JOURNAL OF ALGORITHMS **20,** 137–156 (1996)
ARTICLE NO. 0006

Recognizing Greedy Structures

Yair Caro

Department of Mathematics, School of Education, University of Haifa–Oranim, Tivon 36910, Israel

András Sebő

CNRS, ARTEMIS, Universite Fourier d Grenoble, France

and

Michael Tarsi

Computer Science Department, School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel

Received July 30, 1993; accepted November 2, 1994

We study decision problems of the following form: Given an instance of a combinatorial problem, can it be solved by a greedy algorithm? We present algorithms for the recognition of greedy instances of certain problems, structural characterization of such instances for other problems, and proofs of *NP*-hardness of the recognition problem for some other cases. Previous results of this type are also stated and reviewed. © 1996 Academic Press, Inc.

1. INTRODUCTION

We study in this article decision problems of the following type: *Given an instance of a combinatorial problem*, *can it be solved by a greedy algorithm*? If the answer is positive, we say that the instance at hand is *greedy*. We present efficient algorithms for the recognition of greedy instances of certain combinatorial problems, structural characterization of such instances for other problems, proofs of *NP*-hardness of the recognition problem for other cases, as well as some open questions within this scope. The precise meaning of ‘‘greedy algorithm’’ varies according to the combinatorial problem at hand. It is always based, however, upon some ‘‘best fit’’ or ‘‘any which fits’’ approach and it avoids backtracking.

We focus here on a more rigorously defined, yet not as general a concept, that of ‘‘greedy hereditary systems’’:

A *hereditary system* is a pair $H = (S, \mathscr{F})$, where S is a finite set and $\mathscr{F}$ is a family of subsets of S, where $F \in \mathscr{F}$ and $F' \subseteq F$ implies $F' \in \mathscr{F}$. The members of $\mathscr{F}$ are called the *feasible* sets of the system.

Many combinatorial problems deal with finding maximum feasible sets in certain classes of hereditary systems. The *greedy algorithm* for the construction of a maximal feasible set M of a hereditary system $H = (S, F)$ is defined as follows:

```
begin
  let $M$ be the empty set;
    repeat
          select any $s \in S \setminus M$, such that $M \bigcup \{x\}$ is feasible and let $M$
        become $M \bigcup \{x\}$;
    until such $x$ does not exist;
  return $M$
end.
```

If a weight function is defined on S and a maximum weight feasible set is required (though, not necessarily achieved), then the selected element x should be of maximum possible weight.

Matroids form the family of hereditary systems for which the greedy algorithm indeed provides a maximum weight feasible set for every positive weight function. We study here systems for which the above holds for the ‘‘all 1’’ function:

A *greedy hereditary systems* (*greedy-HS*) is a hereditary system for which the greedy algorithm always produces a *maximum cardinality* feasible set. Equivalently, a hereditary system is greedy if and only if its maximal (with respect to set containment) feasible sets are all of the same cardinality.

There are doubly exponentially many distinct hereditary systems on an underlying set of cardinality n. Thus, the description of a general hereditary system is exponentially long in n. Any complexity analysis highly depends on the specific encoding scheme by which the input is described; e.g., the hereditary system on the vertex set of a graph, whose feasible sets are the stable sets (see the next section), can be efficiently encoded by a description of the graph. On the other hand, an explicit list of all feasible sets of the same system, can be exponentially longer. An important role is hence played by the combinatorial frame in which the hereditary system is described.

In order to verify that a given hereditary system is nongreedy, one should present two maximal feasible sets which differs in size. It implies that, if testing feasibility and maximality of a set (of a system induced by some combinatorial problem) is in *NP*, then the corresponding recognition of a greedy instance would be in *co-NP*. Among known *NPC* problems, we will show some for which recognizing a greedy instance is *co-NPC*, as well as some others where greedy instances can be recognized in polynomial time. We have found no ''natural'' (whatever this last term means) problem, solvable in polynomial time, for which the recognition of a greedy instance is *NP*-hard. It is possible, however, to artificially construct such a problem:

Let $k \geq 4$ be an integer and $G = (V, E)$ a graph on at least $k + 1$ vertices, which contains two vertex disjoint edges e_1 and e_2. Consider the hereditary system on V which consists of all subsets of cardinality k or less and those of cardinality $k + 1$, which do not contain a k vertices stable subset. (A stable set in a graph is a set of vertices with no edge between any two of its members.) A maximum feasible set in such a system is always of size $k + 1$ and it can be constructed in polynomial time by selecting any set of $k + 1$ vertices which includes the endvertices of e_1 and e_2. On the other hand, a k vertices subset is maximal feasible if and only if it is stable. It turns out that an input (k, G) is greedy if and only if G has no stable set of size k. Telling if there exists a given sized stable set in a graph is a well known *NPC* problem (e.g., [9]). Recognizing a greedy instance of the above is hence *co-NPC*.

In what follows we deal with the recognition of greedy-*HS* among some classes of hereditary systems, induced by certain known combinatorial problems.

While studying the subject, we have encountered many results of this nature in the literature. Our notion of ''greedy'' is not commonly used. In some articles it is replaced by ''random'' ([1–3, 19, 23] and others). Several authors use ad-hoc terminology, such as ''well-covered graphs'' in [16, 4, 18], dealing with the maximum stable set problem and ''equimatchable graphs'' in [13], referring to the maximum matching problem. We find ''greedy'' appropriate for the task, while the term ''random'' is heavily loaded with other mathematical connotation. It is worth mentioning two recent articles [11, 25]. Both deal with some greedy schemes to certain linear programming problems. Despite the similar titles, the content of these papers does not overlap with that of ours. Each of the following sections is devoted to a certain family of combinatorial problems. We also consider some cases which do not fit to the framework of hereditary systems.

2. THE MAXIMUM STABLE SET PROBLEM AND ITS DERIVATIVES

2.1. *Maximum Stable Sets*

A stable set (also known as "independent") in a graph is a set of vertices with no edge between any two of its members. Finding a maximum size stable set (more precisely, telling if there exists one of a given size) is a well known *NPC* problem, e.g., [9]. The stable sets of a graph are clearly the feasible sets of a hereditary system. Accordingly, a graph is *maximum stable greedy* (*MS-greedy*) if its maximal stable sets are all of the same cardinality. Let the problem of telling whether a given graph is *MS*-greedy be denoted by the abbreviation *g-MS* (We assume that the size of the input is polynomial in the number of vertices of the input graph. This convention holds throughout this paper, whenever an input graph is considered). We prove now that the recognition of *MS*-greedy graphs is *co-NP*-complete, even when restricted to $K_{1,4}$-free graphs. This particular restriction is, in a way, best possible, as stated later in Theorem 2.7.

