Functions Computable with Nonadaptive Queries to NP

Harry Buhrman [*] CWI Amsterdam

Jim Kadin [†] University of Maine

Thomas Thierauf [‡] Universität Ulm

Abstract

We study $FP_{\|}^{NP}$, the class of functions that can be computed in polynomial time with nonadaptive queries to an NP oracle. This is motivated by the question of whether it is possible to compute witnesses for NP sets within $FP_{\|}^{NP}$. The known algorithms for this task all require sequential access to the oracle. On the other hand, there is no evidence known yet that this should not be possible with parallel queries.

We define a class of optimization problems based on NP sets, where the optimum is taken over a polynomially bounded range (NPbOpt). We show that if such an optimization problem is based on one of the *known* NP-complete sets, then it is hard for $FP_{\|}^{NP}$. Moreover, we will *characterize* $FP_{\|}^{NP}$ as the class of functions that reduces to such optimization functions. We will call this property *strong hardness*.

The main question is whether these function classes are complete for $FP_{\|}^{NP}$. That is, whether it is possible to compute an optimal value for a given optimization problem in $FP_{\|}^{NP}$. We show that these

[*]CWI. PO Box 94079, 1090 GB Amsterdam, The Netherlands. E-mail: buhrman@cwi.nl. Part of this research was done while visiting the Univ. Politècnica de Catalunya in Barcelona. Partially supported by the Dutch foundation for scientific research (NWO) through NFI Project ALADDIN, under contract number NF 62-376. Part of this research was done while visiting Boston University with the support of NSF Grant CCR-8814339

[†]Department of Computer Science, University of Maine, 5752 Neville Hall, Orono, Maine 04469-5752.

[‡]Abteilung Theoretische Informatik, Universität Ulm, Oberer Eselsberg, 89069 Ulm, Germany. E-mail: thierauf@informatik.uni-ulm.de. Part of the work done while visiting the Department of Computer Science at the University of Rochester, Rochester, NY and the Department of Computer Science at the University of Maine, Orono. Supported in part by DFG Postdoctorial Stipend Th 472/1-1 and NSF grant CCR-8957604 and DAAD Acciones Integradas.

optimization problems are complete for $\mathrm{FP}_{\|}^{\mathrm{NP}}$, if and only if one can compute membership proofs for NP sets in $\mathrm{FP}_{\|}^{\mathrm{NP}}$. This indicates that the completeness question is a hard one.

1 Introduction

A fundamental issue in the study of NP and related classes is the complexity of generating proofs that a string is a member of a given language. For NP-complete sets, it is well known that the lexicographically smallest witness of membership can be generated in $\mathrm{FP}^{\mathrm{NP}}$, the class of functions computable in polynomial time with access to an oracle in NP [Va76]. For example, if we consider the NP-complete set SAT, the following function is in $\mathrm{FP}^{\mathrm{NP}}$.

$$f_{left}(\varphi) = \begin{cases} \text{the lexicographically smallest} \\ \text{satisfying assignment of } \varphi, & \text{if } \varphi \in \mathrm{SAT}, \\ \perp, & \text{if } \varphi \notin \mathrm{SAT}, \end{cases}$$

where $\perp$ is some special symbol to denote that a function is undefined at a point. (The name f_{left} comes from the fact that the lexicographically smallest satisfying assignment of a formula is the leftmost one in the standard binary tree to represent all possible assignments.)

Krentel [Kr88] showed that every function in $\mathrm{FP}^{\mathrm{NP}}$ can be reduced to f_{left}, and thus f_{left} is complete for $\mathrm{FP}^{\mathrm{NP}}$. His proof involved showing that the leftmost accepting path of an NP computation can be made to correspond to the correct query path of a $\mathrm{FP}^{\mathrm{NP}}$ computation and to the leftmost satisfying assignment in the output of Cook's reduction to SAT [Co71].

While we have $\mathrm{FP}^{\mathrm{NP}}$ as an upper bound on the complexity of computing NP witnesses, we note that there are satisfying assignments that are *more difficult* to compute than the leftmost one: Toda [To90] showed that computing the lexicographically middle satisfying assignments is $\mathrm{FP}^{\mathrm{PP}}$-complete, where PP denotes probabilistic polynomial time. In this paper, we ask whether there are satisfying assignments that are *easier* to compute than in $\mathrm{FP}^{\mathrm{NP}}$. An interesting candidate class is $\mathrm{FP}_{\|}^{\mathrm{NP}}$, the class of functions in $\mathrm{FP}^{\mathrm{NP}}$ that can be computed by making nonadaptive queries to NP, that is, all the queries must be written down before any answers are received from the oracle. More specifically, our work is motivated by the following questions.

(1) What is the structure of $\mathrm{FP}_{\|}^{\mathrm{NP}}$? What functions are in $\mathrm{FP}_{\|}^{\mathrm{NP}}$? What functions are hard (or complete) for $\mathrm{FP}_{\|}^{\mathrm{NP}}$?

(2) Can proofs of membership for NP-complete sets (for example satisfying assignments) be computed by functions in $\mathrm{FP}_{||}^{\mathrm{NP}}$?

(3) Are there classes of proofs that are easier to generate than the leftmost proof? What is the relationship between different proofs? Do all proofs of membership contain the same amount of information?

With respect to (1), Chen and Toda [CT93] showed many functions to be complete for $\mathrm{FP}_{||}^{\mathrm{NP}}$ (see also [JT93]). An example is the function that for any Boolean formula φ gives the supremum of the satisfying assignments of φ. Observe that the supremum does not necessarily have to be a satisfying assignment of φ.

