<table>
<thead>
<tr>
<th>Title</th>
<th>The Complexity of Selecting Maximal Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Chen, Zhi-Zhong; Toda, Seinosuke</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (1993), 833: 11-21</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1993-04</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/83421">http://hdl.handle.net/2433/83421</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>

Kyoto University
The Complexity of Selecting Maximal Solutions

Zhi-Zhong Chen
Department of Information Engineering, Mie University, Tsu-ishi, Mie 514, Japan.

Seinosuke Toda
Department of Computer Science, University of Electro-Communications,
Chofu-shi, Tokyo 182, Japan.

1 Introduction

Intuitively, a maximization problem is to select a maximal solution for a given input according to some selection criterion. The maximal independent set problem (MIS) [5] and the minimal unsatisfiability problem (MinUnsat) [11] are two standard examples of such problems. Much work has been devoted to the study of maximization problems [1,2,3,4,5,7,9,11,12]. Most of the previous work has involved studying specific maximization problems and either finding an efficient algorithm (e.g., [5]) or proving the problem is hard to solve (e.g., [11]). An attractive alternative approach is to study maximization problems in a general framework and to prove general results.

In this paper, we formalize a maximization problem (MAXP) $Q$ as a pair $(D, R)$, where $D$ is the set of instances and $R : D \times \{0, 1\}^* \rightarrow \{\text{true, false}\}$ 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 $y$ such that $R(x, y)$ is true but changing one or more arbitrary 0-bits of $y$ to 1-bits will change the value of $R(x, y)$ to false. As an example, consider MIS in our framework. For it, $D$ 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, \cdots, $n$) and $\{i : b_i = 1\}$ is an independent set in $G$. Our goal is to demonstrate what factors make $Q$ easy or hard to solve and how the factors influence the complexity of solving $Q$. 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 [5,7] while solving MinUnsat is $D^P$-hard [11]. The other factor is whether $R$ is hereditary or not, where $R$ 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 one or more arbitrary 1-bits of $w$ are changed to 0-bits. The instance-solution relation of MIS (also MinUnsat) is hereditary. In [9], Papadimitriou considered the following problem (MinModel): Given a CNF boolean formula $\phi$, find a satisfying truth assignment $\vec{a}$ to $\phi$ such that changing one or more arbitrary 1-bits of $\vec{a}$ to 0-bits will make $\vec{a}$ no longer satisfy $\phi$. 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 restrict to consider only those MAXP’s whose instance-solution relation is decidable in NP or coNP. We first consider upper bounds on the complexity of solving such MAXP’s. 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 NPMV//OptP[O(log $n$)] if $R$ is decidable in NP; (iii) $Q$ is solvable in FP$^{NP}$ if $R$ is decidable in coNP and hereditary; (iv) $Q$ is solvable in FP$^{P^F}$ 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 $\varepsilon$ be an arbitrary polynomial. Then, there exist a function $F \in \text{FP}_{||}^{\Sigma_{2}^{P}}$ and a polynomial $p$ such that for every $x$, $\Pr[F(x, w)] = 1 - 2^{-\varepsilon(|x|)}$, where $w \in \{0, 1\}^{|x|}$ is randomly chosen under uniform distribution.

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

(v) and (vi) are shown by extending the technique used in [3].

We then show that NPMV/\text{OptP}[O(\log n)] is also a lower bound for solving those MAXP’s whose instance-solution relation is decidable in NP or is decidable in P but not hereditary, and that FP$^{\Sigma_{2}^{P}}_{||}$ is also a lower bound for solving those MAXP’s whose instance-solution relation is decidable in coNP but not hereditary. Combining the upper and lower bounds, we obtain characterizations of NPMV/\text{OptP}[O(\log n)] and FP$^{\Sigma_{2}^{P}}_{||}$ via MAXP’s. As an important consequence of the characterization of NPMV/\text{OptP}[O(\log n)] we obtain the first natural complete problem for NPMV/\text{OptP}[O(\log n)]. The problem (called $X$-$\text{MinModel}$) is defined as follows: Given a CNF boolean formula $\phi$ and a subset $X$ of the set of variables in $\phi$, find a satisfying truth assignment $\vec{a}$ to $\phi$ such that changing one or more arbitrary 1-bits of $\vec{a}$ corresponding to variables in $X$ to 0-bits will make $\vec{a}$ no longer satisfy $\phi$. $X$-$\text{MinModel}$ was first considered by Papadimitriou in [9], and was claimed without a precise proof to be $\Delta_{2}^{P}$-complete there. However, Papadimitriou later withdrew his claim and thus left the complexity of $X$-$\text{MinModel}$ open [10]. In [3], we proved that the complexity of $X$-$\text{MinModel}$ is roughly captured by FP$^{\text{NP}}_{||}$. Now, the results in this paper give, for the first time, the exact complexity of solving $X$-$\text{MinModel}$.