THEOREM 2.1. *g-MS is co-NP-complete. This remains true when restricted to graphs with no* $K_{1,4}$ *induced subgraph.*

Proof. First let us restate the theorem in complementary terminology: The following decision problem is *NP*-complete:

Input: A graph $G = (V, E)$
Question: Are there two maximal stable sets $A, B \subseteq V$ with $|A| \neq |B|$?

The problem is obviously in *NP*. We show *NP*-hardness by providing a polynomial time reduction of the 3-dimensional matching problem (3-*DM*), defined as follows:

Input: Three disjoint sets X, Y, Z of the some finite cardinality k and a collection $\mathscr{T}$ of three element sets, where each member of $\mathscr{F}$ includes exactly one element from each of X, Y, and Z.
Question: Is there a subset M of pairwise disjoint members of $\mathscr{T}$, whose union is $X \cup Y \cup Z$?

3-*DM* is well known to be in *NPC* (see [9]). Let $I = (X, Y, Z, \mathscr{T})$ be an instance of 3-*DM*, where $X = \{x_1, \dots, x_k\}$ and $Y = \{y_1, \dots, y_k\}$. Let t be a new element, $t \notin X \cup Y \cup Z$. Define $\mathscr{T}_x = \{\{t, x_i\} | i = 1, \dots, k\}$ and $\mathscr{T}_y = \{\{y_i\} | i = 1, \dots, k\}$. Let $G(I)$ be the intersection graph of the subset system $\mathscr{T} \cup \mathscr{T}_x \cup \mathscr{T}_y$, that is, the vertex set is $\mathscr{T} \cup \mathscr{T}_x \cup \mathscr{T}_y$, and two are adjacent if and only if the corresponding subsets have a nonempty intersection. Since $|A| \leq 3$ for each $A \in \mathscr{T} \cup \mathscr{T}_x \cup \mathscr{T}_y$, any four sets, each intersecting with A, cannot be pairwise disjoint, hence, $G(I)$ does not admit a $K_{1,4}$ induced subgraph. A stable set of $G(I)$ corresponds to a collection of pairwise disjoint members of $\mathscr{T} \cup \mathscr{T}_x \cup \mathscr{T}_y$. A maximal such collection always includes k disjoint subsets which completely cover Y (otherwise, some $\{y_i\} \in \mathscr{T}_y$ can be appended). If X is not completely covered too, then one member of $\mathscr{T}_x$ can also be added to form a maximal stable set of $G(I)$ of size $k + 1$. A smaller maximal stable set can be formed only by k pairwise disjoint members of $\mathscr{T}$, which completely cover both X and Y, namely, by a complete 3-dimensional matching on the instance I. Any $k + 2$ members of $\mathscr{T} \cup \mathscr{T}_x \cup \mathscr{T}_y$ obviously contain an intersecting pair. Thus, $G(I)$ has maximal stable sets of different size (k and $k + 1$) if and only if the instance I is solvable for 3-*DM*. ■

Note that the result remains valid even if it is a priori known that the cardinality of a maximal stable set is either k or $k + 1$.

Upon completion of this paper a recent work of Plummer [16] was brought to our attention. It contains references to two other proofs for co-*NP*-completeness of *g*-*MS*: one by Chvátal and Slater [4] and another by Sankaranarayana and Stewart [18]. As far as we understand, these proofs hold no restriction on the input graph.

We proceed with studying the ‘‘greedy question’’ on some combinatorial problems which can be viewed as restricted cases of the max stable set problem.

2.2. *Hypergraph Matching*

For further discussion we use hypergraph terminology, which allows uniform formulation for set system problems and their graph analogues.

A *hypergraph* is a pair $H = (V, E)$ where V is a finite set of *vertices* and E a collection of subsets of V, called (*hyper*)*edges*. Let $H = (V, E)$ be a hypergraph and V' a subset of V. The hypergraph (V', E'), where $E' \subseteq E$ is the set of all edges of H contained in V', is called the subhypergraph of H *induced* by V'. If H' is the subhypergraph of $H = (V, E)$ induced by $T \subseteq V$ then $H \setminus H'$, or $H \setminus T$, denote the subhypergraph of H induced by $V \setminus T$. A hypergraph is *k-uniform* if every edge includes exactly k vertices.

We assume that a hypergraph is described (as input for combinatorial problems) by an explicit listing of its edges.

A *matching* M in a hypergraph $H = (V, E)$ is a subset of $|E|$ whose members are pairwise disjoint. The set of all matchings of a hypergraph clearly forms a hereditary system. Maximum matching of a hypergraph (when restated as a decision problem) contains 3-*DM*, as well as many other *NPC* problems. A greedy instance of the maximum hypergraph matching problem is a hypergraph with all maximal matching having the same cardinality. Such hypergraphs will be called greedy. Let the recognition problem of greedy hypergraphs be denoted by *g-MMH*. An immediate corollary of Theorem 2.1 is a similar result for *g-MMH*.

THEOREM 2.2. *g-MMH is co-NPC. This remains true for* 3*-uniform hypergraphs*.

Proof. In the proof of Theorem 2.1, the complement of *g-MMH* on the 3-uniform hypergraph, whose edge set if $\mathscr{T} \cup \mathscr{T}_x \cup \mathscr{T}_y$, is clearly equivalent to 3-*DM* on the input I. To obtain 3-uniformity insert a new vertex into each set in $\mathscr{T}_x$ and two new ones into each of $\mathscr{T}_y$. ∎

2.3. *Perfect Hypergraph Matching*

A *perfect matching* in a hypergraph $H = (V, E)$ is a matching M of H with $\bigcup_{A \in M} A = V$. We call a hypergraph which admits a perfect matching *matchable*. A hypergraph is *greedily matchable* (*g-matchable*) if it is matchable and greedy.

A *minimal nongreedy matchable hypergraph* (*mng-matchable*) is a matchable hypergraph H, which is not g-matchable and every matchable induced subhypergraph of H, except H itself, is g-matchable.

mng-matchable hypergraphs play the main role in our characterization of g-matchable ones, due to the following simple observation:

PROPOSITION 2.1. *A matchable hypergraph is g-matchable if and only if all its matchable induced sub-hypergraphs are g-matchable*.