With respect to (2) and (3), note that f_{left} minimizes over an exponential range. Intuitively, we expect a function to be easier when we optimize over a smaller range. As an example, we might ask for an satisfying assignment having the maximum number of zeros. Then the range is polynomially (in fact, linearly) bounded. However, such an assignment need not to be unique. We define F_{zero} as a class of functions.[1]

$$f \in F_{zero} \iff f(\varphi) = \begin{cases} \text{some satisfying assignment of } \varphi \\ \text{having the maximum} \\ \text{number of zeros,} & \text{if } \varphi \in \mathrm{SAT}, \\ \bot, & \text{if } \varphi \notin \mathrm{SAT}. \end{cases}$$

Although we expect F_{zero} to at least contain satisfying assignments that are easier to compute than the left most (if there are any), we will show in Section 3 that F_{zero} is hard for $\mathrm{FP}_{||}^{\mathrm{NP}}$ under some appropriate functional type of reduction. Moreover, we will show that $\mathrm{FP}_{||}^{\mathrm{NP}}$ is *precisely* the class of functions that reduces to F_{zero}. Thus, although we don't know whether $\mathrm{FP}_{||}^{\mathrm{NP}} \cap F_{zero} \neq \emptyset$, we show that at least any function that reduces to F_{zero} is already in $\mathrm{FP}_{||}^{\mathrm{NP}}$. We will call this property *strong hardness*. Hence. F_{zero} is strongly hard for $\mathrm{FP}_{||}^{\mathrm{NP}}$. An interesting consequence is that f_{left} cannot be reduced to F_{zero}, unless $\mathrm{FP}^{\mathrm{NP}} = \mathrm{FP}_{||}^{\mathrm{NP}}$. This supports the intuition that F_{zero} is computationally easier than f_{left}.

In Section 4, we link the question of whether these functions are complete in the classical sense – *some member* has to be computable in $\mathrm{FP}_{||}^{\mathrm{NP}}$ – to the complexity of generating proofs of membership for certain NP-complete

[1] An alternative definition is in terms of *multivalued functions* as for example in [Se94].

sets. For example consider the following function class.

$$f \in F_{sat} \iff f(\varphi) = \begin{cases} \text{some satisfying assignment of } \varphi, & \text{if } \varphi \in \text{SAT}, \\ \perp, & \text{if } \varphi \notin \text{SAT}. \end{cases}$$

We show that some function in F_{zero} is computable in $\text{FP}_{\|}^{\text{NP}}$ if and only if some function in F_{sat} is computable in $\text{FP}_{\|}^{\text{NP}}$. In other words, if it is at all possible to compute some satisfying assignment within $\text{FP}_{\|}^{\text{NP}}$, then this can be done for F_{zero}. Thus, loosely speaking, the "easy" assignments survive when going from F_{sat} to F_{zero}.

The above results extend to a large class of optimization problems, the *polynomially bounded* NP *optimization problems*, NPbOpt for short, where we have a polynomially bounded cost function associated with the solutions of an NP set which we want to optimize. An example is F_{zero}, where we count the number of zeros. We will show that the NPbOpt's based on one of the *known* NP-complete sets are all *equivalent* under an appropriate functional reducibility. Therefore, the above results for F_{zero} carries over to any such NPbOpt. Examples are maximum clique, longest paths, or various versions of the travelling salesman problem. See Section 2.2 for more examples.

We note that Chen and Toda [CT91] introduced a similar concept, the NP combinatorial optimization problems, NPCOP for short. The difference to an NPbOpt is that the domain of the optimization problem has to be polynomial-time decidable. This difference doesn't matter when considering for example the maximum clique problem, because here, the domain is simply the set of all graphs. However, F_{zero} is not an NPCOP unless P = NP, because here, the domain is SAT. Thus, the class of optimization problems introduced in this paper, NPbOpt, is the more general framework.

NPSV is the class of functions that can be computed by single-valued nondeterministic polynomial-time transducers. NPSV is a subclass of $\text{FP}_{\|}^{\text{NP}}$. Is it possible to compute satisfying assignments even within NPSV? Hemaspaandra et.al. [HNOS94] give a negative answer: this is *not* possible within NPSV, unless the polynomial-time hierarchy collapses to the second level, i.e., $F_{sat} \cap \text{NPSV} \neq \emptyset \implies \text{PH} = \Sigma_2^P$.

An improvement of their result to $\text{FP}_{\|}^{\text{NP}}$ will therefore give a negative answer to the completeness (in the classical sense) of functions computing solutions to F_{zero}. In Section 5, we will make a first step by extending their result to $\text{FP}^{\text{NPSV[1]}}$. Very recently, this has been improved further by Ogihara [Og95] to $\text{FP}^{\text{NPSV}[c \log(n)]}$, for $c < 1$.

Furthermore, we show that F_{sat} is strongly hard for NPSV, $\mathrm{FP}^{\mathrm{NPSV[1]}}$, and $\mathrm{FP}_{||}^{\mathrm{NPSV}}$ under appropriate types of reducibilities, respectively. As a consequence, we improve upon a result of Watanabe and Toda [WT93]. They show that f_{left} cannot be reduced to F_{sat}, unless NP $\neq$ co-NP. We show that even $\mathrm{FP}^{\mathrm{NPSV[1]}}$ cannot be reduced to F_{sat}, unless NP $\neq$ co-NP. (Note that f_{left} is hard for $\mathrm{FP}^{\mathrm{NPSV[1]}}$.)

The following is a summary of our main results:

(1) We extend the work of [CT91] and show that a broad class of functions, that are interreducible, is hard for $\mathrm{FP}_{||}^{\mathrm{NP}}$ (Theorem 3.1 and 3.2).

(2) We show that these functions are unlikely to complete for $\mathrm{FP}_{||}^{\mathrm{NP}}$ by providing a link to the problem of generating proofs of membership for NP-complete sets like SAT (Theorem 4.2).