We also characterize complexity classes of sets via MAXP’s. The following are shown:

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

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

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

(d) IP$_{3}^{P}$ is the class of all sets $L$ that can be expressed as $L = \{x : f(x) \text{ is a maximal solution of } x \text{ in } Q\}$ for some $f \in \text{FP}$ and some MAXP $Q$ whose instance-solution relation is coNP-decidable.

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

2 Preliminaries

We use $\Sigma = \{0, 1\}$ as our alphabet. By a set, we mean a subset of $\Sigma^{*}$. Similarly, by a string, we mean a string in $\Sigma^{*}$. 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, \ldots, $n$th bits, respectively. For a finite string $x$, we usually identify $x$ with the set of all indices $i$ such that the $i$th 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 $x \subseteq y$. A maximal string in a set $S$ of finite strings is a string in $S$ that is not smaller than any other string in $S$.

We assume a one-to-one pairing function from $\Sigma^* \times \Sigma^*$ to $\Sigma^*$ 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 denote 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 $L$, $\overline{L}$ denotes its complement (i.e., $\Sigma^* - L$), and $\chi_L$ denotes the characteristic function of $L$. For a class $C$ of sets, $coC$ denotes the class of all sets whose complement is in $C$. Let $\Sigma^n$ denote the set of all strings with length $n$. For two sets $L_1$ and $L_2$, $L_1 \oplus L_2$ denotes the set $\{x : x \in L_1\} \cup \{y : y \in L_2\}$.

All functions considered here are ones from $\Sigma^*$ to $\Sigma^* \cup \{\#\}$. The symbol $\#$ is assumed to be not in $\Sigma^*$. 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 $x \in \Sigma^n$, $T(x) = \emptyset$ 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: $P$, $NP$, $coNP$, $\Sigma^p_2$, $\Pi^p_2 = co\Sigma^p_2$. Let $D^p = \{L_1 \cap L_2 : L_1 \in NP \text{ and } L_2 \in coNP\}$. $FP$ denotes the class of all functions computed by polynomial-time bounded DTM's. 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 $A$. $FP^A_\parallel$ 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 to the oracle $A$. More precisely, a function $F$ is in $FP^A_\parallel$ if there exist two functions $f$ and $g$ in $FP$ such that for all strings $x$, $F(x) = g(x, \chi_A(y_1) \cdots \chi_A(y_m))$, where $f(x) = \langle y_1, \cdots, y_m \rangle$. For a class $C$ of sets, $FP^C = \cup_{A \in C} FP^A$ and $FP^C_\parallel = \cup_{A \in C} FP^A_\parallel$.

An NP metric Turing machine is a polynomial-time bounded NTM $T$ such that on every input, every branch of $T$ outputs a binary number and halts [6]. OptP$[O(\log n)]$ denotes the class of all (total) integer-valued functions $H$ for which there exist a polynomial $p$ and an NP metric Turing machine $T$ such that for every $x$, $H(x) \leq p(|x|)$ and
$H(x)$ equals to the maximum number in $T(x)$. NPMV//OptP[O($\log n$)] denotes the class of all (partial) multi-valued functions $G$ for which there exist an NTM $T$ and a function $H \in \text{OptP}[O(\log n)]$ such that for every $x$, $G(x) = T(\langle x, H(x) \rangle)$.