Proof. Let $H = (V, E)$ be a g-matchable hypergraph and H' a matchable induced sub-hypergraph of H. Since H is g-matchable, a perfect matching of H' can be greedily expanded into a perfect matching of H, which means that $H \setminus H'$ is g-matchable. The same holds for H' by switching the roles of H' and $H \setminus H'$. The ‘‘if’’ part follows the convention that H is a subhypergraph of itself. ∎

Consequently, a matchable hypergraph is not g-matchable if and only if it contains an *mng*-matchable induced subhypergraph. Following is a basic structural property of minimal nongreedy matchable hypergraphs:

LEMMA 2.1. *If $H = (V, E)$ is mng-matchable then there exists an edge $X \in E$, which intersects with every edge of any perfect matching of H.*

Proof.

CLAIM 1. *Let $H = (V, E)$ be an mng-matchable hypergraph, then there exists an edge X in E such that $H \setminus X$ is not matchable.*

Indeed, let M be a maximal matching of H which is not perfect and X any edge in M. If $H \setminus X$ is matchable, then $M \setminus \{X\}$ is contained in a perfect matching of $H \setminus X$. Adding the edge X, such a matching becomes a perfect *matching of H which contains M. Claim* 1 *is implied by this contradiction. To complete the proof of Lemma* 2.1, *let M be any perfect matching of H and X as stated in Claim* 1. *Assume that an edge $A \in M$ is disjoint from X. $M \setminus \{A\}$ is a perfect matching of $H \setminus A$ and, by minimality of H, $H \setminus A$ is g-matchable. Thus, X is included in a perfect matching M' of $H \setminus A$. But then, $(M' \setminus \{X\}) \cup \{A\}$ is a perfect matching of $H \setminus X$, in contradiction to Claim* 1. ■

Let g-PMH denote the recognition problem of a g-matchable hypergraph.

We do not know the complexity status of *g-PMH*. However, Lemma 2.1 implies the existence of a *g-PMH* algorithm, whose time complexity, although exponential in the maximum size of an edge, is polynomial in all other parameters of the input. Note that this already shows *g-PMH* to be easier (provided $P \neq NP$) than *g-MMH*, which is, as stated in Theorem 2.2, co-*NPC* even when restricted to 3-uniform hypergraphs:

Let us define *g-k-PMH* to be the restriction of *g-PMH* to hypergraphs with edge size bounded by the constant k.

THEOREM 2.3. *g-k-PMH is solvable in $O(n^{k^2})$ time for every $k \in N$*

Proof. A greedy construction of a maximal matching, which fails to produce a perfect matching, supplies a negative answer to *g-k-PMH*. It then suffices to give an algorithm which solves the problem among matchable hypergraphs. By Proposition 2.1, a matchable hypergraph is not g-matchable if and only if it has an *mng*-matchable induced subhypergraph. By Lemma 2.1, an *mng*-matchable hypergraph whose edge size is bounded by k has at most k^2 vertices. With k fixed, there are finitely many (nonisomorphic) such hypergraphs. The number of induced subhypergraphs of this size, in a hypergraph H on n vertices, is of order $O(n^{(k^2)})$ and hence they can all be listed and checked in polynomial time. ■

Theorem 2.3 applies to combinatorial problems which can be interpreted as a search for bounded edge size perfect matchings:

2.4. *Graph Decomposition and Factorization*

An *H-decomposition* of a graph G is a partition of its edge set into subgraphs, isomorphic to the graph H. It is clearly equivalent to a perfect matching in the uniform hypergraph on the edge set of G, whose edges are the edge sets of all H-isomorphic subgraphs of G.

The vertex analogue—*an H-factorization*—is a collection of H-isomorphic subgraphs whose vertex sets form a partition of that of G. Again, it can be viewed as a perfect matching in a uniform hypergraph, whose edges are the vertex sets of all H-isomorphic subgraphs of G. If H is a fixed graph then telling whether an input graph G admits an H-decomposition (or an H factorization) is *NPC* (factorization, for every connected graph H on more than 2 vertices [12]; decomposition, for every graph H which has a connected component of more than 2 edges [6]).

Theorem 2.3 implies that the corresponding problems of recognizing greedy instances are polynomial: Let *g-H-decomposition* (*factorization*) stand for the following decision problem:

Input: A graph G.
Question: Is any collection of edge (vertex) disjoint H-subgraphs of G is contained in an H-decomposition (factorization):

COROLLARY 2.1. *For every fixed graph H, g-H-decomposition and g-H-factorization both are decidable in polynomial time.*

Corollary 2.1 is the main result of [1] our proof of Theorem 2.3 is merely a translation of the proof in [1] to the more general setting of hypergraph perfect matching.

If the graph H is not fixed, but is given as a part of the input then, the obtained problems are co-*NP*-complete, even if H is restricted to be a complete graph.

THEOREM 2.4. *The recognition of greedily K_k-factorizable (decomposable) graphs, where k is a part of the input is co-NP-complete.*

Proof. Let $(G = (V, E), k)$ be an instance of the (*NP*-complete) "clique" problem: *Is there a k vertices clique in G*? Let us construct a graph G' by inserting $k - 1$ distinct new vertices for each vertex x of G, which form a k-clique K_x with the vertex x. If there is no k-clique in G then the only k-cliques of G' are $K_x, x \in V$. In that case G' is clearly greedily K_k-factorazible. On the other hand a k-clique of G, if it exists, cannot be completed to a K_k-factorization of G'.

To obtain the analogous result for decomposition, a new k-clique should be inserted for every *edge* of G, which includes this edge and is otherwise disjoint of G. ■

Notice that the last result does not settle the complexity status of *g-PMH*. Although it deals with a perfect matching in a hypergraph, the hypergraph of k-cliques is encoded by the graph G, which makes the input exponentially shorter, in comparison to an explicit listing of the hyperedges.

2.5. *Vertex Packing*

Closely related to H-factorization is the H-(vertex) packing problem, where the goal is to find a maximum set of vertex disjoint H-isomorphic subgraphs of the input graph G. That is, a maximum matching in that hypergraph where H-factorization searches for a perfect matching. Let *g-H-packing* stand for the recognition problem of graphs for which this hypergraph is greedy. Although *g-H*-factorization is polynomial for any fixed graph H, greedy packing is *co-NPC* even where H is the triangle K_3.