(3) We will characterize $\mathrm{FP}_{||}^{\mathrm{NP}}$ as the class of functions that is (metric) reducible to F_{zero} (Theorem 4.5). Furthermore, NPSV, $\mathrm{FP}_{||}^{\mathrm{NPSV}}$, and $\mathrm{FP}_{||}^{\mathrm{NP}}$ are precisely the class of functions that are reducible to F_{sat} for appropriate types of reducibilities, respectively (Theorem 5.5). We will call this property strong hardness.

(4) As a consequence of the strong hardness results, we strengthen a result of Watanabe and Toda [WT93] (Corollary 5.7).

2 Preliminaries

2.1 NP-Relations and Optimization Problems

Let $\Sigma = \{0, 1\}$ be an alphabet and let R be a relation on $\Sigma^* \times \Sigma^*$. The *domain of* R is the set $D_R = \{\, x \in \Sigma^* \mid \exists y \in \Sigma^*\ xRy \,\}$. Any $y \in \Sigma^*$ witnessing that some x is in D_R is called a *solution for* x *with respect to* R. The set of all solutions for some $x \in \Sigma^*$ with respect to R is denoted by $S_R(x)$, and S_R is the set of all solutions.

We say that R is a NP-*relation*, if the following two conditions hold.

(i) There is a polynomial p such that for all $x \in D_R$, any solution for x has length $p(|x|)$, i.e., $S_R(x) \subseteq \Sigma^{p(|x|)}$, and

(ii) R is decidable in polynomial time.

By F_R, we denote the class of all functions $f : \Sigma^* \to S_R \cup \{\perp\}$ such that for all $x \in \Sigma^*$

$$f(x) \quad = \quad \begin{cases} \text{some } y \in S_R, & \text{if } x \in D_R, \\ \perp, & \text{otherwise.} \end{cases}$$

Note that there is not a unique terminology for such function classes. The proof generating functions such as F_R are called *inverse functions* in [WT93]. Most of the classes we are considering were defined in [BLS84] or [Se94]. Selman [Se94] denotes these classes in terms of *multivalued functions*. An inclusion of two function classes here becomes a *refinement* in terms of multivalued functions.

In this paper, we investigate how hard it is to compute *some* solution for a given x with respect to some NP-relation, i.e., some function in F_R. For comparing the complexity of this task for different NP-relations, we need to consider reductions between them. Here, the following question arises. Given two NP-relations R_0 and R_1 such that we can many-one reduce D_{R_0} to D_{R_1} via some function $h \in \mathrm{FP}$, where FP denotes the set of polynomial-time computable functions. For any $x \in D_{R_0}$, can we compute a solution for x with respect to R_0 from a given solution for $h(x)$ with respect to R_1? In general, this is not known. But in some cases, there are such *witness-preserving* reductions. Consider for example Cook's reduction from an arbitrary NP set to SAT [Co71]. For a fixed NP machine M, for any input x for M, Cook constructed a Boolean formula φ_x such that x is accepted by M iff $\varphi_x \in \mathrm{SAT}$. And in fact, from any satisfying assignment for φ_x, one can compute in polynomial time an accepting path of M on input x.

But in general, a many-one reduction doesn't respect the structure of the solution spaces of the instances that are mapped to each other. It just guarantees the existence/nonexistence of solutions. However, looking at the many-one reductions of the NP-completeness proofs in the standard textbooks (see for example in [BDG88, HU79]), we know that all the known NP-complete sets in fact share the above described property with SAT.

Let R be an NP-relation such that D_R is NP-complete. We call R *witness-preserving complete*, if, for any NP-relation R', there exist functions g, $h \in \mathrm{FP}$ such that h is a many-one reduction from $D_{R'}$ to D_R, and for any $x \in D_{R'}$ and any $y \in S_R(h(x))$, we have that $g(x, y) \in S_{R'}(x)$. That is, g computes a witness for x given a witness for $h(x)$. A set L is *witness-preserving complete*, if it has a witness-preserving complete NP-relation.

Note that all sets that are isomorphic to SAT are witness-preserving NP-complete: let L be isomorphic to SAT via an isomorphism h, say, from L

to SAT. Define R_L by $x R_L y$ if y is a satisfying assignment for $h(x)$. Then R_L is a witness-preserving complete NP-relation for L because for any NP set A, if f is the Cook reduction from A to SAT, then $h^{-1} \circ f$ reduces A to L in a witness-preserving way since the Cook reduction does so. Thus, all the *known* NP-complete sets are in fact witness-preserving complete.

In a very general approach, Agrawal and Biswas [AB92] introduced the notion of a *universal relation*, that captures the idea of witness-preserving in a very strict way. They showed that any universal NP-relation is witness-preserving complete. Since all sets that are isomorphic to SAT have a universal relation, the comment in the previous paragraph also follows from their result.

We now turn to optimization problems associated with an NP-relation R. A function $c \in \mathrm{FP}$, $c : D_R \times S_R \to \mathbf{N}$, is called a *solution cost function for* R. The *optimal solution cost function for* R, $c^* : D_R \to \mathbf{N}$ is defined by

$$c^*(x) \;=\; \max\{\, c(x,y) \mid y \in S_R(x) \,\}.$$

For any $x \in D_R$, we define the set of optimal solutions by $\mathrm{OptSol}_{R,c}(x) = \{\, y \in S_R(x) \mid c(x,y) = c^*(x) \,\}$. For any NP-relation R and any cost function c for R, we say that (R,c) is an NP *optimization problem*, namely the problem to compute an optimal solution for any given $x \in D_R$. For any function $f : \Sigma^* \to S_R \cup \{\perp\}$, we define

$$f \in \mathrm{Opt}_{R,c} \iff f(x) \;=\; \begin{cases} \text{some } y \in \mathrm{OptSol}_{R,c}(x), & \text{if } x \in D_R, \\ \perp, & \text{otherwise.} \end{cases}$$

(R,c) is called a *polynomially bounded* NP *optimization problem*, NPbOpt for short, if there is a polynomial that bounds the solution cost function c. Note that, from any NP set, we can derive a polynomially bounded NP optimization problem by taking some relation witnessing the set being in NP and some arbitrary polynomially bounded cost function c. In contrast, Chen and Toda [CT91] defined the more restricted notion of an NP combinatorial optimization problem (NPCOP) which is defined similar to an NPbOpt, but with the additional constraint that the domain D_R is a set in P.