A maximization problem (MAXP) $Q$ is a pair $(D, R)$, where (i) $D$ is the set of instances and (ii) $R : D \times \Sigma^* \rightarrow \{\text{true}, \text{false} \}$ 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 compute, given an instance $x \in D$, a maximal solution of $x$.

Each MAXP $Q = (D, R)$ considered in this paper is required to 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.

**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 $H$ of (single-valued or multi-valued) functions if some $H \in H$ solves $Q$.

**Definition 2.2** Let $F$ be a function, and let $G$ be a multi-valued function. Then, $F$ (resp., $G$) is reducible to $Q$ if there exist two functions $f$, $g$ in FP such that for every $x$, $f(x) \in D$ and $g(x, w) = F(x)$ (resp., $g(x, w) \in G(x)$) for every maximal solution $w$ of $f(x)$ in $Q$. $Q$ is hard for a class $H$ of (single-valued or multi-valued) functions if every $H \in H$ is reducible to $Q$. $Q$ is complete for a class $H$ of (single-valued or multi-valued) functions if $Q$ is solvable in and hard for $H$. $Q$ is hard for a class $C$ of sets if $Q$ is hard for the class $\{\chi_L : L \in C\}$.

**Definition 2.3** The set $L_Q = \{(x, w) : w \text{ is a maximal solution of } x \text{ in } Q\}$ is called the decision problem associated with $Q$.

### 3 Upper bounds

In this section, we show upper bounds on the complexity of solving MAXP’s. The following 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 NPMV//OptP[O($\log n$)].
3. If $R$ is hereditary and coNP-decidable, then $Q$ is solvable in FP$^{\text{coNP}}$.
4. If $R$ is coNP-decidable, then $Q$ is solvable in FP$^{\text{coNP}}$.

We next proceed to show two other non-trivial upper bounds. To do this, we need several definitions and a known result.

**Definition 3.1** Let $F$ be a class of functions. Then we define a class RP-$F$ of...
multi-valued functions as follows: A multi-valued function $G$ is in \text{RP}\text{-}\text{F} if for every polynomial $e$, there exist a function $F \in \text{F}$ 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 $G(x)$ is undefined and (b) $\Pr[F(x, w) \in G(x) \cup \{\#\}] = 1$ and $\Pr[F(x, w) \in G(x)] \geq 1 - 2^{-c(|x|)}$ otherwise, where $w$ is a random string chosen from $\{0, 1\}^{p(|x|)}$. Intuitively speaking, $G$ is in \text{RP}\text{-}\text{F} if for every string $x$, we can randomly pick up an element of $G(x)$ using a function in $\text{F}$.

**Notation:** For $k \geq 1$, $[1, k]$ denotes the set of all integers $i$ with $1 \leq i \leq k$.

**Definition 3.2** 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 $s \in S$, $f(s)$ is in $[1, k]$. 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 $[1, k]$. The weight of a subset $S'$ of $S$ under a weight function $f$ is $\Sigma_{s \in S'} f(s)$. Note that for every $k$-weight function over $S$, the weight of each subset of $S$ under $f$ is no more than $k||S||$ and that the empty set $\emptyset$ is the unique subset of $S$ with weight $0$.

**Lemma 3.1** [8]. Let $S$ be a nonempty family of subsets of a finite set $S$. Then, for any random $k$-weight function $f$ over $S$ with $k \geq 2||S||$, $\Pr[\text{There is a unique maximum weight set in } S \text{ under } f] \geq \frac{1}{2}$.

Now we are ready to show the two non-trivial upper bounds. The idea used in the proof is a generalization of the one used in [3].

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

1. If $R$ is NP-decidable, then $Q$ is solvable in $\text{RP}\cdot\text{FP}^{\Sigma_2^P}$.
2. If $R$ is coNP-decidable, then $Q$ is solvable in $\text{RP}\cdot\text{FP}^{\Sigma_2^P}$.