THEOREM 2.5. *g-K_3-packing is co-NPC.*

Proof. An instance $I = (X, Y, Z, \mathscr{T})$ of 3-*DM* is *consistent* if for $x, x_0 \in X$, $y, y_0 \in Y$, and $z, z_0 \in Z$, if $\{x_0, y, z\} \in \mathscr{T}$, $\{x, y_0, z\} \in \mathscr{T}$ and $\{x, y, z_0\} \in \mathscr{T}$, then also $\{x, y, z\} \in \mathscr{T}$. When restricted to consistent instances, 3-*DM* remains *NPC* (see [7, p. 209, exercise 9]). Let $I = (X, Y, Z, \mathscr{T})$ be a consistent instance of 3-*DM*. Let $G_0(I) = (V, E)$ be the *skeleton* graph of I, where $V = X \cup Y \cup Z$ and two vertices are adjacent if and only if there exists a member of $\mathscr{T}$ which includes both. The consistency condition on I means that $\{x, y, z\} \subset V$ induces a triangle in G if and only if $\{x, y, z\} \in \mathscr{T}$. We proceed now along the same lines as in the proof of Theorem 2.1: A graph $G(I)$ is constructed by appending to $G_0(I)$ two sets of triangles, $\mathscr{T}_X = \{\{x, a_x, t\} | x \in X\}$ and $\mathscr{T}_Y = \{\{y, a_y, b_y\} | x \in X\}$, where a_x for every $x \in X$, a_y, b_y for every $y \in Y$ and t are new vertices. As in the proof of Theorem 2.1, $G(I)$ admits maximal K_3 packings of different size (k and $k + 1$, where $k = |X| = |Y| = |Z|$) if and only if there exists a 3-*DM* of I. ■

Modifications of the above provide similar results for other graphs H, e.g., by a proper subdivision of all edges, K_3 can be replaced by any simple circuit C_{3n}. We believe that *co-NP*-completeness holds for every graph H, say, connected and large enough, but have no proof for such a general statement.

2.6. *Maximum Matching and Perfect Matching in Graphs*

A graph $G = (V, E)$ is merely a 2-uniform hypergraph. Our general hypergraph terminology and results are still valid in this restricted setting. Theorem 2.3 implies that greedily matchable graphs can be recognized in polynomial time, by denying the existence of the forbidden induced subgraphs: all graphs on four vertices which contain a three-edge-long path with nonadjacent end vertices (see Lemma 2.1 and [1]). A connected matchable graph with no such subgraphs is clearly either a complete graph of even order, or a complete bipartite graph $K_{n,n}$. This characterization of *g*-matchable graphs is found (along totally different lines) in [19].

Unlike hypergraphs with larger edges, graphs allow maximum matching to be constructed in polynomial time. Based on a maximum matching algorithm, Lesk *et al.* [13] have developed an algorithm to recognize graphs in which all maximal matchings have the same cardinality. They refer to such graphs as *Equimatchable*.

THEOREM 2.6 (Lesk, Plummer, and Pulleyblank). *The restriction of g-MMH to graphs is solvable in polynomial time*.

A matching in a graph G is a stable set in the line graph of G. Thus, Theorem 2.6 equivalently states the existence of a polynomial time algorithm to solve the restriction of *g-MS* to line graphs. Line graphs are characterized by a list of forbidden induced subgraphs, one of which is $K_{1,3}$. Tankus and Tarsi [21] have recently proven the following generalization of Theorem 2.6.

THEOREM 2.7. *The restriction of g-MS to $K_{1,3}$-free graphs is solvable in polynomial time*.

The proof is too involved to be sketched here. It is based, however, on a reduction of the maximum stable set problem on $K_{1,3}$-free graphs to a maximum matching problem, presented in [14]. Two other important tools on which the proof relies are Theorem 2.6 and the existence of a polynomial algorithm to find maximum weight stable set in $K_{1,3}$-free graphs, e.g., the algorithm presented by Minty in [15].

Let us recall Theorem 2.1, which states that the restriction of *g-MS* to $K_{1,4}$-free graphs, (as well as the construction of a maximum stable set in such graphs) is already *NP*-hard.

2.7. *The Structure of Critical Nongreedy Hypergraphs*

Let $H = (V, E)$ be a hypergraph. Let us denote by $\nu = \nu(H)$ the maximum number of disjoint edges in H. Let us say that H is *critical*, if it is not a greedy hypergraph, but $H \setminus X$ is a greedy hypergraph for every $X \in E$. Although the recognition of greedy hypergraphs, even with bounded edge size, is *NP*-hard, the structure of critical hypergraph can be characterized as follows.

THEOREM 2.8. *The hypergraph $H = (V, E)$ is critical if and only if there exists hypergraphs $H_1 = (V_1, E_1), \dots, H_m = (V_m, E_m)$ (m is an integer, $m \geq 2$) so that $\{E_1, \dots, E_m\}$ is a partition of E, and*

(i) *H_i is a greedy hypergraph ($i = 1, \dots, m$);*

(ii) *$K \in H_i$, $L \in H_j$ ($i \neq j$) implies $K \cap L \neq \emptyset$;*

(iii) *there exist $1 \leq i < j \leq m$ such that $\nu(H_i) \neq \nu(H_j)$.*

Proof. We first prove the sufficiency of the conditions. Suppose (i), (ii), and (iii) hold. (ii) implies that a maximal matching of H_t is also maximal in H for all $t = 1, \dots, m$. Let i and j be the numbers defined in (iii). Then the selection of $t = i$ and $t = j$ leads to maximal matchings of different size, because according to (i) the size of a maximal matching is $\nu(H_i)$ in the former case and it is $\nu(H_j)$ in the latter case. By (iii), these two numbers are not equal, and hence H is not greedy. For any edge $X \in E_i$ $H \setminus X$ is, by condition ii, an induced subhypergraph of H_i. It turns out that $H \setminus X$ is greedy and thus H is indeed critical.

To prove the necessity of (i), (ii), and (iii) suppose that H is critical.

CLAIM 1. *For every edge X of H all maximal matchings containing X have the same size, let us denote this size by ν_X. With this notation, there exist edges $X_1, X_2 \in E$ so that $\nu_{X_1} \neq \nu_{X_2}$.*

The first sentence follows immediately from the greediness of $H \setminus X$. The last sentence is true because H is not greedy.

