INFORMATION AND COMPUTATION **119**, 231–239 (1995)

The Complexity of Selecting Maximal Solutions*

ZHI-ZHONG CHEN

Department of Information Engineering, Mie University, Tsu-shi, Mie 514, Japan
E-mail: chen@info.mie-u.ac.jp

AND

SEINOSUKE TODA

Department of Computer Science and Information Mathematics, University of Electro-Communications, Chofu-shi, Tokyo 182, Japan
E-mail: toda@cs.uec.ac.jp

Many important computational problems involve finding a maximal (with respect to set inclusion) solution in some combinatorial context. We study such maximality problems from the complexity point of view, and categorize their complexity precisely in terms of tight upper and lower bounds. Our results give characterizations of coNP, D^P, Π_2^P, $FP_{\|}^{NP}$, FNP//OptP[log n] and $FP_{\|}^{\Sigma_2^P}$ in terms of subclasses of maximality problems. An important consequence of our results is that finding an X-minimal satisfying truth assignment for a given CNF boolean formula is complete for FNP//OptP[log n], solving an open question by Papadimitriou [*Proceedings of the 32nd IEEE Symposium on the Foundations of Computer Science*, 1991, pp. 163–169].

1. INTRODUCTION

Combinatorial optimization problems have played a central role in the development of the theory of computational complexity, and have been the motivating influence for the introduction of many important complexity classes and concepts. Recently, a natural variant of combinatorial optimization problems, called *maximality problems* (MAXPs) in this paper, has attracted attention [And85, CM87, Che91, CT93, KW84, KUW88, Lub85, Pap91, PW85, PY82]. Intuitively speaking, a MAXP asks one to select a maximal solution, under set inclusion, for a given input. The maximal independent set problem (MIS) [KW84] and the minimal unsatisfiability problem (MinUnsat) [PW85] are two important examples of MAXPs. Most of the previous work has involved studying the complexity of *specific* MAXPs and either designing an efficient algorithm (e.g., [KW84]) or proving the computational hardness of the task (e.g., [PW85]). An attractive alternative approach is to study MAXPs in a general framework, to identify the essential factors which influence the complexity of solving MAXPs, and to characterize their complexity precisely in terms of complexity classes.

In this paper, we formalize a MAXP Q as a pair (D, R), where D is the set of *instances* and R: $D \times \{0, 1\}^* \to \{$*true*, *false*$\}$ is the *instance–solution relation*. The objective in solving Q is to select, given an instance $x \in D$, a *maximal solution*, i.e., a binary string w such that $R(x, w)$ is true but changing one or more arbitrary 0-bits of w to 1-bits will change the value of $R(x, w)$ to false. As an example, consider the MIS problem in our framework. In this case, D is the set of all undirected graphs, and $R(G, b_1 b_2 \cdots b_n)$ is true if and only if G has n vertices (say, 1, 2, ..., n) and $\{i: b_i = 1\}$ is an independent set in G. Our goal is to demonstrate what factors make Q computationally easy or hard to solve. We are able to find two such factors. One obvious factor is the complexity of R. This can be seen by comparing MIS with MinUnsat. The instance–solution relation of MIS is decidable in NC while that of MinUnsat is coNP-complete. Because of this gap, solving MinUnsat is much harder than solving MIS. In fact, MIS is solvable in NC [KW84, Lub85] while solving MinUnsat is D^P-hard [PW85]. The other factor is whether R is *hereditary* or not; R is said to be hereditary if and only if for every x and w, whenever $R(x, w)$ is true, $R(x, w)$ remains true even if one or more arbitrary 1-bits of w are changed to 0-bits. The instance–solution relation of MIS (also of MinUnsat) is hereditary. In [Pap91], Papadimitriou considered the following problem (MinModel): Given a CNF boolean formula φ, find a satisfying truth assignment $\vec{a}$ of φ such that changing one or more arbitrary 1-bits of $\vec{a}$ to 0-bits will make $\vec{a}$ no longer satisfy φ. The instance-solution relation of MinModel is not hereditary but is decidable in NC. Unlike MIS, solving MinModel is obviously NP-hard.

In this paper, we consider only those MAXPs whose instance–solution relation is decidable in NP or coNP.

* A preliminary version of this paper appeared in the "Proceedings of the 8th Annual IEEE Conference on Structure in Complexity Theory, San Diego, May 1993," and was supported in part by International Information Science Foundation Grant 93-2-2-468.

0890-5401/95 $6.00

643/119/2-7

We first consider upper bounds on the complexity of solving such MAXPs. Let $Q = (D, R)$ be a MAXP. The following give trivial upper bounds: (i) Q is solvable in FP if R is decidable in P and hereditary; (ii) Q is solvable in FNP//OptP[log n] if R is decidable in NP; (iii) Q is solvable in $\mathrm{FP}^{\mathrm{NP}}$ if R is decidable in coNP and hereditary; (iv) Q is solvable in $\mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$ if R is decidable in coNP. Our main results concerning upper bounds are the following:

(v) Suppose Q is a MAXP whose instance-solution relation is NP decidable. Let e be an arbitrary polynomial. Then, there exist a function $F \in \mathrm{FP}_{\|}^{\mathrm{NP}}$ and a polynomial p such that for every x, Prob[$F(x, w)$ is a maximal solution of x in Q] $\geqslant 1 - 2^{-e(|x|)}$, where $w \in \{0, 1\}^{p(|x|)}$ is randomly chosen under uniform distribution.

(vi) Suppose Q is a MAXP whose instance–solution relation is coNP decidable. Let e be an arbitrary polynomial. Then there exist a function $F \in \mathrm{FP}_{\|}^{\Sigma_2^{\mathrm{P}}}$ and a polynomial p such that for every x, Prob[$F(x, w)$ is a maximal solution of x in Q] $\geqslant 1 - 2^{-e(|x|)}$, where $w \in \{0, 1\}^{p(|x|)}$ is randomly chosen under uniform distribution.

We then show that FNP//OptP[log n] is also a lower bound for solving those MAXPs whose instance–solution relation is decidable in NP or is decidable in P but not hereditary, and that $\mathrm{FP}_{\|}^{\Sigma_2^{\mathrm{P}}}$ is also a lower bound for solving those MAXPs whose instance–solution relations are decidable in coNP but not hereditary. Combining the upper and lower bounds, we obtain characterizations of FNP//OptP[log n] and $\mathrm{FP}_{\|}^{\Sigma_2^{\mathrm{P}}}$ via MAXPs. As an important consequence of the characterization of FNP//OptP[log n], we obtain the first natural maximality problem complete for FNP//OptP[log n]. The problem (called X-MinModel) is defined as follows: Given a CNF boolean formula $\varphi(x_1, ..., x_n)$ and a subset X of $\{x_1, ..., x_n\}$, find a satisfying truth assignment $a_1 a_2 \cdots a_n$ of φ such that no other satisfying truth assignment $b_1 b_2 \cdots b_n$ of φ can meet the condition that $\{x_i: b_i = 1\} \cap X$ is properly contained in $\{x_i: a_i = 1\} \cap X$. X-MinModel was first considered by Papadimitriou in [Pap91], and was claimed without a precise proof to be Δ_2^{P}-complete there. However, Papadimitriou later withdrew his claim and thus left the complexity of X-MinModel open [Pap92]. In [CT93], we proved that the complexity of X-MinModel is roughly captured by $\mathrm{FP}_{\|}^{\mathrm{NP}}$. Our results in this paper succeed in finally characterizing exactly the complexity of X-MinModel.