Next, we define two operations for NP-relations: join and embedding. The join corresponds to the *linear paddability* operation for NPCOP's from Chen and Toda [CT91]. An NP-relation R has a *join function*, if there are two functions $join_R$, $g \in \mathrm{FP}$ such that for any $x_1, \ldots, x_n \in \Sigma^*$, if

$join_R(x_1, \ldots, x_n) = z$ and $y \in S_R(z)$, then $g(x_1, \ldots, x_n, y) = (y_1, \ldots, y_n)$ where $y_i \in S_R(x_i)$, for $i = 1, \ldots, n$.

That is, the join function combines several given strings into one string z in such a way that from any solution for z, we can compute solutions for the given strings. An even stronger version of the join function is required in the definition of a universal relation by Agrawal and Biswas [AB92]. Since they have shown that all the known NP sets have a universal relation, in particular, they have a join function.

For example, the join function for SAT is essentially the conjunction. That is, for any two Boolean formulas φ_1 and φ_2, after renaming the variables so that φ_1 and φ_2 have disjoint sets of variables, $join_{sat}(\varphi_1, \varphi_2) = \varphi_1 \wedge \varphi_2$.

For an NPbOpt (R, c) we say that the join function of R *respects* c, if, when we join two instances, then we can compute an optimal solution for the two instances from an optimal solution for the join of the instances. More formally, there has to exist a function $g \in \mathrm{FP}$, $g : D_R \times D_R \times S_R \to S_R \times S_R$ such that for all x_0, $x_1 \in D_R$ and for all $y \in \mathrm{OptSol}_{R,c}(join_R(x_0, x_1))$, if $g(x_0, x_1, y) = (y_0, y_1)$, then $y_0 \in \mathrm{OptSol}_{R,c}(x_0)$ and $y_1 \in \mathrm{OptSol}_{R,c}(x_1)$.

We say that (R, c) has an *embedding function*, if there exist two functions $e, g \in \mathrm{FP}$, $e : \Sigma^* \to D_R$ and $g : \Sigma^* \times S_R \to S_R$, such that there is a $f \in \mathrm{Opt}_{R,c}$ such that for all $x \in \Sigma^*$

$$\forall y \in \mathrm{OptSol}_{R,c}(e(x)) : \quad g(x, y) = f(x).$$

That is, e maps a given string x to some string z in the domain D_R of R such that from an optimal solution for z, one can either compute an optimal solution for x, if $x \in D_R$, or detect that $x \notin D_R$.

We adapt the notion of Agrawal and Biswas [AB92] to our framework and call an NPbOpt (R, c) *universal*, if

1. R is a witness-preserving complete NP-relation,
2. R has a join function that respects c, and
3. (R, c) has an embedding function.

2.2 Examples

First of all, note that all examples of Chen and Toda for NPCOP's can easily be modified to be universal NPbOpt's. They mention for example

- Maximum Two Satisfiability, where each clause of a CNF formula contains at most two literals,
- Maximum Clique,
- Minimum Coloring,
- Longest Path,
- 0-1 Integer Programming, and
- 0-1 Travelling Salesman, where the edges have weights zero or one.

These problems can be formulated as universal NPbOpt's because there is an associated NP-complete decision problem for each of them. This, however, we expect for any interesting optimization problem, and therefore, we extend the concept of Chen and Toda. We give some examples that are not expressible as an NPCOP unless P = NP, because the domain of these problems is NP-complete.

- F_{zero}

Let R_{sat} be the NP-relation for SAT that checks satisfying assignments, and $\#_{zero}$ be the cost function that counts the number of zeros in a satisfying assignment. Then $(R_{sat}, \#_{zero})$ is an NPbOpt that requires to compute some satisfying assignment with the maximum number of zeros, i.e., a function from F_{zero}.

The join function is the conjunction which respects $\#_{zero}$. Furthermore, $(R_{sat}, \#_{zero})$ has an embedding function. Let $\varphi = \varphi(x_1, \ldots, x_n)$ be a Boolean formula and let $z_1, \ldots, z_{n+1}$ be new variables. Define

$$\Phi(x_1, \ldots, x_n, z_1, \ldots, z_{n+1}) \;=\; \varphi \vee (z_1 \wedge \cdots \wedge z_{n+1}).$$

Then $\Phi \in$ SAT, and if $\varphi \in$ SAT and a is a satisfying assignment with the maximum number of zeros for φ, then $a0^{n+1}$ is a satisfying assignment for Φ with the maximum number of zeros. On the other hand, if $\varphi \notin$ SAT, then $0^n 1^{n+1}$ is a satisfying assignment for Φ with the maximum number of zeros. Therefore, getting a satisfying assignment with the maximum number of zeros for Φ, one can either get one for φ or detect that $\varphi \notin$ SAT.

We conclude that $(R_{sat}, \#_{zero})$ is a universal NPbOpt.