CLAIM 2. *If $X_1, X_2 \in E$, $X_1 \cap X_2 = \emptyset$, then $\nu_{X_1} = \nu_{X_2}$.*

Indeed, if $X_1 \cap X_2 = \emptyset$, then there exists a maximal matching containing both X_1 and X_2. According to Claim 1 the size of this maximal matching is equal to both ν_{X_1} and ν_{X_2}, and the claim is proved.

Let G be the complement of the intersection graph (line-graph) of H. (The vertices of G are the hyperedges of H and two vertices of G are adjacent if the corresponding hyperedges are disjoint.) Let m be the number of components of G, and denote the set of hyperedges corresponding to the vertex-sets of the components by E_i $(i = 1, \dots, m)$, and let $H_i := (V, E_i)$. Clearly, $\{E_1, \dots, E_m\}$ is a partition of E.

CLAIM 3. *H_i $(i = 1, \dots, m)$ is a greedy hypergraph.*

Indeed, since H_i corresponds to a connected subgraph of G, by Claim 2 every maximal matching of H_i has the same size.

To finish the proof of the theorem note that $m \geq 2$, because otherwise H would be greedy by Claim 3. Now (i) is just Claim 3, (ii) holds by the definition of H_i; if (iii) did not hold, then again, H would be a greedy hypergraph: by (ii) all hyperedges of a maximal matching belong to the same E_i, and if (iii) does not hold, then the maximal matchings of all of the H_i's and hence all the maximal matchings of H have the same size. ■

3. 0−1 KNAPSACK (SUBSET SUM)

Among the fundamental *NPC* problems many involve questions about sums of subsets of a given list of integers. Such problems are those defined in [9] as Partition, Knapsack, Bin packing, Subset sum, and more. While considering the length of a numeric input, the integer n is assumed to be defined by $\Theta(\log(n))$ digits. As a representative of such problems, we deal here with 0−1 *Knapsack*.

An instance of 0−1 *Knapsack* is a finite sequence $A = \{a_1, a_2, \dots, a_n\}$ of positive integers and an additional integer b. The question is to tell whether there exists a subsequence of A whose elements' sum equals b. This problem is also known as Subset sum [9, p. 223]. We call an instance of 0−1 *Knapsack* greedy if every subsequence of A whose sum of elements is less than b is contained in a subsequence whose sum of elements is exactly b. Notice that all subsequences whose sums are at most b form a hereditary system. Considering the value of an element as a weight function, an instance is greedy if the greedy algorithm, as it is defined in Section 1 for hereditary systems with a weight function, is guaranteed to produce a maximum weight feasible set. Let g-01*KNAP* denote the problem of recognizing a greedy instance of 0−1 *Knapsack*. Although the underlying problem is *NP*-complete, there exists a polynomial time algorithm for the recognition of a greedy instance.

THEOREM 3.1. *g-01KNAP is solvable in polynomial time.*

Proof. Let $C = (A = \{a_1, \dots, a_n\}, b)$ be an instance of 0−1 *Knapsack*, and suppose $a_1 \leq \dots \leq a_n$.

CLAIM 1. *If C is greedy, then there exists $k \in \{1, \dots, n\}$ such that $\sum_{i=1}^{k} a_i = b$*

Indeed, if C is greedy, then $\sum_{i=1}^{n} a_i \geq b$, and consequently there exists minimal $k \in \{1, \dots, n\}$ such that $b \leq \sum_{i=1}^{k} a_i$. The inequality must be satisfied with equality, because otherwise $a_l + \sum_{i=1}^{k-1} a_i > b$ for every $l \in \{k, \dots, n\}$, in contradiction with the greediness of C.

In what follows, we assume that C satisfies the condition of Claim 1, in particular we fix k to be the index defined in Claim 1.

Let $A_i = \{a_1, a_2, \dots, a_i\}$, $b_j = \sum_{i=1}^{j} a_i$, and define the 0−1 *Knapsack* problems $C_{i,j}$ for all pairs of integers $n \geq i \geq j \geq 1$ as $C_{i,j} = (A_i, b_j)$. (For example, by Claim 1, $C = C_{n,k}$).

CLAIM 2. *$C_{p,q} (n \geq p \geq q)$ is greedy if and only if either $p = q$, or for every nonempty $T \subseteq A_p \setminus A_q$ for which $t = \sum_{a_i \in T} a_i < b$, the following two properties hold*:

(i) *$b_q - t = b_r$ for some $r \in \{1, 2, \dots, q-1\}$;*

(ii) *$C_{q,r}$ is greedy.*

Suppose first that $C_{p,q}$ is greedy. This is of course true if $p = q$ so let us also assume $p > q$. Start the summation with the elements of T. Then it is apparent that $(A_p \setminus T, b_q - t)$ must also be a greedy instance. Since $\sum_{i=1}^{q} a_i = b_q$, when applying the greedy algorithm to this instance, we can always make our selection from A_q, until reaching $b_q - t$. We have proved that $(A_q, b_q - t)$ is greedy. Now (i) is exactly Claim 1 applied to $(A_q, b_q - t)$ and (ii) is the statement we have just proved.

Conversely, to prove the if part, let $B \subseteq A_p$, $\sum_{a_i \in B} a_i \leq b_q$. All we have to prove, is that $\sum_{a_i \in B} a_i$ can be completed to b_q. Define $T = B \setminus A_q$, and $t = \sum_{a_i \in T} a_i$. Since $T \subseteq B, t \leq b_q$ holds, (i) and (ii) can be applied to this particular T and t. According to (i), $b - t = b_r$ for some $r \in \{1, 2, \dots, q\}$, and according to (ii) $B \cap A_q$ can be completed to $b_r = b_q - t$, and Claim 2 is proved.

The above provides a scheme to determine the greediness of an instance $C_{p,q}$ by recursively checking all $C_{q,r}$ descendent from it according to Claim 2. An instance $C_{p,q}$ is a terminal node of such a recursion path if either $p = q$, in which case it is greedy, or if there exists some nonempty $T \subseteq A_p \setminus A_q$ for which $t := \sum_{a_i \in T} a_i < b$ and $b_q - t \neq b_r$ for all $r \in \{1, 2, \dots, q - 1\}$. In that last case $C_{p,q}$ is nongreedy, violating condition (i) of Claim 2. Since $p \geq q$, every recursion path eventually reaches a terminal node. An instance is greedy if and only if all terminal nodes descendent from it are greedy. It remains to translate that scheme into a polynomial algorithm.