We also characterize several complexity classes in terms of MAXPs. The following are shown:

(a) coNP is the class of all sets L that can be expressed as $L = \{x\colon f(x)$ is a maximal solution of x in $Q\}$ for some $f \in$ FP and some MAXP Q whose instance–solution relation is P-decidable.

(b) $\mathrm{D^P}$ is the class of all sets L that can be expressed as $L = \{x\colon f(x)$ is a maximal solution of x in $Q\}$ for some $f \in$ FP and some MAXP Q whose instance–solution relation is NP-decidable.

(c) $\mathrm{D^P}$ is the class of all sets L that can be expressed as $L = \{x\colon f(x)$ is a maximal solution of x in $Q\}$ for some $f \in$ FP and some MAXP Q whose instance–solution relation is coNP-decidable and hereditary.

(d) Π_2^{P} is the class of all sets L that can be expressed as $L = \{x\colon f(x)$ is a maximal solution of x in $Q\}$ for some $f \in$ FP and some MAXP Q whose instance–solution relation is coNP-decidable.

As consequences, we obtain several new natural problems that are $\leqslant_m^{\mathrm{P}}$-complete for coNP or $\mathrm{D^P}$.

In the final section, the results obtained in this paper are summarized in Tables 1 and 2. We will also exhibit several open questions related to this work.

2. PRELIMINARIES

2.1. *Basic Definitions and Notations*

We use $\Sigma = \{0, 1\}$ as our alphabet. By a *set*, we mean a subset of Σ^*. Similarly, by a *string*, we mean a string in Σ^*. We denote by $|x|$ the length of a finite string x. The bits of a finite string with length n are indexed from left to right as the 1st, 2nd, ..., nth bits, respectively. For a finite string x, we usually identify x with the set of all indices i such that the ith bit of x is 1. Thus we will often use some set-theoretical notations for finite strings. A finite string x is *smaller* than another finite string y if either $|x| < |y|$ or $|x| = |y|$ and $x \subset y$. A *maximal string* in a set S of finite strings is a string in S that is not smaller than any other string in S.

Σ^n denotes the set of all strings with length n. For a nonnegative integer n, *bin*(n) denotes the standard binary representation of n, and 0^n (resp., 1^n) denotes the string in Σ^n whose bits are all 0-bits (resp., 1-bits). Throughout this paper, all logarithms are to base 2 and $n \cdot m$ denotes the product of integer n by integer m.

We assume a standard one-to-one pairing function from $\Sigma^* \times \Sigma^*$ to Σ^* that is polynomial-time computable and polynomial-time invertible. For strings x and y, we denote the output of the pairing function by $\langle x, y \rangle$; this notation is extended to any k-tuples for $k > 2$ in a usual manner. W.l.o.g., we assume that $|\langle x, y \rangle|$ depends only on $|x|$ and $|y|$.

For any finite set A, $\|A\|$ denotes the number of strings in A. For a set L, $\bar{L}$ denotes its complement (i.e., $\Sigma^* - L$), and χ_L denotes the characteristic function of L. For a class **C** of sets, co**C** denotes the class of all sets whose complements are in **C**.

All functions considered here are ones from Σ^* to $\Sigma^* \cup \{\#\}$. The symbol # is assumed to be not in Σ^*. We consider both single-valued functions and multi-valued functions, but by a *function* we mean a (partial) single-valued function. For a multi-valued function G, $G(x)$ denotes the set of all possible values of G at x. Thus, when $G(x)=\emptyset$, the multi-valued function G is *undefined* at the argument x.

We assume that the reader is familiar with the basic concepts from the theory of computational complexity. Our computational models are variations of standard Turing machines. A machine is either an acceptor or a transducer, and may be deterministic or nondeterministic. An acceptor is denoted by M or M_i, while a transducer is denoted by T or T_i. A deterministic (resp., nondeterministic) Turing machine is abbreviated as DTM (resp., NTM). On a given input, a branch of a (nondeterministic) machine may halt by entering either a rejecting state or an accepting state. For simplicity, we say that a branch of a machine *halts* if the branch halts by entering an accepting state. Let $L(M)$ denote the set of all strings accepted by M. A transducer T *computes a string* y *on input* x if some branch of T on input x halts with y on the output tape. $T(x)$ denotes the set of all strings computed by T on input x. A DTM T *computes a function* f if for all $x \in \Sigma^*$, $T(x)=\emptyset$ if $f(x)$ is undefined, and the unique element of $T(x)$ is $f(x)$ otherwise.

Classes in the first three levels of the polynomial-time hierarchy are denoted in the usual way: P, NP, coNP, Σ_2^{P}, $\Pi_2^{\mathrm{P}} = \mathrm{co}\Sigma_2^{\mathrm{P}}$. Let $\mathrm{D}^{\mathrm{P}} = \{L_1 \cap L_2 : L_1 \in \mathrm{NP} \text{ and } L_2 \in \mathrm{coNP}\}$.

FP denotes the class of all functions computed by polynomial-time bounded DTMs. Let A be a set. FP^A denotes the class of all functions computed by polynomial-time bounded deterministic oracle Turing machines (DOTM) with oracle A. $\mathrm{FP}_{\parallel}^A$ denotes the class of all functions F for which there exists a polynomial-time bounded DOTM T such that T, while computing $F(x)$ for a given x, prepares all its query strings before asking them of the oracle A. More precisely, a function F is in $\mathrm{FP}_{\parallel}^A$ if there exist two functions f and g in FP such that for all strings x, $F(x) = g(x, \langle \chi_A(y_1), ..., \chi_A(y_m)\rangle)$, where $f(x) = \langle y_1, ..., y_m\rangle$. For a class $\mathbf{C}$ of sets, $\mathrm{FP}^{\mathbf{C}} = \bigcup_{A \in \mathbf{C}} \mathrm{FP}^A$ and $\mathrm{FP}_{\parallel}^{\mathbf{C}} = \bigcup_{A \in \mathbf{C}} \mathrm{FP}_{\parallel}^A$.