- $F_{max\text{-}zero\text{-}guess}$

$F_{max\text{-}zero\text{-}guess}$ is defined on instances $(N, x, 1^m)$ for the standard universal NP-complete set, i.e., it is asked whether the nondeterministic Turing machine N accepts input x in at most m steps. Any nondeterministic computation path of N can be represented as a binary string corresponding to the nondeterministic branch points in the computation. $F_{max\text{-}zero\text{-}guess}$ is the class of functions that, on input $(N, x, 1^m)$, give some accepting path of N on x with the maximum number of zeros, and are undefined, if there is no accepting path. Join and embedding functions are similar as for F_{zero}. Here, one has to manipulate the input machine N appropriately.

- Unary-TSP

In unary-TSP, there is given an undirected graph G with integer weights given in unary notation on the edges, so that the weights are bounded by the size of the input. The task is to determine a traveling salesman tour in G having minimal weight.[2] For the join function see [AB92]. For the embedding function, let B be the sum of the weights of the edges of G. Let G' be the extension of G to a complete graph, where all new edges have weight $B + 1$. Now, G' clearly has a traveling salesman tour. Furthermore, if the tour with minimum weight in G' is bounded by B, then this is also a minimum tour in G. Otherwise, there is no traveling salesman tour in G.

In Section 3 and 4, we show several properties of universal NPbOpt's. Hence, in particular, this applies to all the above mentioned optimization problems.

2.3 Functional Reducibilities

There are several notions of reducibility between functions. Krentel [Kr88] introduced the *metric reduction*. Let f, g be functions.

$$f \leq^{FP}_{1\text{-}T} g \iff \exists t_1, t_2 \in \mathrm{FP} : \quad f(x) = t_2(x, g \circ t_1(x)).$$

This clearly captures the idea of being able to compute $f(x)$ from one call to g.

Watanabe and Toda [WT93] and Chen and Toda [CT91] extended this reduction to function classes. Let G be a class of functions. We distinguish the case that *one pair* of translation functions reduces f to all functions in

[2] Note that unary-TSP is a minimization problem. By defining an appropriate solution cost function, this can be easily turned into a maximization problem.

G, or that for each function $g \in G$ there is a pair of translation functions that reduces f to g. In the first case, we call the reduction *uniform*.

$$f \leq_{1\text{-}T}^{uniform\text{-}FP} G \iff \exists t_1, t_2 \in \mathrm{FP}\ \forall g \in G: \quad f(x) = t_2(x, g \circ t_1(x)),$$
$$f \leq_{1\text{-}T}^{FP} G \iff \forall g \in G\ \exists t_1, t_2 \in \mathrm{FP}: \quad f(x) = t_2(x, g \circ t_1(x)).$$

We will also consider the more general type of reduction when more than one instance is given to a function in G. That is, $t_1(x)$ produces a list of instances and t_2 gets the function values of some function in G of these instances. This is called a *truth-table reduction* and denoted by $\leq_{tt}^{uniform\text{-}FP}$ and $\leq_{FP}^{tt}$, respectively.

If G is a class of partial functions, we must deal with the case that $f(x)$ is defined, while $g \circ t_1(x)$ is undefined. We call a reduction *strict*, denoted by $f \leq_{tt}^{FP\text{-}strict} G$, if there are functions t_1, $t_2 \in \mathrm{FP}$ witnessing that $f \leq_{FP}^{tt} G$ such that for all x, if $f(x)$ is defined, then g is defined for all instances produced by $t_1(x)$, for all $g \in G$.

Let $\leq_r$ be any of the reducibilities defined here. For a class F of functions, we say that G is hard for F with respect to $\leq_r$-reduction, if for all functions $f \in F$, we have $f \leq_r G$. This is denoted by $F \leq_r G$. Furthermore, we say that G is *complete for* F, if in addition there is some function in G that is also in F, i.e., $F \cap G \neq \emptyset$.

We also consider the case that eventually not all functions in F are reducible to G, but that any function in G can be used to compute *some* function in F. We call this a *weak* reduction.

$$F \leq_{1\text{-}T}^{weak\text{-}FP} G \iff \forall g \in G\ \ \exists t_1, t_2 \in \mathrm{FP}: \quad t_2(x, g \circ t_1(x)) \in F.$$

The uniform and truth-table versions of this reduction are defined analogously. The *uniform weak Turing reduction* was defined in [FHOS93].

It is easy to see that all the reducibilities defined here are transitive, but in general, only the weak reducibilities are reflexive.

Although the uniformness condition seems to be a strong restriction on the reduction, Watanabe and Toda [WT93], using a proof technique from Grollmann and Selman [GS88], have shown that for many function classes these two reduction types are in fact equivalent.

Lemma 2.1 [WT93] *Let f be a function, R an* NP*-relation and c a solution cost function for R. Then we have*

(i) $f \leq_{FP}^{tt} F_R \iff f \leq_{tt}^{uniform\text{-}FP} F_R$,

(ii) $f \leq_{FP}^{tt} \mathrm{Opt}_{R,c} \iff f \leq_{tt}^{uniform\text{-}FP} \mathrm{Opt}_{R,c}$.

The same holds for the other reducibilities defined above, i.e., $\leq_{1\text{-}T}^{FP}$, $\leq_{tt}^{FP\text{-}strict}$, $\leq_{1\text{-}T}^{FP\text{-}strict}$, $\leq_{1\text{-}T}^{weak\text{-}FP}$, *and* $\leq_{tt}^{weak\text{-}FP}$.

When we consider NP optimization problems that have embedding functions as in the previous section, then reductions to it can always be made strict.

Lemma 2.2 *Let f be a function and (R,c) a universal* NPbOpt. *Then we have* $f \leq_{FP}^{tt} \mathrm{Opt}_{R,c} \iff f \leq_{tt}^{FP\text{-}strict} \mathrm{Opt}_{R,c}$.