Let $C_{p,q}, p > q$ be an instance as defined above. The following procedure either finds that $C_{p,q}$ is a terminal nongreedy node, or generates all the indices r, such that, by Claim 2, the greediness of $C_{p,q}$ is equivalent to that of all $C_{q,r}$:

```
begin
  $S := \{0\}$; $R := \emptyset$;
  for $i := q + 1$ to $p$ do
  begin
    for every $s \in S$ do
      begin
        if $s + a_i < b_q$ then check if there exists $r, 1 \leq r \leq q - 1$ such
        that $b_r = b_q - (s + a_i)$. If such $r$ exists then $R := R \bigcup \{r\}$ and
        $S := S \bigcup \{s + a_i\}$.
        If there is no such $r$ then $C_{p,q}$ is a terminal nongreedy node and
        the procedure halts.
      end
  end
  return the set $R$ of all the indices $r$ as defined above.
end.
```

A similar ''Dynamic programming'' procedure supplies a pseudo-polynomial algorithm for 0−1 *Knapsack* (see, e.g., [9]). The set R contains positive integers all smaller than q. Hence $|R| = |S| = O(n)$ and the time complexity of the entire procedure is $O(n^3)$.

Starting at $C = C_{n,k}$ and applying the above procedure in, say, a depth first order (on $O(n^2)$ instances $C_{i,j}$), the greediness of C is decidable in

$O(n^5)$ time. In fact, for that price, the algorithm provides all integers b for which (A, b) is greedy. ■

4. MAXIMUM AND HAMILTONIAN PATHS AND CIRCUITS

Three basic greedy schemes to construct a Hamiltonian path/circuit are considered in the literature:

(i) *One way progression*: A graph on a vertex set $V = \{v_1, \dots, v_n\}$ is one way greedily Hamiltonian if every simple path $(v_{i_1}, v_{i_2}, \dots, v_{i_k})$ can be completed to a Hamiltonian path $(v_{i_1}, v_{i_2}, \dots, v_{i_k}, v_{i_{k+1}}, \dots, v_{i_n})$.

(ii) *Two ways progression*: A graph is two ways greedily Hamiltonian if every simple path is contained (not necessarily as a prefix) in a Hamiltonian path.

(iii) *Parallel progression*: A graph is parallel greedily Hamiltonian if every linear forest (vertex disjoint union of simple paths) is contained in a Hamiltonian path.

Note that (iii) fits to the general framework of hereditary system, while (i) and (ii) do not. In each of the above, ‘‘Hamiltonian path’’ might be replaced by ‘‘Hamiltonian circuit.’’ Also each applies to either undirected, or directed graphs (with directed-simple paths, linear forests, Hamiltonian paths/circuits). It turns out that we deal here with 12 distinct families of graphs.

Apparently, a considerable amount of work has been carried out, regarding (i) and (ii). A complete characterization of six out of the corresponding eight families can be found in the union of [2, 3, 5, 8, 23, 24]. Different proofs of these results have recently been given by Tankus [20], who also considered and solved (iii). From these characterization results, it follows that the corresponding recognition problems are all solvable in polynomial time. The two cases still open are (ii) and (iii) for directed Hamiltonian path.

When ‘‘Hamiltonian’’ in the classes defined above, is replaced by ‘‘maximum,’’ a new set of problems is obtained, which apply to all graphs, not necessarily Hamiltonian. One of us has recently obtained the following, clearly polynomial, characterization of undirected graphs with the ‘‘two ways greedy maximum path’’ property [22]. We list it here, as a typical example of the results mentioned in this section. Since the case where the graph at hand admits a Hamiltonian path was treated separately in [24], it is not included here.

THEOREM 4.1. *Let G be a connected simple graph which admits no Hamiltonian path. Every simple path of G is contained in a maximum length simple path, if and only if G is one of the following*:

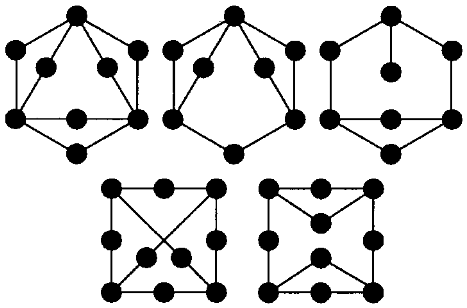

FIG. 1.

The union of simple paths, all of the same length, which share one common endvertex and are otherwise vertex disjoint; or

A bipartite graph on vertex set $V = X \cup Y$, where $|Y| \geq |X| + 2$ and every vertex in X is adjacent to all, but at most one, of the vertices in Y, or one of the five graphs, described in Fig. 1.

Although the characterization is simple and easy to state, the proof of Theorem 4.1 is rather long and involved. Similar phenomena are observed in the other references mentioned above.

5. SATISFIABILITY

An instance of the satisfiability (SAT) problem is a formula in conjunctive normal form (CNF), $V = \bigwedge_{i=1}^{m} c_i$ where a clause c_i is the Boolean sum of literals, each of which is either a variable $x \in X$, or its negation $\bar{x}$ (see, e.g., [9]). C is *greedily satisfiable* if any assignment of truth values to some of the variables, where no clause has all its literals assigned with "false," can be completed to an assignment to all variables in X, which makes $C =$ "true." Although SAT is the old ancestor of all NPC problems, the recognition of its greedy instances is polynomial.

THEOREM 5.1. *An instance $C = \bigwedge_{i=1}^{m} c_i$ of SAT is greedily satisfiable unless there exist two clauses c_i and c_j, which satisfy the following*:

(i) *There exists exactly one variable x whose negated literals are included one in c_i and the other in c_j; and*

(ii) *No clause c_k is contained in $c_i \cup c_j \setminus \{x, \bar{x}\}$ (a clause is referred to as a set of literals).*

Proof. Let c_i, c_j and x satisfy (i) and (ii). Assign values to all the variables involved in $c_i \cup c_j \setminus \{x, \bar{x}\}$, such that all the literals in $c_i \cup c_j \setminus \{x, \bar{x}\}$ are assigned "false." This is possible by (i) and according to (ii) no clause is fully assigned with "false." Once a value is selected also for x, either c_i or c_j will have all its literals assigned with "false" and hence C is not greedily satisfiable.