**Proof.** We only show a proof for (2). (1) can be shown in a similar manner.

(2) We first explain the idea behind the proof. Let $p_Q$ be a polynomial such that for all $x \in D$, the length of each solution of $x$ is no more than $p_Q(|x|)$. Let $x$ be an instance of $Q$. Then, we consider $S$, the family of all solutions of $x$ with maximum length. To find a maximal solution for $x$, we first get a random $2p_Q(|x|)$-weight function $f$ over $[1, p_Q(|x|)]$. Then, by Lemma 3.1, with probability at least $\frac{1}{2}$, there is a unique solution in $S$ of maximum weight. To find this unique solution of maximum weight, it suffices to ask only one round of parallel queries to a $\Sigma_2^P$ oracle set. Since the weight assigned to each element of $[1, p_Q(|x|)]$ is positive, all maximum weight solutions are maximal solutions (but not necessarily solutions of maximum 1-bits). In order to get the high probability of success, we may perform several copies of this computation in parallel.

We now proceed to give the precise proof. Let $p_Q$ be a polynomial that bounds the lengths of solutions of $x$ from above. For convenience, let $n_x = p_Q(|x|)$ for all $x \in D$. Then we define five sets as follows:

$$L_x = \{x : x \text{ has a solution}\},$$

$$B_1 = \{(x, i) : 0 \leq i \leq n_x \text{ and } x \text{ has a solution of length } i\},$$

$$B_2 = \{(x, i, f, j) : x \in D, 0 \leq i \leq n_x, f \text{ is a } 2^{1+\lceil \log_2 n_x \rceil}\text{-weight function over } [1, n_x], 0 \leq j \leq \left\lfloor 2^{1+\log_2 n_x} \right\rfloor, \text{ and } x \text{ has a solution } u \text{ such that } |u| = i \text{ and } j \text{ is the weight of } u \text{ under } f\},$$

$$B_3 = \{(x, i, f, j) : x \in D, 0 \leq i \leq n_x, f \text{ is a } 2^{1+\lceil \log_2 n_x \rceil}\text{-weight function over } [1, n_x],$$
$0 \leq j \leq i2^{1+\lceil \log_2 n_x \rceil}$, and $x$ has two or more solutions $u$ such that $|u| = i$ and $j$ is the weight of $u$ under $f$), and

$$B_4 = \{(x, i, f, j, k) : x \in D, 0 \leq i \leq n_x, f \text{ is a } 2^{1+\lceil \log_2 n_x \rceil}, \text{ weight function over } [1, n_x],$$

$$0 \leq j \leq i2^{1+\lceil \log_2 n_x \rceil}, 1 \leq k \leq i, \text{ and } x \text{ has a solution } u \text{ such that } |u| = i, j \text{ is the weight of } u \text{ under } f, \text{ and the } k\text{th bit of } u \text{ is 1}\}$.

Obviously, $L_R, B_1, B_2, B_3$, and $B_4$ are in $\Sigma^P_2$. Let $B = (((L_R \oplus B_1) \oplus B_2) \oplus B_3) \oplus B_4)$. Then, $B \in \Sigma^P_2$.

Let $e$ be an arbitrary polynomial. We define a polynomial $p$ as follows: $p(i) = e(i) \cdot (p_0^2(i) + p_Q(i))$. Below, we define a DOTM $T$ which uses $B$ as an oracle set. Given an input $\langle x, w \rangle$ with $x \in D$ and $w \in \{0, 1\}^{p(|x|)}$, $T$ operates as follows:

Step 1: $T$ checks whether $x$ has a solution by asking a query to $L_R$. If $x$ has no solution, then $T$ halts by entering a rejecting state.

Step 2: $T$ finds $n_x$, the length of the longest solutions of $x$. This is done by asking the queries $\langle x, 0 \rangle, \langle x, 1 \rangle, \ldots, \langle x, n_x \rangle$ to the oracle set $B_1$.