The lemma also holds for the other reducibilities defined above.

3 Function Classes Hard for $\mathrm{FP}_{||}^{\mathrm{NP}}$

Chen and Toda [CT91] showed that linearly paddable NPCOP's are hard for $\mathrm{FP}_{||}^{\mathrm{NP}}$ under $\leq_{1\text{-}T}^{FP}$-reductions. Our first theorem states that this holds as well for universal NPbOpt's, as for example F_{zero}. The proof is similar to that of Chen and Toda, however, we need the embedding function to get around the difficulty that the domains of our optimization problems are in NP.

Theorem 3.1 *Let (R,c) be a universal* NPbOpt. *Then* $\mathrm{FP}_{||}^{\mathrm{NP}} \leq_{1\text{-}T}^{FP} \mathrm{Opt}_{R,c}$.

Proof. Let $f \in \mathrm{FP}_{||}^{\mathrm{NP}}$ via some polynomial-time transducer T and some NP set A. Let $x \in \Sigma^*$ be fixed. We show how to compute $f(x)$ when getting an arbitrary optimal solution for some instance z with respect to (R,c).

Let $w_1, \ldots, w_k$ be the queries of transducer T on input x to A. Since D_R is NP-complete, there is a function $h \in \mathrm{FP}$ reducing A to D_R. Let e and g be embedding functions for (R,c). We use e to map all strings $h(w_i)$ to D_R and then combine all the resulting strings into one string z using the join function, $join_R$, of R. That is, we define

$$z = join_R(e \circ h(w_1), \ldots, e \circ h(w_k)).$$

Let $y \in \mathrm{OptSol}_{R,c}(z)$. Since $join_R$ respects c, from y we can compute solutions $y_i \in \mathrm{OptSol}_{R,c}(e \circ h(w_i))$, for $i = 1, \ldots, k$. Now, $g(h(w_i), y_i)$ either gives a witness that $h(w_i)$ is in D_R, and hence w_i is in A, or $g(h(w_i), y_i)$ is undefined, and hence w_i is not in A.

Thus, we can compute the answers to $w_1, \ldots, w_k$ from y, and therefore, we can compute $f(x)$. □

Our next theorem shows that any universal NPbOpt is hard for any other NPbOpt under $\leq_{1\text{-}T}^{weak\text{-}FP}$-reductions, and hence, any two such NPbOpt's are equivalent to each other. In the previous theorem it was not necessairy for the relation R to be witness-preserving complete, but now we seem to need this property.

Theorem 3.2 *Let (R_0, c_0) be a universal* NPbOpt. *Then, for any* NPbOpt *(R, c), we have* $\mathrm{Opt}_{R,c} \leq_{1\text{-}T}^{weak\text{-}FP} \mathrm{Opt}_{R_0,c_0}$.

Proof. We will show that for any $x \in \Sigma^*$, we can map x to some string $z \in D_{R_0}$ such that from an optimal solution for z with respect to (R_0, c_0), we can either compute an optimal solution for x with respect to (R, c) or detect that x is not in D_R.

Let us define the NP-relation R' as follows. For any $x \in \Sigma^*$ and $k \leq p(|x|)$, where p is some polynomial that bounds the solution cost function c,

$$(x,k)R'y \iff xRy \text{ and } c(x,y) \geq k.$$

Let $x \in \Sigma^*$ be fixed and let k^* be the maximum k such that $(x,k) \in D_{R'}$. Observe that any solution for (x,k^*) with respect to R' is an optimal solution for x with respect to (R,c), i.e., $S_{R'}(x,k^*) \subseteq \mathrm{OptSol}_{R,c}(x)$. We will show how to compute a witness for each $(x,k) \in D_{R'}$ when getting an arbitrary optimal solution for some instance z with respect to (R_0, c_0). From these witnesses, we output the one for (x,k^*).

Since R' is an NP-relation and since R_0 is witness-preserving complete, there is a function $h \in \mathrm{FP}$ that reduces $D_{R'}$ to D_{R_0} in such a way that for any $(x,k) \in D_{R'}$ and from any witness for $h(x,k) \in D_{R_0}$ we can compute a witness for $(x,k) \in D_{R'}$.

As in the proof of Theorem 3.1, using the embedding function and the join function for R_0, we combine all the resulting strings $h(x,1), \ldots, h(x,p(|x|))$ into one string z such that from a witness for $z \in D_{R_0}$, we can compute witnesses for all $h(x,k)$ that are in D_{R_0}.

□

Corollary 3.3 *Let (R,c) and (R',c') be universal* NPbOpt*'s. Then we have* $\mathrm{Opt}_{R',c'} \equiv_{1\text{-}T}^{weak\text{-}FP} \mathrm{Opt}_{R,c}$

Thus all the examples of optimization problems we give in Section 2.2 are equivalent with respect to $\leq_{1\text{-}T}^{weak\text{-}FP}$ reductions. So although F_{zero} might look as a somewhat technical problem, it is in fact equivalent to any of the more natural NPbOpt's.

We remark that if we don't assume the existence of an embedding function for the NPbOpt's, then the above theorems still hold, but with the corresponding truth-table reductions, respectively.

4 Completeness

In the previous section, we established a framework for proving certain functions hard for $\mathrm{FP}_{||}^{\mathrm{NP}}$. The natural question that arises is whether these functions are also complete for $\mathrm{FP}_{||}^{\mathrm{NP}}$. (Recall that G is complete for F if G is hard for F and, in addition, $F \cap G \neq \emptyset$). Chen and Toda [CT91] showed that a *randomized version* of $\mathrm{FP}_{||}^{\mathrm{NP}}$ can actually compute any NPCOP in the following sense: for any NPCOP there is a two-place function $f \in \mathrm{FP}_{||}^{\mathrm{NP}}$, that, when given as one input the problem instance x and as the other input some randomly chosen string, outputs with high probability an optimal solution for x with respect to the given NPCOP. This result holds also for NPbOpt's.