On the other hand, if C is not greedily satisfiable then there exists a maximal set $X' \not\subseteq X$ with an assignment to the variables in X', such that no clause is fully assigned with "false." Take $x \in X \setminus X'$. By maximality of X', there must exist clauses c_i and c_j, one of which is filled up with "false" literals once x is assigned with "true" and the other one, when x is given the value "false." Clearly c_i and c_j satisfy (i) and (i). ■

Note that if c_i and c_j satisfy (i) then $c_i \cup c_j \setminus \{x, \bar{x}\}$ is obtained from c_i and c_j by resolution. Theorem 5.1 can be restated as: *C is greedily satisfiable if and only if it is closed to resolution*. (A clause, obtained by resolution, which contains another is not considered a new one.) In fact the completeness of the resolution method [17] is proven by showing that greedy satisfiability fails only when a new clause is obtained by resolution.

Let C be a formula in *CNF* on a variable set X. A *valid partial assignment* of C is a set of pairs $\{(x_1, b_1), \dots, (x_k, b_k)\}$, where $x_i \in X, x_i \neq x_j$ for $i \neq j, b_i \in \{true, false\}, k < n$ and no clause of C has all its literals "false" when $x_i = b_i$ for every $i = 1, \dots, m - 1$. The valid partial assignments clearly form a hereditary system on the ground set $X \times \{true, false\}$. Accordingly, we call a formula C *greedily saturated* if all its maximal valid partial assignments are of the same cardinality. A satisfiable formula is greedily satisfiable if and only if it is greedily saturated. The recognition problem, however, is harder (provided $P \neq NP$) when applied to nonsatisfiable formulas.

THEOREM 5.2. *The recognition of greedily saturated CNF formulas is co-NPC. This remains true where every clause consists of either one or two literals.*

Proof. Let the graph $G = (V, E)$ be an input for *g-MS*. We construct a *CNF* formula $C(G)$ as follows: Let V be the variable set and for every $x \in V$ let the singleton $\{x\}$ be a clause of $C(G)$. Thus, a valid partial assignment of $C(G)$ is forced to contain only pairs of the form $(x, true)$. Also for every edge $(x, y) \in E$ let $\{\bar{x}, \bar{y}\}$ be a clause of $C(G)$. This allows only one endvertex of an edge to be represented in a valid partial assignment. It turns out that every valid partial assignment of $C(G)$ corresponds to a stable set of G. Hence *g-MS* (co-*NPC* by Theorem 2.1) is reducible to the problem at hand. ■

6. GRAPH COLORING

The simplest approach toward greedy coloring of a graph is taken by arbitrarily selecting (one at a time) a, not yet colored, vertex x and a color for x, provided that the selected color differs from those already given to neighbors of x. It can easily be observed that the above always produces a proper k-coloring of a graph G, if and only if the maximum degree of a vertex in G is at most $k - 1$. Such graphs are of course polynomially recognizable.

A more sophisticated greedy scheme is defined and studied by Gyarfas and Lehel in [10]. Focusing on hereditary systems, we find the following fitting better to the scope of this paper:

For a graph $G = (V, E)$, let $\mathscr{V}_k(G)$ be the family of subsets of V, which induce k-colorable subgraphs. Notice that $\mathscr{V}_1(G)$ is the family of stable sets of G and hence, by Theorem 2.1, the recognition of graphs G for which $\mathscr{V}_1(G)$ is greedy is *co-NPC*. We present here a similar result for any positive integer k. Telling that an induced k-colorable subgraph is maximal, might not be in *NP* for $k \geq 3$. Accordingly, *co-NPC* is replaced in that case by *co-NP-hard*. For the same reason there is not much of a greedy *algorithm* here, where checking the feasibility of a set is already *NPC*.

THEOREM 6.1. *For any positive integer k, telling for an input graph G whether $\mathscr{V}_k(G)$ is a greedy system is co-NP-hard (co-NPC for $k = 1$ and $k = 2$).*

Proof. Let $G = (V, E)$ be an input graph for *g-MS* (see Section 2.1). The graph $G \times K_k$ is obtained by taking a k-clique K_x for every vertex $x \in V$ and k^2 edges, connecting every vertex of K_x with every vertex of K_y, for every edge $(x, y) \in E$. Take now any collection of k (not necessarily distinct) stable sets of G. Its members can be viewed as disjoint stable sets of $G \times K_k$ (whose union is a set in $\mathscr{V}_k(G \times K_k)$), by taking distinct vertices of the clique K_x, one for each set of the collection which includes x. On the other hand, a set of $\mathscr{V}_k(G \times K_k)$ clearly corresponds to a collection of k (possibly intersecting) stable sets of G. It turns out that $\mathscr{V}_k(G \times K_k)$ is greedy if and only if G is a greedy instance of *g-MS*. The proof is completed by Theorem 2.1. ■

In his survey paper [16], Plummer discusses the graph classes W_n: A graph G belongs to W_n if, given any collection of n disjoint stable sets $\{I_1, \dots, I_n\}$ in G, there exist disjoint maximum stable sets $J_1, \dots, J_n$, such that $I_i \subseteq J_i$, for each $i = 1, \dots, n$. Obviously $\mathscr{V}_n(G)$ is greedy for $G \in W_n$. The converse is wrong already for $n = 2$ (check a simple path on three vertices). However, $G \times K_n \in W_n$ if and only if G is MS-greedy, and thus we obtain as a result of the previous proof:

THEOREM 6.2. *For any positive integer n, recognizing a member of W_n is co-NP-complete.*

Here the complement problem is NP, because $G \notin W_n$ can be verified by a maximal collection of n disjoint stable sets, at least one of which is not maximum.

7. OPEN PROBLEMS AND CONCLUDING REMARKS

We mention here some topics, not covered in previous sections, which might carry some general interest.

In this article, we focus on hereditary systems induced by problems where the goal is to find a maximum set of certain property. We showed no example where the underlying problem is of ‘‘minimum’’ type. Take, for example, the minimum dominating set problem (find a minimum set of vertices with at least one member adjacent to every vertex out of that set). A greedy instance of that problem is a graph where all minimal dominating sets are of the same size (that is, the complementary hereditary system is greedy in the usual sense). We mention this specific problem (min dominating set), mainly because we had given it some thought and yet have not determined the complexity status of the corresponding ‘‘recognizing greedy instances’’ problem.

Another greedy approach to minimum problems is greedily collecting elements until the required property is reached. This method significantly differs from our general scheme. Greedy instances seem to be more rare and hence, maybe, easier to characterize.