Step 3: $T$ computes $n_2 = 2^{1+\lceil \log_2 n_x \rceil}$ and constructs, from $w$, $n_2$-weight functions $f_1$, $f_2$, $\cdots$, $f_{e(|x|)}$ over the set $[1, n_x]$ as follows:

Step 3.1: $T$ first computes $e(|x|)$ strings $w_1, \ldots, w_{e(|x|)}$ from $w$ such that $|w_1| = \cdots = |w_{e(|x|)}| = n_x \log_2 n_2$ and the string $w_1w_2\cdots w_{e(|x|)}$ is a prefix of $w$ (the remaining part of $w$ is ignored), and then for each $1 \leq k \leq e(|x|)$, it partitions $w_k$ into $n_2$ substrings $w_{k,1}, \ldots, w_{k,n_2}$ each of length $\log_2 n_2$. (Note: $T$ can do this because $n_2^2 + n_x \geq n_x \log_2 n_2$.)

Step 3.2: For each $1 \leq k \leq e(|x|)$ and each $l$ in $[1, n_x]$, $T$ sets $f_k(l) = d_{k,l} + 1$, where $d_{k,l}$ is the integer whose binary representation is $w_{k,l}$.

Step 4: For each $1 \leq k \leq e(|x|)$, $T$ computes the maximum number $m_k$ with $\langle x, n_1, f_k, m_k \rangle \in B_2$. This is done by asking the queries $\langle x, i, f_k, j \rangle$ with $0 \leq i \leq n_x$, $1 \leq k \leq e(|x|)$, and $0 \leq j \leq in_2$ to the oracle set $B_2$. (Note: In this step, $T$ asks the queries of the form $\langle x, i, f_k, j \rangle$ for all possible values of $i$, $k$, and $j$ because the machine needs to prepare all queries independently of each other.)

Step 5: For $1 \leq k \leq e(|x|)$ and $1 \leq l \leq n_1$, $T$ computes $a_{k,l} = \chi_{B_4}(\langle x, n_1, f_k, m_k, l \rangle)$. This is done by asking the queries $\langle x, i, f_k, j, l, l \rangle$ with $0 \leq i \leq n_x$, $1 \leq k \leq e(|x|)$, $0 \leq j \leq in_2$, and $1 \leq l \leq i$ to the oracle set $B_4$. (Note: In this step, $T$ asks the queries of the form $\langle x, i, f_k, j, l \rangle$ for all possible values $i$, $k$, $j$, and $l$ because the machine needs to prepare all queries independently of each other.)

Step 6: For each $1 \leq k \leq e(|x|)$, $T$ checks whether $\langle x, n_1, f_k, m_k \rangle \in B_3$. This is done by asking the queries $\langle x, i, f_k, j \rangle$ with $0 \leq i \leq n_x$, $1 \leq k \leq e(|x|)$, and $0 \leq j \leq in_2$ to the oracle set $B_3$. If for some $k$, $\langle x, n_1, f_k, m_k \rangle \notin B_3$, then $T$ outputs $a_{k,1}a_{k,2}\cdots a_{k,n_1}$ and halts; otherwise, $T$ outputs the special symbol \# and halts.

Let $F$ denote the function computed by $T$ with oracle $B$. We can easily see that $T$ is polynomial-time bounded and all query strings are prepared independently of each other; this means that the query strings made by $T$ on input $\langle x, w \rangle$ can be realized as parallel queries to the oracle set $B$. Thus, $F$ is in $\mathcal{FP}^P_\parallel$.

Let $G$ be a multi-valued function defined by $G(x) = \{F(x, w) : w \in \{0, 1\}^{p(|x|)}$ and $F(x, w)$ is defined\} $- \{\#\}$. We show that $G$ solves $Q$ and is in $\mathcal{RP} \cdot \mathcal{FP}^P_\parallel$. To this end, we first prove two claims.

Claim 1 Suppose that for some $k$ with $1 \leq k \leq e(|x|)$, $x$ has a unique solution with
length $n_1$ and of weight $m_k$ under $f_k$. Then, the string $a_{k,1}a_{k,2}\cdots a_{k,n_1}$ output by $T$ is a maximal solution of $x$.

**Claim 2** Suppose that $x$ has solutions and $w$ is randomly chosen from $\{0,1\}^{\lceil e(|x|) \rceil}$. Then, $\Pr[F(x, w) \text{ is a maximal solution of } x] \geq 1 - 2^{-e(|x|)}$.