Theorem 4.1 [CT91] *Let (R, c) be an* NPbOpt *and let e be a polynomial. Then there exist a function $f \in \mathrm{FP}_{||}^{\mathrm{NP}}$ and a polynomial r such that for all $x \in D_R$, $|x| = n$,*

$$Prob\{\, w \in \{0,1\}^{r(n)} \mid f(x,w) \in \mathrm{OptSol}_{R,c}(x)\,\} \quad \geq \quad 1 - 2^{-e(n)}.$$

However, at present time, we do not know whether the NPbOpt results from the previous section can be extended to completeness results. Our next theorem states that if it is at all possible to compute some satisfying assignment with parallel queries to NP, then this is also possible within F_{zero}. In other words, obtaining such a completeness result is exactly as hard as any proof that one can indeed compute some satisfying assignment in $\mathrm{FP}_{||}^{\mathrm{NP}}$.

Theorem 4.2 *Let (R, c) be an* NPbOpt *and R_0 be a witness-preserving complete* NP*-relation. Then $F_{R_0} \cap \mathrm{FP}_{||}^{\mathrm{NP}} \neq \emptyset \iff \mathrm{Opt}_{R,c} \cap \mathrm{FP}_{||}^{\mathrm{NP}} \neq \emptyset$.*

Proof. Let $f \in F_{R_0} \cap \mathrm{FP}_{||}^{\mathrm{NP}}$. We define an NP-relation R' as follows. For any $x \in \Sigma^*$ and $k \leq p(|x|)$, where p is some polynomial that bounds the solution cost function c,

$$(x,k)R'y \iff xRy \text{ and } c(x,y) \geq k.$$

Since R' is an NP-relation and since R_0 is witness-preserving complete, there are functions h, $g \in \mathrm{FP}$ such that h many-one reduces $D_{R'}$ to D_{R_0} and for any $(x,k) \in D_{R'}$ and any string z witnessing that $h(x,k) \in D_{R_0}$, $g(x,k,z)$ is a witness that (x,k) is in $D_{R'}$.

Let $x \in \Sigma^*$ be fixed. We show that we can compute some value in $\mathrm{OptSol}_{R,c}(x)$ with parallel queries to some NP set.

Let k^* be the maximal k such that $(x,k) \in D_{R'}$, i.e., we have $c^*(x) = k^*$. Then $h(x,k^*)$ is in D_{R_0} and, by our assumption, $z = f \circ h(x,k^*)$ is some witness for this. Hence, $g(x,k^*,z)$ is a witness that $(x,k^*) \in D_{R'}$ and therefore, we have that $g(x,k^*,z) \in \mathrm{OptSol}_{R,c}(x)$. That is, we define

$$f'(x) = g(x, c^*(x), f \circ h(x, c^*(x))).$$

It remains to show that $f' \in \mathrm{FP}_{||}^{\mathrm{NP}}$. We leave this to the reader. □

Corollary 4.3 $F_{sat} \cap \mathrm{FP}_{||}^{\mathrm{NP}} \neq \emptyset \iff F_{zero} \cap \mathrm{FP}_{||}^{\mathrm{NP}} \neq \emptyset$.

On the other hand, we will show in the next theorem that all functions that are $\leq_{1\text{-}T}^{FP}$-reducible to some NPbOpt are already in $\mathrm{FP}_{||}^{\mathrm{NP}}$, and therefore, together with Theorem 3.1, it follows that $\mathrm{FP}_{||}^{\mathrm{NP}}$ can be *characterized* as the class of functions that are $\leq_{1\text{-}T}^{FP}$-reducible to some NPbOpt. This can be interpreted as a weaker form of completeness.

Definition 4.4 *Let F and G be function classes. We say that G is* strongly hard *for F under $\leq_r$-reduction, if $F = \{ f \mid f \leq_r G \}$.*

The next theorem and corollary show that the hardness results obtained for NPbOpt's can indeed be strengthened to strong hardness.

Theorem 4.5 *Let f be a function such that $f \leq_{1\text{-}T}^{FP} \mathrm{Opt}_{R,c}$, for some* NPbOpt *$(R,c)$. Then f is in $\mathrm{FP}_{||}^{\mathrm{NP}}$.*

Proof. By Lemma 2.1, we can assume that the reduction is uniform. Let f be reducible to $\mathrm{Opt}_{R,c}$ via $t_1, t_2 \in \mathrm{FP}$, i.e., we have for any x and for all $y \in \mathrm{OptSol}_{R,c}(t_1(x))$ that $f(x) = t_2(x,y)$.

Define NP sets A and B as follows. For any $x \in \Sigma^*$, $k \leq p(|x|)$, and $i \leq q(|x|)$, where p is some polynomial that bounds the solution cost function c and q is some polynomial that bounds the length of the solutions for x with respect to R

$$\begin{aligned} (x,k) \in A &\iff \exists y \in S_R(t_1(x)) : c(t_1(x),y) \geq k, \\ (x,k,i) \in B &\iff \exists y \in S_R(t_1(x)) : c(t_1(x),y) \geq k \text{ and} \\ &\quad \text{the } i\text{-th bit of } t_2(x,y) \text{ is a one.} \end{aligned}$$

Let k^* be the maximal k such that $(x,k) \in A$. Then the i-th bit of $f(x)$ is one, if $(x,k^*,i) \in B$, and zero, otherwise, for $i = 1, \ldots, q(|x|)$. Therefore, we can compute $f(x)$ by asking in parallel the queries (x,k) to A and (x,k,i) to B, for $k = 1, \ldots, p(|x|)$ and $i = 1, \ldots, q(|x|)$. Thus $f \in \mathrm{FP}_{||}^{A \oplus B} \subseteq \mathrm{FP}_{||}^{\mathrm{NP}}$. □

In fact, in Theorem 4.5, it suffices to assume that $f \leq_{FP}^{tt} \mathrm{Opt}_{R,c}$.