A *metric Turing machine* is an NTM T such that on every input, every branch of T outputs a binary number and halts [Kre88]. $\mathrm{OptP}[\log n]$ denotes the class of all (total) integer-valued functions H for which there exist a polynomial p and a polynomial-time bounded metric Turing machine T such that for every x, $H(x) \leq p(|x|)$ and $H(x)$ equals to the maximum number in $T(x)$. $\mathrm{FNP//OptP}[\log n]$ denotes the class of all (partial) multi-valued functions G for which there exist a polynomial-time bounded NTM T and a function $H \in \mathrm{OptP}[\log n]$ such that for every x, $G(x) = T(\langle x, bin(H(x))\rangle)$. $\mathrm{FNP//OptP}[\log n]$ is the function analogue of $\mathrm{NP//OptP}[\log n]$ defined in [KT90].

2.2. *Maximality Problems*

A *maximality problem* (MAXP) Q is a pair (D, R), where

(i) D is the set of *instances* and

(ii) R: $D \times \Sigma^* \to \{true, false\}$ is the *instance–solution relation*.

R is said to be *hereditary* if for every $x \in D$ and every $w \in \Sigma^*$, whenever $R(x, w)$ is true, $R(x, w')$ is also true for every w' with $|w'| = |w|$ and $w' \subset w$. Let $x \in D$. A string w is called a *solution* of x if $R(x, w)$ is true. A *maximal solution* of x is a maximal string in the set of all solutions of x. The objective in solving Q is to either determine that x has no solution in Q or find a maximal solution of x in Q for any given $x \in D$.

In this paper, we consider only those MAXPs $Q=(D, R)$ that satisfy the following:

(1) D is P-decidable (i.e., decidable in polynomial time),

(2) there is a polynomial p such that for every $x \in D$ and every string w, whenever $R(x, w)$ is true, $|w| \leq p(|x|)$, and

(3) R is NP-decidable or coNP-decidable.

Our goal is to investigate the complexity of solving MAXPs. Let $Q=(D, R)$ be a MAXP. To discuss the computational easiness of solving Q, we need some definitions.

Definition 2.1. A function F *solves* Q if for every $x \in D$, (a) $F(x)$ is undefined if x has no solution in Q, and (b) $F(x)$ is a maximal solution of x in Q otherwise. A multi-valued function G *solves* Q if for every $x \in D$, (a) $G(x)=\emptyset$ if x has no solution in Q, and (b) $G(x)$ is nonempty and each element of $G(x)$ is a maximal solution of x in Q otherwise. Q is *solvable* in a class $\mathbf{H}$ of (single-valued or multi-valued) functions if some $H \in \mathbf{H}$ solves Q.

Note that, in the above definition, we do not require that $G(x)$ contain all maximal solutions of x in Q.

To discuss the computational hardness of solving Q, we need to define a kind of reducibility. The following definition is analogous to that of Krentel's metric reducibility [Kre88].

Definition 2.2. A function F is *reducible* to Q if there exist two functions f, g in FP such that for every x, (a) $f(x)$ is in D and has solutions in Q, and (b) $F(x) = g(x, w)$ for every maximal solution w of $f(x)$ in Q. A multi-valued function G is *reducible* to Q if there exist two functions f, g in FP such that for every x, (a) $f(x)$ is in D and has solutions in Q, (b) $g(x, w)$ is undefined for every maximal solution w of $f(x)$ in Q if $G(x)=\emptyset$, and (c) $g(x, w) \in G(x)$ for every maximal solution w of $f(x)$ in Q if $G(x) \neq \emptyset$.

Definition 2.3. Q is *hard* for a class $\mathbf{H}$ of (single-valued or multi-valued) functions if every $H \in \mathbf{H}$ is reducible to Q. Q is *complete* for a class $\mathbf{H}$ of (single-valued or multi-valued) functions if Q is solvable in and hard for $\mathbf{H}$. Q is *hard* for a class $\mathbf{C}$ of sets if Q is hard for the class $\{\chi_L : L \in \mathbf{C}\}$.

To translate Q into a decision problem, we do the following definition.

DEFINITION 2.4. The set $L_Q = \{\langle x, w\rangle : w$ is a maximal solution of x in $Q\}$ is called the *decision problem associated with Q*.

3. UPPER BOUNDS

In this section, we show upper bounds on the complexity of solving MAXPs. The following simple proposition shows trivial upper bounds.

PROPOSITION 3.1. *Let $Q = (D, R)$ be a MAXP.*

(1) *If R is hereditary and* P-*decidable, then Q is solvable in* FP.

(2) *If R is* NP-*decidable, then Q is solvable in* FNP// OptP[log n].

(3) *If R is hereditary and* coNP-*decidable, then Q is solvable in* $\mathrm{FP}^{\mathrm{NP}}$.

(4) *If R is* coNP-*decidable, then Q is solvable in* $\mathrm{FP}^{\Sigma_2^P}$.

Proof. We only give a proof for (3). Let p be a polynomial such that for all $x \in D$, the length of each solution of x in Q is no more than $p(|x|)$. Consider a DTM T which, given $x \in D$, operates as follows:

Step 1. T checks whether there is some l such that $0 \leqslant l \leqslant p(|x|)$ and 0^l is a solution of x in Q.

Step 2. If there is no such l, then T halts by entering a rejecting state.

Step 3. Let m be the maximum such l. T performs the following step for $i = 1, 2, ..., m$ in turn:

If $b_1 b_2 \cdots b_{i-1} 10^{m-i}$ is a solution of x in Q,

then T sets $b_i = 1$; otherwise, T sets $b_i = 0$.

Step 4. T outputs $b_1 b_2 \cdots b_m$ and halts.

Since R is hereditary, T correctly finds a maximal solution of x in Q. Moreover, it is clear that T runs in polynomial time by asking queries to some NP oracle. ▮

Before proceeding to two other non-trivial upper bounds, we need to define a new class of multi-valued functions from a given class **F** of functions. The new class is denoted by RP · **F**. Intuitively speaking, RP · **F** consists of all multi-valued functions G for which there is a randomized polynomial-time algorithm that uses some function in **F** as a subroutine and outputs some element of $G(x)$ for any given x with high probability.

DEFINITION 3.2. Let **F** be a class of functions. A multi-valued function G is in RP · **F** if for every polynomial e, there exist a function $F \in$ **F** and a polynomial p such that for every string x, (a) $F(x, w)$ is undefined for all $w \in \{0, 1\}^{p(|x|)}$ if $G(x) = \varnothing$, and (b) $\mathrm{Prob}[F(x, w) \in G(x) \cup \{\#\}] = 1$ and $\mathrm{Prob}[F(x, w) \in G(x)] \geqslant 1 - 2^{-e(|x|)}$ if $G(x) \neq \varnothing$, where w is a random string chosen from $\{0, 1\}^{p(|x|)}$.

