses in C (a clause is satisfied if any one of its disjuncts is x_i for some "true" variable x_i, or $\bar{x}_j$ for some "false" x_j).

Clique [13]

Input: Graph $G = (N, A)$, positive integer k.

Property: G has a clique of size greater than or equal to k, i.e., a set $N' \subseteq N$ with $|N'| \geqslant k$ and such that for all $n_1, n_2 \in N'$, $\{n_1, n_2\} \in A$.

Exact Cover [13]

Input: Collection of sets $S = \{S_1, S_2, ..., S_n\}$.

Property: S has an exact cover, i.e., a subcollection $S' \subseteq S$ such that $\bigcup_{S_i \in S'} S_i = \bigcup_{i=1}^{n} S_i$, and for all $S_i, S_j \in S'$, $S_i \cap S_j = \emptyset$.

Graph k-Colorability [13]

Input: Graph $G = (N, A)$.

Property: G has a legal k-coloring of its nodes, i.e., there is a map $f: N \to \{1, 2, ..., k\}$ such that $\{n_1, n_2\} \in A$ implies $f(n_1) \neq f(n_2)$.

Undirected (Directed) Hamiltonian Path (Circuit) [13]

Input: Graph $G = (N, A)$. (Directed graph $G = (N, A)$).

Property: G has a Hamiltonian path (circuit), i.e., an ordering of the nodes $N = \{n_1, n_2, ..., n_{|N|}\}$ such that for $1 \leqslant i \leqslant |N|$, $\{n_i, n_{i+1}\} \in A$ $(\langle n_i, n_{i+1}\rangle \in A)$, and (in the circuit case only) $\{n_{|N|}, n_1\} \in A$ ($\langle n_{|N|}, n_1 \rangle \in A$).

Node Cover [13]

Input: Graph $G = (N, A)$, positive integer k.

Property: G has a node cover of size less than or equal to k, i.e., a subset $N' \subseteq N$ with $|N'| \leqslant k$ and such that for all $\{x, y\} \in A$, $\{x, y\} \cap N' \neq \emptyset$.

References

[1] D. Adolphson and T. C. Hu, Optimal linear ordering, SIAM J. Appl. Math. 25 (1973) 403–423.

[2] R. L. Brooks, On coloring the nodes of a network, Proc. Cambridge Philos. Soc. 37 (1941) 194–197.

[3] S. A. Cook, The complexity of theorem-proving procedures, Proceedings of the 4th Annual ACM Symposium on Theory of Computing (1971) 151–158.

[4] J. Edmonds and E. L. Johnson, Matching: A well-solved class of integer linear programs, Combinatorial Structures and Their Applications, Gordon and Breach, New York, (1970) 89–92.

[5] S. Even, A. Lempel and A. Pnueli, Permutation graphs and transitive graphs, J. ACM 19 (1972) 400–410.

[6] M. R. Garey, R. L. Graham and J. D. Ullman, Worst-case analysis of memory allocation algorithms, STOC (1972) 143–150.

[7] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, SIAM J. Computing 1 (1972) 180–187.

[8] F. Gavril, Algorithms for a maximum clique and a maximum independent set of a circle graph, Networks 3 (1973) 261–273.

[9] R. L. Graham, Bounds on multiprocessing anomalies and related packing algorithms, Proceedings of the AFIPS Spring Joint Computer Conference (1972) 205–217.

[10] P. Herrmann, Reducibility among combinatorial problems, Masters Thesis, Massachusetts Institute of Technology (1973).

[11] D. S. Johnson, Fast algorithms for bin packing, J. Compt. System Sci. 8 (1974) 272–314.

[12] D. S. Johnson, Approximation algorithms for combinatorial problems, J. Compt. System Sci, 9(1974) 256–178.

[13] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Millera nd J. W. Thatcher, eds., Complexity of Computer Computations, (Plenum Press, New York, (1972) 85-104.

[14] G. I. Orlova and Y. G. Dorfman, Finding the maximum cut in a graph, Engrg. Cybernetics 10 (1972) 502–506.

[15] S. Sahni, Some related problems from network flows, game theory, and integer programming, Proceedings of 13th Annual IEEE Symposium on Switching and Automata Theory (1972) 130–138.

[16] S. Sahni, On the knapsack and other computationally related problems, Doctoral Thesis, Cornell University (1973).

[17] R. Sethi, Complete register allocation problems, Proceedings of the 5th Annual ACM Symposium on Theory of Computing (1973) 182–195.

[18] L. Stockmeyer, Planar 3-colorability is polynomial complete, SIGACT News 5, No. 3 (1973) 19–25.

[19] J. D. Ullman, Polynomial complete scheduling problems, 4th Symp. on Operating System Principles (1973) 96–101.