**Proof.** Since $w$ is randomly chosen from $\{0,1\}^{\lceil e(|x|) \rceil}$, the functions $f_1, f_2, \cdots, f_{e(|x|)}$ constructed in Step 3 must be random $n_2$-weight functions over $[1, n_2]$. Note that $n_2 = 2^{1+\lceil \log_2 n_x \rceil} \geq 2n_x$. Thus, from Claim 1 and Lemma 3.1, we have that

$$\Pr[F(x, w) \text{ is a maximal solution of } x] = \Pr[(\exists k, 1 \leq k \leq e(|x|)) a_{k,1}a_{k,2}\cdots a_{k,n_1} \text{ is a maximal solution of } x] \geq \Pr[(\exists k, 1 \leq k \leq e(|x|)) \text{ $x$ has a unique solution with maximum length and of maximum weight under } f_k] \geq 1 - \Pi_{k=1}^{e(|x|)} \Pr[x \text{ has two or more solutions with maximum length and of maximum weight under } f_k] \geq 1 - \left(\frac{1}{2}\right)^{e(|x|)} = 1 - 2^{-e(|x|)}.$$

In the case when $x$ has no solution, $F(x, w)$ is undefined for all $w \in \{0,1\}^{\lceil e(|x|) \rceil}$ by Step 1 and hence $G(x) = \emptyset$. On the other hand, when $x$ has solutions, $G(x) \neq \emptyset$ by Claim 2 and the definition of $G$, and each element of $G(x)$ is a maximal solution of $x$ by Claim 1. Therefore, $G$ solves $Q$.

If $G(x) = \emptyset$, we know that $x$ has no solution by the discussions in the last paragraph and thus that $F(x, w)$ is undefined for all $w \in \{0,1\}^{\lceil e(|x|) \rceil}$ by Step 1. On the other hand, if $G(x) \neq \emptyset$, then $x$ has solutions by the discussions in the last paragraph, $\Pr[F(x, w) \in G(x) \cup \{\#\}] = 1$ by Step 6 and the definition of $G$, and $\Pr[F(x, w) \in G(x)] \geq 1 - 2^{-e(|x|)}$ by Claim 2. Therefore, $G$ is in RP-FP$^\Sigma_2^P$.

## 4 Hardness of solving MAXP's

In the light of Proposition 3.1(1), the following proposition shows that FP is a tight lower bound on the complexity of solving MAXP's whose instance-solution relation is P-decidable and hereditary.

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

By Proposition 3.1(2), the following theorem shows that NPMV//OptP[$O(\log n)$] is a tight lower bound on the complexity of solving MAXP's whose instance-solution relation is P-decidable but not hereditary.

**Theorem 4.1** There is a MAXP $Q = (D, R)$ such that $R$ is P-decidable (but not hereditary) and $Q$ is hard for NPMV//OptP[$O(\log n)$].

The following corollary is immediate from the proof of Theorem 4.1.

**Corollary 4.1** The following problem (called X-MaxModel hereafter) is complete for NPMV//OptP[$O(\log n)$]:

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

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

$X$-MaxModel is essentially the same problem as considered by Papadimitriou in Section 3 of [9]. In [9], Papadimitriou claimed without a precise proof that the problem is complete for $\text{FP}^\text{NP}$. However, he later withdrew his claim and thus left the complexity of the problem open [10]. In [3], we proved that the complexity of the problem is roughly captured by $\text{FP}^\text{NP}$. Now, Corollary 4.1 gives, for the first time, the exact complexity of the problem. Corollary 4.1 is also of special interest in the sense that no natural problem complete for $\text{NPMV}/\text{OptP}[O(\log n)]$ has been shown before.

By modifying the proof of Theorem 4.1, we can show that $\text{NPMV}/\text{OptP}[O(\log n)]$ is a tight lower bound on the complexity of solving MAXP's whose instance-solution relation is NP-decidable and hereditary.

**Theorem 4.2** There is a MAXP $Q = (D, R)$ such that $R$ is NP-decidable and hereditary and $Q$ is hard for $\text{NPMV}/\text{OptP}[O(\log n)]$.