DEFINITION 3.3. Let S be a finite set and let k be a positive integer. A *weight function* over S is a function from the elements of S to positive integers. A *k-weight function* over S is a weight function f over S such that for each $s \in S$, $1 \leqslant f(s) \leqslant k$. A *random k-weight function* over S is a k-weight function f over S such that for each $s \in S$, $f(s)$ is chosen uniformly and independently from $\{1, 2, ..., k\}$. The *weight* of a subset S' of S under a weight function f is the sum of all $f(s)$ with $s \in S'$. Note that for every k-weight function over S, the weight of each subset of S under f is no more than $k \cdot \|S\|$ and that the empty set is the unique subset of S with weight 0.

LEMMA 3.4 [MVV87]. *Let* **S** *be a nonempty family of subsets of a finite set S. Then, for any random k-weight function f over S with $k \geqslant 2 \cdot \|S\|$,* Prob[*There is a unique maximum weight set in* **S** *under f*] $\geqslant \frac{1}{2}$.

Now we are ready to show the two non-trivial upper bounds.

THEOREM 3.5. *Let $Q = (D, R)$ be a MAXP.*

(1) *If R is* NP-*decidable, then Q is solvable in* $\mathrm{RP} \cdot \mathrm{FP}_{\parallel}^{\mathrm{NP}}$.

(2) *If R is* coNP-*decidable, then Q is solvable in* $\mathrm{RP} \cdot \mathrm{FP}_{\parallel}^{\Sigma_2^P}$.

Proof. We only give a proof for (1). The second part of the theorem can be shown in a similar manner.

We first explain the idea behind the proof. Let x be an instance of Q, and let n_1 be the maximum length of solutions of x in Q. Then, we consider **S**, the family of all solutions of x with length n_1. To find a maximal solution for x, we first get a random $2 \cdot n_1$-weight function f over $S = \{1, 2, ..., n_1\}$. (Recall that we identify a solution $s_1 s_2 \cdots s_{n_1}$ with the set $\{i : s_i = 1\}$. So **S** can be thought of as a family of subsets of S.) Then, by Lemma 3.4, with probability at least $\frac{1}{2}$, there is a unique solution in **S** of maximum weight. Being of maximum weight, this solution must be maximal. To find this solution, it suffices to ask only one round of parallel queries to an NP oracle. In order to get the high probability of success, we may perform several copies of this computation in parallel.

We now make the above idea precise. Let q be a polynomial that bounds the lengths of solutions of instances of Q from above. We define a polynomial p from q as follows: $p(i) = q^2(i) + q(i)$. Consider an algorithm which, given an input $\langle x, w\rangle$ with $x \in D$ and $w \in \{0, 1\}^{p(|x|)}$, performs the following steps:

Step 1. Check whether x has a solution or not. If not, then halt by entering a rejecting state.

Step 2. Find n_1, the length of the longest solutions of x.

Step 3. Let $n_2 = 2^{1+\lceil \log n_1 \rceil}$. Construct, from w, an n_2-weight function f over the set $\{1, 2, ..., n_1\}$ as follows: Let $w_1 w_2 \cdots w_{n_1}$ be a prefix of w such that $|w_1| = \cdots = |w_{n_1}| = \log n_2$. (The remaining part of w is ignored.) Then, for each $1 \leqslant i \leqslant n_1$, set $f(i) = d_i + 1$, where $bin(d_i) = w_i$.

Step 4. Check whether x has a *unique* solution of length n_1 which has maximum weight under f. If not, output the special symbol # and halt.

Step 5. For $1 \leqslant i \leqslant n_1$, compute b_i, the ith bit of the unique solution; output $b_1 b_2 \cdots b_{n_1}$ and halt.

Step 3 can be trivially done in polynomial time. Other steps can be done in polynomial time by asking queries to some NP oracle. Since n_1 and the maximum weight of solutions of x under f are polynomial in $|x|$, the queries can be prepared independently of each other. Thus, we can implement the algorithm as a DOTM T which runs in polynomial time by asking one round of parallel queries to an NP oracle. Moreover, if w is randomly chosen from $\{0, 1\}^{p(|x|)}$, then by Lemma 3.4, the probability that $T(\langle x, w \rangle)$ is a maximal solution of x is at least $\frac{1}{2}$.

Let e be an arbitrary polynomial. Consider a DOTM T' which, on input $\langle x, w_1 \cdots w_{e(|x|)} \rangle$ with $x \in D$ and each $w_i \in \{0, 1\}^{p(|x|)}$, simulates in parallel $e(|x|)$ copies of T on inputs $\langle x, w_1 \rangle, ..., \langle x, w_{e(|x|)} \rangle$, respectively. If for some i, $T(\langle x, w_i \rangle)$ is defined and is not #, then T' finds, say, the smallest such i and halts with output $T(\langle x, w_i \rangle)$. Otherwise, $T'(\langle x, w_1 \cdots w_{e(|x|)} \rangle)$ is $T(\langle x, w_1 \rangle)$. Let F be the function computed by T'. Obviously, $F \in \mathrm{FP}_{\parallel}^{\mathrm{NP}}$. Moreover, if $w_1 \cdots w_{e(|x|)}$ is randomly chosen from $\{0, 1\}^{e(|x|) \cdot p(|x|)}$, then the probability that $F(\langle x, w_1 \cdots w_{e(|x|)} \rangle)$ is a maximal solution of x is at least $1 - 2^{-e(|x|)}$. We define a multi-valued function G as follows: $G(x) = \{F(x, w) \colon w \in \{0, 1\}^{p(|x|)}$ and $F(x, w)$ is defined$\} - \{\#\}$. Then from the above discussions, one can easily see that G solves Q and is in $\mathrm{RP} \cdot \mathrm{FP}_{\parallel}^{\mathrm{NP}}$. Hence, the theorem holds.

4. HARDNESS OF SOLVING MAXPs

The following proposition shows that FP is a tight lower bound on the complexity of solving MAXPs whose instance-solution relation is P-decidable and hereditary.

PROPOSITION 4.1. *There is a MAXP $Q = (D, R)$ such that R is* P*-decidable and hereditary and Q is complete for* FP.

Proof. Let D be the set of all triples $\langle T, 1^t, x \rangle$ such that T is a DTM, $t > 0$, and $x \in \Sigma^*$. Define a relation R as follows: for all $\langle T, 1^t, x \rangle \in D$ and all $w \in \Sigma^*$, $R(\langle T, 1^t, x \rangle, w)$ is true if and only if $w = 0$ or w is smaller than or equal to $1w'$, where w' is the string output by T on input x in t steps. Obviously, R is P-decidable and hereditary. By Proposition 3.1(1), (D, R) is solvable in FP. We next show the hardness. Let f be a function in FP. Then, there is a polynomial-time bounded DTM T computing f. Let p be a polynomial that bounds the running time of T from above and let x be a string in Σ^* for which $f(x)$ is sought. By the definition of D, $\langle T, 1^{p(|x|)}, x \rangle \in D$. Fix an arbitrary maximal solution w of $\langle T, 1^{p(|x|)}, x \rangle$ in (D, R) to consider. By the definition of R, $w = 0$ or $w = 1w'$ for some $w' \in \Sigma^*$. If $w = 0$, we know that $f(x)$ is undefined; otherwise, we know that $f(x) = w'$. Thus, (D, R) is hard for FP. ∎

The following theorem shows that FNP//OptP[log n] is a tight lower bound on the complexity of solving MAXPs whose instance–solution relation is P-decidable but not hereditary.