A phenomenon, about the width of its nature we wonder, is observed when comparing Theorem 2.2 to Theorem 2.3, Theorem 5.2 to Theorem 5.1, and Corollary 2.1 to Theorem 2.5. In all three cases we start with an *NPC* underlying problem which has a ‘‘perfect’’ version and a more generalized ‘‘max’’ version. The recognition of greedy ‘‘perfect’’ instances is polynomial, while the recognition of greedy instances of the general ‘‘max’’ problems is co-*NP*-complete. A different scenario is drawn by the results in Section 4: Again, we face an *NPC* problem, which has a ‘‘perfect’’ and a general ‘‘max’’ versions (Hamiltonian path versus maximum path), but here the recognition of greedy instances is polynomial for both versions. However, the family of simple paths (as subsets of the edge set) is not a hereditary system. The ‘‘right’’ system to look at for that matter is the family of linear forests: unions of vertex disjoint simple paths. Can a graph for which this system is greedy, be recognized in polynomial time?

Among open problems, let us mention again that of recognizing a greedily matchable hypergraph, where the edge size is not bounded. Proposition 2.1 and the structural characterization stated in Lemma 2.1 (both originally developed in [1]) might be a step forward, but so far we have not determined the complexity status of that problem.

By a theorem of Yannakakis [26], finding a maximum feasible set of a hereditary system on the vertex set of a graph, defined by a non-trivial property of the induced subgraph, is always *NP*-hard. What about the recognition of greedy systems of that type? A property is nontrivial if for every n there exists a graph on more than n vertices for which the property holds, as well as one for which it does not. Being k-colorable, in particular 1-colorable (that is, having a stable vertex set), are of course such properties and hence Theorems 2.1 and 6.1 are results of that type, we tend to believe that such recognition problems are always co-*NP*-hard. However, our attempt to adapt Yannakakis' scheme seems to be leading nowhere. Proving a general ''Yannakakis-type'' result might be a very hard task. Can it be done at least for some wide families of nontrivial properties?

A potentially interesting direction might be the recognition of ''*approximately greedy*'' structures—instances where the greedy algorithm guarantees to provide a good approximation of the optimal goal. ''Good'' can be defined to that matter in terms of bounded ratio or difference between the optimum and the reached outcome.

To conclude: Almost every combinatorial problem can be approached by some greedy scheme and hence it gives raise to at least one ''recognizing greedy instances'' problem. We hope the few problems treated within the scope of our article and those listed in this last section to be a ''representing sample'' which can shed some light on the area and encourage further research.

ACKNOWLEDGMENT

The authors thank Mike Plummer, who stimulated our research in this area by providing related material of his own.

REFERENCES

1. Y. Caro, J. Rojas, and S. Ruiz, A forbidden subgraph characterization and a polynomial algorithm for randomly decomposable graphs *Czech. Math. J.*, to appear.
2. G. Chartrand and H. W. Kronk, Randomly traceable graphs, *Siam J. Appl. Math.* **16** (1968), 696–700.

3. G. Chartrand, H. V. Kronk, and D. R. Lick, Randomly Hamiltonian digraphs, *Fund. Math.* **65** (1969), 223–226.
4. V. Chvátal and P. J. Slater, A note on well covered graphs, preprint, 1991.
5. G. Dirac and C. Thomassen, Graphs in which every finite path is contained in a circuit, *Math. Ann.* **203** (1973), 65–75.
6. D. Dor and M. Tarsi, Graph decomposition is NPC—A complete proof of Holyer's conjecture, *in* ''Proc. 24th Ann. ACM Symp. on Theory of Computing, Victoria, British Columbia, 1992''; *SIAM J. Comput.*, to appear.
7. S. Even, ''Graph Algorithms,'' Computer Science Press, Potomac, MD, 1979.
8. J. F. Fink, Randomly antitraceable digraphs, *J. Graph Theory* **6** (1982), 481–488.
9. M. R. Garey and D. S. Johnson, ''Computers and Intractability, A Guide to the Theory of NP-Completeness,'' Freeman, San Francisco, 1979.
10. A. Gyarfas and J. Lehel, On-line and first fit colorings of graphs, *J. Graph Theory* **12** (1988), 217–227.
11. P. Helman, B. M. E. Moret, and H. Shapiro, An Exact Characterization of greedy structures, preprint, 1992.
12. D. G. Kirkpatrick and P. Hell, On the completeness of a generalized matching problem, *in* ''Proc. 10th Annual ACM Symposium on Theory of Computing Machinery, New York, 1978.''
13. M. Lesk, M. D. Plummer, and W. R. Pulleyblank, Equimatchable graphs, *in* ''Graph Theory and Combinatorics'' (B. Bollobàs, Ed.), pp. 239–254, Academic Press, New York/London, 1984.
14. L. Lovàsz and M. D. Plummer, Matching theory, *Ann. Discrete Math.* **29** (1986).
15. G. J. Minty, On maximal independent sets of vertices in claw-free graphs, *J. Combin. Theory Ser. B* **28** (1980), 284–304.
16. M. D. Plummer, Well-covered graphs: A survey, preprint, 1992.
17. J. A. Robinson, A machine oriented logic based on the resolution principle, *J. Assoc. Comput. Mach.* **12** (1965), 23–41.
18. R. S. Sankaranarayana and L. K. Stewart, Complexity results for well covered graphs, Univ. of Alberta, Dept. of Comp. Science, preprint, 1991.
19. D. P. Sumner, Randomly matchable graphs, *J. Graph Theory* **3** (1979), 183–186.
20. D. Tankus, Greedily Hamiltonian graphs, Tel-Aviv University, unpublished, 1993.
21. D. Tankus and M. Tarsi, Well-covered claw-free graphs, *J. Combin. Theory Ser. B*, to appear.
22. M. Tarsi, Graphs where every maximal path is maximum, *J. Combin. Theory Ser. B*, to appear.
23. C. Thomassen, On randomly Hamiltonian digraphs, *Math. Ann.* **200** (1973), 195–208.
24. C. Thomassen, On Graphs in which every path is contained in a hamiltonian path, *J. Reine Angew. Math.* **268/269** (1974), 271–282.
25. B. Vizvari, A polynomial algorithm to prove or disprove the optimality of the greedy solution of a Knapsack problem, preprint, 1992.
26. N. Yannakakis, Note and edge-deletion NP-complete problems, *in* ''Proc. 10th Ann. ACM Symp. on Theory of Computing, New York, 1978.