The instance-solution relation of $X$-MaxModel is NP-decidable but not hereditary. A natural question arises: Are there natural MAXP's $Q$ such that $Q$ is hard for $\text{NPMV}/\text{OptP}[O(\log n)]$ and the instance-solution relation of $Q$ is either NP-decidable and hereditary or P-decidable (but not hereditary)? Unfortunately, we are unable to settle this question. However, we below show that the question will have a positive answer if $\text{NPMV}/\text{OptP}[O(\log n)]$ in it is replaced by $\text{FP}^\parallel$.

**Definition 4.1** A MAXP $Q = (D, R)$ is paddable if there are two functions $f$ and $g$ in FP such that for every list $I = (x_1, x_2, \ldots, x_m)$ of instances of $Q$, $f(I) \in D$ and for every maximal solution $w$ of $f(I)$, $g(I, w)$ gives a maximal solution for each $x_i$.

**Lemma 4.1** If a MAXP is paddable and hard for NP, then it is hard for $\text{FP}^\parallel$.

**Theorem 4.3** The following MAXP's are hard for $\text{FP}^\parallel$.

1. **MAXIMAL MODEL** (MaxModel)
   Instance: A CNF boolean formula $\phi$.
   Output: A maximal satisfying truth assignment to $\phi$, i.e., a satisfying truth assignment $\vec{a}$ to $\phi$ such that there is no other satisfying truth assignment $\vec{b}$ to $\phi$ 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 $\phi = \{C_1, C_2, \ldots, C_m\}$.
   Output: A maximal subset $\phi'$ of $\phi$ that is satisfiable.

4. **MAXIMAL $k$-COLORABILITY** ($k \geq 3$) (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 $(G, w)$ 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.

We here note that a different proof for the $\text{FP}^\text{NP}_{\parallel}$-hardness of MaxModel has been given in [3]. Note that the instance-solution relations of the first two problems in Theorem 4.3 are $P$-decidable but not hereditary, while the instance-solution relations of the third and fourth problems in Theorem 4.3 are $\text{NP}$-decidable and hereditary. The last problem in Theorem 4.3 is a concrete MAXP whose instance-solution relation is $\text{NP}$-decidable but not hereditary.

For those MAXP’s $Q$ whose instance-solution relation is $\text{coNP}$-decidable and hereditary, we are only able to show a loose lower bound.

**Proposition 4.2** There is a MAXP $Q = (D, R)$ such that $R$ is a $\text{coNP}$-decidable hereditary relation and $Q$ is hard for $\text{FP}^\text{NP}_{\parallel}$.

In the light of Theorem 3.1(2), the following theorem shows that $\text{FP}^\text{P}_{\parallel}$ is a nearly optimal lower bound on the complexity of solving MAXP’s whose instance-solution relation is $\text{coNP}$-decidable but not hereditary.

**Theorem 4.4** There is a MAXP $Q = (D, R)$ such that $R$ is $\text{coNP}$-decidable and $Q$ is hard for $\text{FP}^\Sigma^\text{P}_{2}$.

5 **Characterizations of coNP, $D^P$ and $\Pi^P_2$**

The following proposition can be easily proved.

**Proposition 5.1** Let $Q = (D, R)$ be a MAXP.

1. If $R$ is $P$-decidable and hereditary, then $L_Q$ is in $P$.
2. If $R$ is $P$-decidable, then $L_Q$ is in $\text{coNP}$.
3. If $R$ is $\text{NP}$-decidable, then $L_Q$ is in $D^P$.
4. If $R$ is $\text{coNP}$-decidable and hereditary, then $L_Q$ is in $D^P$.
5. If $R$ is $\text{coNP}$-decidable, then $L_Q$ is in $\Pi^P_2$.

Similar to Proposition 4.1, we can simply show that $P$ is a tight lower bound on the complexity of $L_Q$ for MAXP’s $Q$ whose instance-solution relation is $P$-decidable and hereditary.

The following theorem gives us characterizations of coNP, $D^P$, and $\Pi^P_2$ via MAXP’s.