THEOREM 4.2. *There is a MAXP $Q = (D, R)$ such that R is* P*-decidable and Q is complete for* FNP//OptP[log n].

Proof. We define D to be the set of all six-tuples $\langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle$, where $x \in \Sigma^*$, T_1 is a metric Turing machine, $t_1 > 0$, $d \geqslant 0$, T_2 is an NTM, and $t_2 > 0$. We define R as follows: for $\langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle \in D$ and $w \in \Sigma^*$, $R(\langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle, w)$ is true if and only if $w = 0^{(d+1) \cdot (t_1 + t_2 + 1)}$ or $w = 1^{k \cdot (t_1 + t_2 + 1)} 1 u 0^{(d-k) \cdot (t_1 + t_2 + 1)}$ such that $k \leqslant d$, $|u| = t_1 + t_2$, and u specifies both a t_1-step accepting computation path of T_1 on input x with output k and a t_2-step accepting computation path of T_2 on input $\langle x, bin(k) \rangle$. It is easy to see that R is P-decidable but not hereditary. By Proposition 3.1(2), (D, R) is solvable in FNP//OptP[log n].

Let $H \in$ FNP//OptP[log n] witnessed by a polynomial-time bounded metric Turing machine T_1, a polynomial q, and a polynomial-time bounded NTM T_2. Let p_1 be a polynomial that bounds the running time of T_1 from above, and let p_2 be a polynomial such that on any input $\langle x, bin(k) \rangle$ with $0 \leqslant k \leqslant q(|x|)$, T_2 runs in at most $p_2(|\langle x, bin(q(|x|)) \rangle|)$ steps. Fix an arbitrary $x \in \Sigma^*$. We set $y = \langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle$, where $t_1 = p_1(|x|)$, $d = q(|x|)$, and $t_2 = p_2(|\langle x, bin(q(|x|)) \rangle|)$. By the definition of R, $H(x) = \varnothing$ if and only if $0^{(d+1) \cdot (t_1 + t_2 + 1)}$ is the unique (maximal) solution of y in Q. Moreover, if $w = 1^{k \cdot (t_1 + t_2 + 1)} 1 u 0^{(d-k) \cdot (t_1 + t_2 + 1)}$ is a maximal solution of y in Q, then we can use u to compute an element of $H(x)$ in polynomial time. Thus, (D, R) is hard for FNP//OptP[log n]. ∎

As a by-product of the proofs of Theorem 3.5 and Theorem 4.2, we obtain the following corollary.

COROLLARY 4.3. FNP//OptP[log n] $\subseteq \mathrm{RP} \cdot \mathrm{FP}_{\parallel}^{\mathrm{NP}}$.

The problem defined in the following corollary is essentially the problem defined by Papadimitriou in Section 3 of

[Pap91]. The following corollary gives, for the first time, the exact complexity of this problem.

COROLLARY 4.4. *The following problem (called X-MaxModel hereafter) is complete for* FNP//OptP[log n]:

Instance: A CNF boolean formula φ and a subset X of the set of variables in φ.

Output: A truth assignment $\vec{a}$ to the variables in X such that $\vec{a}$ can be extended to a satisfying truth assignment of φ but no $\vec{b}$ with $\vec{a} \subset \vec{b}$ and $|\vec{a}| = |\vec{b}|$ can be extended to a satisfying truth assignment of φ.

Proof. X-MaxModel is obviously solvable in FNP//OptP[log n]. We next show its hardness. Let $Q = (D, R)$ be the MAXP constructed in the proof of Theorem 4.2. Recall that there is a polynomial p such that for every $x \in D$, the length of each solution of x in Q is equal to $p(|x|)$. Consider an NTM T which on input $x \in D$, guesses a string w of length $p(|x|)$, checks whether w is a solution of x in Q, and halts with output w if so. Obviously, for every $x \in D$, $T(x)$ is exactly the set of all solutions of x in Q. Let q be a polynomial that bounds the running time of T from above. Then by Cook's theorem, for every $x \in D$, we can construct from $\langle T, 1^{q(|x|)}, x \rangle$ in polynomial time a CNF boolean formula $\varphi(y_1, ..., y_m, z_1, ..., z_k)$ such that $y_1, y_2, ..., y_m$ encode a valid computation of some branch of T on input x; and $z_1 z_2 \cdots z_k$ is the output on this branch [Kre88]. Let $X = \{z_1, z_2, ..., z_k\}$. Clearly, the solutions of $\langle \varphi, X \rangle$ in X-MaxModel are exactly the elements of $T(x)$ and thus are exactly the solutions of x in Q. Therefore, the hardness of X-MaxModel follows from the hardness of Q. ∎

We next modify the proof of Theorem 4.2 to show that FNP//OptP[log n] is a tight lower bound on the complexity of solving MAXPs whose instance-solution relation is NP-decidable and hereditary.

THEOREM 4.5. *There is a MAXP $Q = (D, R)$ such that R is* NP*-decidable and hereditary and Q is complete for* FNP//OptP[log n].

Proof. The proof is a slight modification of the one for Theorem 4.2. D is the same as defined in the proof of Theorem 4.2. Let R_{old} be the relation defined in the proof of Theorem 4.2. Then, we define R as follows: for $\langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle \in D$ and $w \in \Sigma^*$, $R(\langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle, w)$ is true if and only if either $R_{\text{old}}(\langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle, w)$ is true or there is a u such that $w \subset u$, $|w| = |u|$, and $R_{\text{old}}(\langle x, T_1, 1^{t_1}, 1^d, T_2, 1^{t_2} \rangle, u)$ is true.

R is obviously NP-decidable and hereditary. Moreover, one can easily verify that (D, R) is hard for FNP//OptP[log n]. ∎

The instance-solution relation of X-MaxModel is NP-decidable but not hereditary. A natural question arises: Is there a natural MAXP Q such that Q is complete for FNP//OptP[log n] and the instance-solution relation of Q is either NP-decidable and hereditary or P-decidable (but not hereditary)? Unfortunately, we have been unable to settle this question. However, we below show that the question will have a positive answer if the complexity class concerned is replaced by $\text{FP}_{\parallel}^{\text{NP}}$. Before proceeding, we define a kind of property of MAXPs, called *paddability*, by borrowing some ideas from [CT91]. Intuitively speaking, we say that a MAXP Q is paddable if we can efficiently pad a list $\langle x_1, ..., x_m \rangle$ of instances of Q into a single instance x of Q and we can efficiently compute a maximal solution for each x_i from any given maximal solution of x.

DEFINITION 4.6. A MAXP $Q = (D, R)$ is *paddable* if there are two total functions h_1 and h_2 in FP such that for every list $I = \langle x_1, ..., x_m \rangle$ of instances of Q, (a) $h_1(I)$ is in D and has solutions in Q, and (b) for every maximal solution w of $h_1(I)$, $h_2(I, w) = \langle w_1, ..., w_m \rangle$, where for $1 \leqslant i \leqslant m$, $w_i = \#$ if x_i has no solution and w_i is a maximal solution of x_i in Q otherwise.

LEMMA 4.7. *If a MAXP $Q = (D, R)$ is paddable and hard for* NP, *then it is hard for* $\text{FP}_{\parallel}^{\text{NP}}$.

Proof. Let the paddability of Q be witnessed by two total functions h_1 and h_2 in FP. Let $F \in \text{FP}_{\parallel}^{\text{NP}}$. Then, there are two functions f_1, f_2 in FP and a set $L \in$ NP such that for every $x \in \Sigma^*$, $F(x) = f_2(x, \langle \chi_L(y_1), ..., \chi_L(y_m) \rangle)$, where $f_1(x) = \langle y_1, ..., y_m \rangle$. Since L is in NP and Q is hard for NP, there are two functions g_1, g_2 in FP such that for every $y \in \Sigma^*$, (a) $g_1(y)$ is in D and has solutions in Q, and (b) $y \in L$ if and only if $g_2(y, w) = 1$ for every maximal solution w of $g_1(y)$ in Q. Consider a DTM T which on input $x \in \Sigma^*$, operates as follows: T first computes $f_1(x)$ to get a list $\langle y_1, ..., y_m \rangle$ of strings, next computes $I = \langle g_1(y_1), ..., g_1(y_m) \rangle$, and finally computes $h_1(I)$. Clearly, T runs in polynomial time. Let f_T be the function computed by T.

Fix an arbitrary $x \in \Sigma^*$ and let w be an arbitrary maximal solution of $f_T(x)$ in Q. We next show that $F(x)$ can be computed in polynomial time from $\langle x, w \rangle$ and thus that Q is hard for $\text{FP}_{\parallel}^{\text{NP}}$. To compute $F(x)$ from $\langle x, w \rangle$, we first compute $h_2(I, w) = \langle w_1, ..., w_m \rangle$, where I is computed from x as stated in the definition of T. Next, for each y_i in $f_1(x) = \langle y_1, ..., y_m \rangle$, we compute $g_2(y_i, w_i)$ which is equal to $\chi_L(y_i)$. Finally, we compute $f_2(x, \langle \chi_L(y_1), ..., \chi_L(y_m) \rangle)$ which is equal to $F(x)$. Obviously, these computations can be done in polynomial time. ∎

THEOREM 4.8. *The following MAXPs are hard for* $\text{FP}_{\parallel}^{\text{NP}}$.

(1) *MAXIMAL MODEL (MaxModel)*

Instance: A CNF boolean formula φ.

Output: A maximal satisfying truth assignment *of φ, i.e., a satisfying truth assignment $\vec{a}$ of φ such that there is no other satisfying truth assignment $\vec{b}$ of φ with $\vec{a} \subset \vec{b}$.*

(2) *MAXIMAL CUBIC SUBGRAPH (MaxCubSubgraph)*

Instance: An undirected graph G.

Output: A maximal subset F of E(G) such that every vertex in the graph (V(G), F) has either degree 3 or degree 0. (Note: V(G) and E(G) denote hereafter the sets of vertices and edges of G, respectively.)

(3) *MAXIMAL SATISFIABILITY (MaxSat)*

Instance: A CNF boolean formula $\varphi = \{C_1, C_2, ..., C_m\}$.

Output: A maximal subset φ' *of* φ *that is satisfiable.*

(4) *MAXIMAL k-COLORABILITY* $(k \geqslant 3)$ *(Max-k-Colorability)*

Instance: An undirected graph G.

Output: A maximal subset of V(G) whose induced subgraph is k-colorable.

(5) *MAXIMAL HAMILTONIAN SUBGRAPH (MaxHamSubgraph)*

Instance: A pair $\langle G, w\rangle$ *of a connected undirected graph and a vertex in G.*

Output: A maximal subset U of V(G) such that $w \in U$ *and the subgraph induced by U has a Hamiltonian circuit.*

Proof. From several known results in the theory of NP-completeness, all MAXPs above can be easily shown to be hard for NP. It is also easy to verify the paddability of each MAXP in the theorem. Hence, by Lemma 4.7, each MAXP in the theorem is hard for $\mathrm{FP}_{\parallel}^{\mathrm{NP}}$. ▮

Note that the instance-solution relations of the first two problems in Theorem 4.8 are P-decidable but not hereditary, while the instance–solution relations of the third and fourth problems in Theorem 4.8 are NP-decidable and hereditary. The last problem in Theorem 4.8 is a concrete MAXP whose instance–solution relation is NP-decidable but not hereditary.

For those MAXPs Q whose instance–solution relations are coNP-decidable and hereditary, we are only able to show a loose lower bound.

PROPOSITION 4.9. *There is a MAXP* $Q=(D, R)$ *such that R is a* coNP-*decidable hereditary relation and Q is hard for* $\mathrm{FP}_{\parallel}^{\mathrm{NP}}$.

Proof. Let D be the set of all lists of boolean formulas. Define a relation R as follows: for all $\langle \varphi_1, \varphi_2, ..., \varphi_m\rangle \in D$ and all $b_1 b_2 \cdots b_m \in \Sigma^m$, $R(\langle \varphi_1, ..., \varphi_m\rangle, b_1 \cdots b_m)$ is true if and only if for each $1 \leqslant i \leqslant m$ with $b_i = 1$, φ_i is unsatisfiable. It is easy to verify that (D, R) satisfies the conditions in the proposition. ▮

In light of Theorem 3.5 (2), the following theorem shows that $\mathrm{FP}_{\parallel}^{\Sigma_2^{\mathrm{P}}}$ is a nearly optimal lower bound on the complexity of solving MAXPs whose instance–solution relations are coNP-decidable but not hereditary.

THEOREM 4.10. *There is a MAXP* $Q=(D, R)$ *such that R is* coNP-*decidable and Q is hard for* $\mathrm{FP}_{\parallel}^{\Sigma_2^{\mathrm{P}}}$.