**Theorem 5.1** The following hold:

1. A set $L$ is in $\text{coNP}$ if and only if it can be expressed as $L = \{x : f(x) \text{ is a maximal solution of } x \text{ in } Q\}$ for some $f \in \text{FP}$ and some MAXP $Q$ whose instance-solution relation is $P$-decidable.
2. A set $L$ is in $D^P$ if and only if it can be expressed as $L = \{x : f(x) \text{ is a maximal solution of } x \text{ in } Q\}$ for some $f \in \text{FP}$ and some MAXP $Q$ whose instance-solution relation is $\text{NP}$-decidable (and hereditary).
3. A set $L$ is in $D^P$ if and only if it can be expressed as $L = \{x : f(x) \text{ is a maximal solution of } x \text{ in } Q\}$ for some $f \in \text{FP}$ and some MAXP $Q$ whose instance-solution relation is $\text{coNP}$-decidable and hereditary.
of the proof of Theorem 5.1(1), we easily see that there is a MAXP whose instance-solution relation is P-decidable (but not hereditary) and whose associated decision problem is $\leq_{m}^{P}$-complete for coNP. However, the following proposition gives us two concrete such MAXP's.

**Proposition 5.2** The decision problems associated with MaxModel and MaxCubSubgraph are $\leq_{m}^{P}$-complete for coNP:

The following corollary follows immediately from the proof of Theorem 5.1(2) and Cook's theorem.

**Corollary 5.1** The decision problem associated with X-MaxModel is $\leq_{m}^{P}$-complete for $D^{P}$.

We next show three natural MAXP's whose instance-solution relations are in NP and whose associated decision problems are $\leq_{m}^{P}$-complete for $D^{P}$.

**Proposition 5.3** The decision problems associated with MaxSat, Max-k-Colorability and MaxHamSubgraph are $\leq_{m}^{P}$-complete for $D^{P}$:

We next show a natural MAXP whose instance-solution relation is coNP-decidable and hereditary and whose associated decision problem is $\leq_{m}^{P}$-complete for $D^{P}$. Other such natural MAXP's may be found in [2,11,12].

**Proposition 5.4** The following problem is $\leq_{m}^{P}$-complete for $D^{P}$:

**Instance:** A triple $(\phi, X, \vec{a})$, where $\phi$ is a CNF boolean formula, $X$ is a set of variables appearing only positively in $\phi$, and $\vec{a}$ is a truth assignment to the variables in $X$.

**Question:** Is it the case that $\vec{a}$ has no extension satisfying $\phi$ but each $\vec{b} \in \Sigma^{||X||}$ with $\vec{a} \subset \vec{b}$ has an extension satisfying $\phi$?

6 Conclusion

In this paper, we have suggested a general framework for studying the complexity of solving maximization problems. Our results are summarized in Table 1 and Table 2. The results give, systematically, characterizations of several important complexity classes via MAXP's. An important consequence of the results is that the complexity of the problem X-MinModel is exactly captured by NPMV//OptP[$O(\log n)$], giving an answer to an open question of Papadimitriou [9].

As seen from Table 1, the complexity of solving those MAXP's whose instance-solution relation is coNP-decidable and hereditary is unclear. Two obvious open questions are to ask whether the trivial upper bound $FP_{NP}$ can be lowered and to ask whether the trivial lower bound $FP_{||NP}$ can be raised. As a step toward the investigation of these questions, we may first consider what is the complexity of solving MinUnsat (or other natural such problems). Although $FP_{||NP}$ is a loose lower bound, proving the $FP_{||NP}$-hardness of solving MinUnsat seems to be a hard task in the sense that at least the ideas used in proving the
$D^P$-hardness of the decision problem associated with MinUnsat do not work [11]. Also, showing that MinUnsat is solvable in a class below $FP^{NP}$ needs new ideas; at least, our ideas used in the proof of Theorem 3.1 do not seem to be applicable.

It would be also interesting to consider the complexity of MAXP's whose instance-solution relation is $C$-decidable and hereditary for some complexity class $C$ below $P$. These MAXP's are obviously solvable in FP. Are they solvable in a class below FP or is there such a MAXP $Q$ that solving $Q$ is complete for FP (say, under $\leq_{NC^T}$ reductions)? The two questions are important in parallel computation in the case when $C \subseteq NC$.

References