Proof. We define D to be the set of all quadruples $\langle \alpha, \beta, M, \gamma\rangle$, where $\alpha = \langle y_1, ..., y_l\rangle$ is a list of strings, $\beta = \langle 1^{d_1}, ..., 1^{d_l}\rangle$ with each $d_i > 0$, $\gamma = \langle 1^{h_1}, ..., 1^{h_l}\rangle$ with each $h_i > 0$, and M is an NTM. For $\langle \alpha, \beta, M, \gamma\rangle \in D$ and $w \in \Sigma^*$, we define $R(\langle \alpha, \beta, M, \gamma\rangle, w)$ to be true if and only if $w = 0$ or w is a string $1^{k \cdot m} 1 b_1 \cdots b_l \langle z_1, ..., z_l\rangle 0^{(l-k) \cdot m}$ satisfying the following conditions:

(a) $b_i \in \{0, 1\}$ for all i with $1 \leqslant i \leqslant l$,

(b) k is the number of 1-bits in $b_1 b_2 \cdots b_l$,

(c) $m = 1 + l + |\langle z_1, ..., z_l\rangle|$,

(d) $|z_i| = d_i$ for all i with $1 \leqslant i \leqslant l$, and

(e) for all i with $1 \leqslant i \leqslant l$, if $b_i = 1$, then no branch of M on input $\langle y_i, z_i\rangle$ halts within h_i steps.

It is easy to see that $R \in \mathrm{coNP}$.

Let $F \in \mathrm{FP}_{\parallel}^{\Sigma_2^{\mathrm{P}}}$ witnessed by two functions f, g in FP and a set $L \in \Sigma_2^{\mathrm{P}}$. Since $L \in \Sigma_2^{\mathrm{P}}$, there must be a polynomial q and a polynomial-time bounded NTM M such that for every $y \in \Sigma^*$, $y \in L$ if and only if there is some string $z \in \Sigma^{q(|y|)}$ with $\langle y, z\rangle \notin L(M)$. Let r be a polynomial bounding the running time of M from above. Fix an arbitrary $x \in \Sigma^*$. Let $f(x) = \langle y_1, ..., y_l\rangle$, $d_i = q(|y_i|)$ and $h_i = r(|\langle y_i, 1^{d_i}\rangle|)$ for $1 \leqslant i \leqslant l$. (Here recall that the length of a pair is assumed to depend only on the lengths of its two arguments.) We set $\alpha = f(x)$, $\beta = \langle 1^{d_1}, ..., 1^{d_l}\rangle$, and $\gamma = \langle 1^{h_1}, ..., 1^{h_l}\rangle$. Then, by the definition of R, an arbitrary maximal solution of $\langle \alpha, \beta, M, \gamma\rangle$ in (D, R) together with the function g enables us to compute $F(x)$ in polynomial time. ▮

5. CHARACTERIZATIONS OF coNP, $\mathrm{D^P}$, AND Π_2^{P}

The following theorem gives us characterizations of coNP, $\mathrm{D^P}$, and Π_2^{P} via MAXPs.

THEOREM 5.1. *The following hold:*

(1) coNP *is the class of all sets L that can be expressed as* $L = \{x\colon f(x)$ *is a maximal solution of x in* $Q\}$ *for some* $f \in \mathrm{FP}$ *and some MAXP Q whose instance–solution relation is* P-*decidable.*

(2) $\mathrm{D^P}$ *is the class of all sets L that can be expressed as* $L = \{x\colon f(x)$ *is a maximal solution of x in* $Q\}$ *for some* $f \in \mathrm{FP}$ *and some MAXP Q whose instance–solution relation is* NP-*decidable.*

(3) $\mathrm{D^P}$ *is the class of all sets L that can be expressed as* $L = \{x\colon f(x)$ *is a maximal solution of x in* $Q\}$ *for some* $f \in \mathrm{FP}$ *and some MAXP Q whose instance–solution relation is* coNP-*decidable and hereditary.*

(4) Π_2^{P} *is the class of all sets L that can be expressed as* $L = \{x\colon f(x)$ *is a maximal solution of x in* $Q\}$ *for some* $f \in \mathrm{FP}$ *and some MAXP Q whose instance–solution relation is* coNP-*decidable.*

Proof. (1) Suppose $L \in \text{coNP}$. Then, there must exist a polynomial p and a polynomial-time bounded DTM M such that for every $x \in \Sigma^*$, $x \in L$ if and only if $\langle x, y \rangle \in L(M)$ for all $y \in \Sigma^{p(|x|)}$. We define $D = \Sigma^*$, and define R as follows: for every $x \in D$ and $w \in \Sigma^*$, $R(x, w)$ is true if and only if either $w = 0^{p(|x|)+1}$ or w is of the form $1y$ such that $y \in \Sigma^{p(|x|)}$ and $\langle x, y \rangle \notin L(M)$. Then, R is P-decidable. It is easy to see that for every $x \in \Sigma^*$, $x \in L$ if and only if $0^{p(|x|)+1}$ is a maximal solution of x in (D, R).

On the other hand, if $L = \{x: f(x)$ is a maximal solution of x in $Q\}$ for some $f \in \text{FP}$ and some MAXP Q whose instance–solution relation is P-decidable, then L is clearly contained in coNP.

(2) Suppose $L = L_1 \cap L_2$ for some $L_1 \in \text{NP}$ and $L_2 \in \text{coNP}$. We define $D = \Sigma^*$. For every $x \in D$ and $b_1 b_2 \in \Sigma^*$, we define $R(x, b_1 b_2)$ to be true if and only if the following two conditions are satisfied: (a) $b_1 = 0$ or $x \in L_1$ and (b) $b_2 = 0$ or $x \notin L_2$. Then, R is NP-decidable and hereditary. It is easy to see that for every $x \in \Sigma^*$, $x \in L$ if and only if 10 is a maximal solution of x in (D, R).

On the other hand, if $L = \{x: f(x)$ is a maximal solution of x in $Q\}$ for some $f \in \text{FP}$ and some MAXP Q whose instance–solution relation is NP-decidable, then L is clearly contained in D^{P}.

(3) Suppose $L = L_1 \cap L_2$ for some $L_1 \in \text{NP}$ and $L_2 \in \text{coNP}$. We define $D = \Sigma^*$. For every $x \in D$ and $b_1 b_2 \in \Sigma^*$, we define $R(x, b_1 b_2)$ to be true if and only if the following two conditions are satisfied: (a) $b_1 = 0$ or $x \notin L_1$ and (b) $b_2 = 0$ or $x \in L_2$. Then, R is coNP-decidable and hereditary. It is easy to see that for every $x \in \Sigma^*$, $x \in L$ if and only if 01 is a maximal solution of x in (D, R).

On the other hand, if $L = \{x: f(x)$ is a maximal solution of x in $Q\}$ for some $f \in \text{FP}$ and some MAXP Q whose instance-solution relation is coNP-decidable and hereditary, then L is clearly contained in D^{P}.

(4) The proof is very similar to that of (1). ∎

From the proof of Theorem 5.1(2) and Cook's theorem, it immediately follows that the decision problem associated with X-MaxModel is $\leqslant_m^{\text{P}}$-complete for D^{P}. In contrast to this, one can easily show that the decision problem associated with MaxModel is $\leqslant_m^{\text{P}}$-complete for coNP. The two facts exhibit a difference between X-MaxModel and MaxModel. For MaxSat, Max-k-Colorability, and MaxHamSubgraph, one can easily show that the decision problem associated with each of them is complete for D^{P} by giving a simple reduction from the SAT–UNSAT problem (see [PY82] for the definition of this problem).

6. CONCLUSION

In this paper, we have suggested a general framework for studying the complexity of solving maximality problems.

TABLE 1

The Complexity of solving MAXPs
(**M** denotes FNP//OptP[log n])

Condition for R	*Upper bound*	*Lower bound*
P-decidable and hereditary	FP	FP
P-decidable but not hereditary	**M**, $\text{RP} \cdot \text{FP}_{\parallel}^{\text{NP}}$	**M**
NP-decidable and hereditary	**M**, $\text{RP} \cdot \text{FP}_{\parallel}^{\text{NP}}$	**M**
NP-decidable but not hereditary	**M**, $\text{RP} \cdot \text{FP}_{\parallel}^{\text{NP}}$	**M**
coNP-decidable and hereditary	FP^{NP}	$\text{FP}_{\parallel}^{\text{NP}}$
coNP-decidable but not hereditary	$\text{RP} \cdot \text{FP}_{\parallel}^{\Sigma_2^{\text{P}}}$	$\text{FP}_{\parallel}^{\Sigma_2^{\text{P}}}$

Our results are summarized in Table 1 and Table 2. The results give, systematically, characterizations of several important complexity classes via MAXPs. An important consequence of the results is that the complexity of the problem X-MinModel is exactly captured by FNP//OptP[log n], giving an answer to an open question of Papadimitriou [Pap91].

As seen from Table 1, the complexity of solving those MAXPs whose instance–solution relations are coNP-decidable and hereditary is unclear. Two obvious open questions are whether the trivial upper bound FP^{NP} can be lowered and whether the trivial lower bound $\text{FP}_{\parallel}^{\text{NP}}$ can be raised. As a step toward the investigation of the two questions, we may first consider what is the complexity of solving MinUnsat (or other natural such problems). Although $\text{FP}_{\parallel}^{\text{NP}}$ is a loose lower bound, proving the $\text{FP}_{\parallel}^{\text{NP}}$-hardness of solving MinUnsat seems to be a hard task in the sense that at least the ideas of Papadimitriou and Wolfe [PW85] used in proving the D^{P}-hardness of the decision problem associated with MinUnsat do not work. Also, showing that MinUnsat is solvable in a class below FP^{NP} seems to need new ideas; at least, our ideas used in the proof of Theorem 3.5 do not seem to be applicable.

It would be also interesting to consider the complexity of MAXPs whose instance–solution relation is hereditary and NC-decidable. These MAXPs are obviously solvable in FP. Are they solvable in a class below FP or is there such a

TABLE 2

The Complexity of the Decision Problems
Associated with MAXPs

Condition for R	*Upper bound*	*Lower bound*
P-decidable and hereditary	P	P
P-decidable but not hereditary	coNP	coNP
NP-decidable and hereditary	D^{P}	D^{P}
NP-decidable but not hereditary	D^{P}	D^{P}
coNP-decidable and hereditary	D^{P}	D^{P}
coNP-decidable but not hereditary	Π_2^{P}	Π_2^{P}

MAXP Q that solving Q is complete for FP (say, under $\leqslant_{1\text{-T}}^{\text{NC}}$ reductions)? The two questions are important in parallel computation.

ACKNOWLEDGMENTS

The authors are grateful to Professor Christos H. Papadimitriou for helpful comments on this paper and for suggestions which improved the presentation. The first author thanks Professor Michio Oyamaguchi and Professor Yoshikatsu Ohta for their constant support.

Received July 16, 1993; final manuscript received February 23, 1994

REFERENCES

[And85] Anderson, R. (1985), A parallel algorithm for the maximal path problem, *in* "Proceedings, 17th ACM Symposium on Theory of Computing," pp. 33–37.

[CM87] Cai, J.-y., and Meyer, G. E. (1987), Graph minimal uncolorability is D^P-complete, *SIAM J. Comput.* **16**, 259–277.

[Che91] Chen, Z.-Z. (1991), A randomized NC algorithm for the maximal tree cover problem, *Inform. Process. Lett.* **40**, 241–246.

[CT91] Chen, Z.-Z., and Toda, S. (1991), On the complexity of computing optimal solutions, *Int. J. Found. Comput. Sci.* **2**, 207–220.

[CT93] Chen, Z.-Z., and Toda, S. (1993), The complexity of finding a minimal satisfying truth assignment to a Boolean formula, *in* "Proceedings, 3rd International Conference for Young Computer Scientists," pp. 637–640.

[GJ79] Garey, M., and Johnson, D. S. (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, San Francisco.

[KW84] Karp, R. M., and Wigderson, A. (1984), A fast parallel algorithm for the maximal independent set problem, *in* "Proceedings, 16th ACM Symposium on Theory of Computing," pp. 266–272.

[KUW88] Karp, R. M., Upfal, E., and Wigderson, A. (1988), The complexity of parallel search, *J. Comput. System Sci.* **36**, 225–253.

[KT90] Kobler, J., and Thierauf, T. (1990), Complexity classes with advice, *in* "Proceedings, 5th IEEE Conference on Structure in Complexity Theory," pp. 305–315.

[Kre88] Krentel, M. W. (1988), The complexity of optimization problems, *J. Comput. System Sci.* **36**, 490–509.

[Lub85] Luby, M. (1985), A simple parallel algorithm for the maximal independent set problem, *in* "Proceedings, 17th ACM Symposium on Theory of Computing," pp. 1–10.

[MVV87] Mulmuley, K., Vazirani, U. V., and Vazirani, V. V. (1987), Matching is as easy as matrix inversion, *in* "Proceedings, 19th ACM Symposium on Theory of Computing," pp. 345–354.

[Pap91] Papadimitriou, C. H. (1991), On selecting a satisfying truth assignment, *in* "Proceedings, 32nd IEEE Symposium on the Foundations of Computer Science," pp. 163–169.

[Pap92] Papadimitriou, C. H. (1992), Personal communication.

[PW85] Papadimitriou, C. H., and Wolfe, D. (1985), The complexity of facets resolved, *in* "Proceedings, 26nd IEEE Symposium on the Foundations of Computer Science," pp. 74–78.

[PY82] Papadimitriou, C. H., and Yannakakis, M. (1982), The complexity of facets (and some facets of complexity), *J. Comput. System Sci.* **28**, 244–259.