Taking Theorem 4.5 and Theorem 3.1 together, we obtain the already mentioned characterization of $\mathrm{FP}_{||}^{\mathrm{NP}}$ as the class of functions that is reducible to any universal NPbOpt.

Corollary 4.6 *Let (R,c) be a universal* NPbOpt. *Then*

$$\mathrm{FP}_{||}^{\mathrm{NP}} = \{ f \mid f \leq_{1\text{-}T}^{FP} \mathrm{Opt}_{R,c} \} = \{ f \mid f \leq_{FP}^{tt} \mathrm{Opt}_{R,c} \}.$$

Corollary 4.7 $\mathrm{FP}_{||}^{\mathrm{NP}} = \{ f \mid f \leq_{FP}^{tt} F_{zero} \}$.

It follows that if any $\mathrm{FP}^{\mathrm{NP}}$-complete function is reducible to, say F_{zero}, then this function can already be computed with parallel queries to NP, and hence $\mathrm{FP}^{\mathrm{NP}}$ would be the same as $\mathrm{FP}_{||}^{\mathrm{NP}}$.

Corollary 4.8 *Let (R,c) be a universal* NPbOpt. *Then*

$$f_{left} \leq_{1\text{-}T}^{FP} \mathrm{Opt}_{R,c} \iff \mathrm{FP}_{||}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{NP}} \iff \mathrm{P}^{\mathrm{NP}} = \mathrm{P}_{||}^{\mathrm{NP}}.$$

5 Negative Results and NPSV

For certain subclasses of $\mathrm{FP}_{\|}^{\mathrm{NP}}$, one can show that it is *not* possible to compute satisfying assignments, unless the polynomial-time hierarchy, PH, collapses. Hemaspaandra et al. [HNOS94] showed such a result for the class NPSV.

Definition 5.1 *A nondeterministic Turing transducer N is* single-valued, *if, for each input x, N generates the same output on all accepting computations.* NPSV *is the class of partial functions that can be computed by single-valued nondeterministic polynomial-time transducers.* $\mathrm{FP}_{\|}^{\mathrm{NPSV}[k]}$ *denotes the class of functions that is computable in polynomial time with k nonadaptive queries to an* NPSV *oracle.*

Note that $\mathrm{NPSV} \subseteq \mathrm{FP}_{\|}^{\mathrm{NP}}$, since with the help of an NP set one can get in parallel all the bits of an NPSV function value. In fact, $\mathrm{FP}_{\|}^{\mathrm{NP}} = \mathrm{FP}_{\|}^{\mathrm{NPSV}}$ [FHOS93].

Theorem 5.2 [HNOS94] *If* $\mathrm{NPSV} \cap F_{sat} \neq \emptyset$, *then* $\mathrm{PH} = \Sigma_2^P$.

The following lemma will enable us to extend this result to $\mathrm{FP}^{\mathrm{NPSV}[1]}$.

Lemma 5.3 *Let R be an* NP*-relation. Then*

$$F_R \cap \mathrm{FP}^{\mathrm{NPSV}[1]} \neq \emptyset \iff F_R \cap \mathrm{NPSV} \neq \emptyset.$$

Proof. If $F_R \cap \mathrm{FP}^{\mathrm{NPSV}[1]} = \emptyset$ then the lemma clearly holds. So assume that $f \in F_R \cap \mathrm{FP}^{\mathrm{NPSV}[1]}$. Let M be a FP machine and N be an NPSV machine witnessing that $f \in \mathrm{FP}^{\mathrm{NPSV}[1]}$. We have to show that $F_R \cap \mathrm{NPSV} \neq \emptyset$.

Consider the following machine N' on input x. First, N' simulates M on input x until M queries it's oracle. Let q_x be the query. Then N' assumes that the answer to the query is $\perp$ and continues the simulation of M. Let y be the output of M. If xRy holds, then N' outputs y and halts. (Note that this is a deterministic computation up to here.) Otherwise, N' simulates N on input q_x. If N rejects, then so does N'. If N accepts, let z be the value computed by N. Now, N' continues the simulation of M with z as the answer to q_x. Note that z is the answer that M actually gets when asking its oracle. Therefore, N' will generate the same output as M at the end of the computation.

Clearly, N' is an NPSV machine. Furthermore, if $x \notin D_R$, then N' generates no output. If $x \in D_R$, then N' outputs some $y \in S_R(x)$. □

Corollary 5.4 *If* $F_{sat} \cap \mathrm{FP}^{\mathrm{NPSV[1]}} \neq \emptyset$ *then* $\mathrm{PH} = \Sigma_2^P$.

This result has been improved recently by Ogihara [Og95] who showed that $F_{sat} \cap \mathrm{FP}^{\mathrm{NPSV}[c \log(n)]} = \emptyset$, for $c < 1$, unless the polynomial-time hierarchy collapses. It is an interesting open problem whether these results can be extended to even larger function classes.

The following theorem shows that for any witness-preserving complete NP-relation R, F_R is hard, and, in fact, even strongly hard for NPSV, $\mathrm{FP}^{\mathrm{NPSV[1]}}$ and $\mathrm{FP}_{||}^{\mathrm{NPSV}}$ with respect to different types of reductions.

Theorem 5.5 *Let* R *be a witness-preserving complete* NP*-relation.*

(i) $\mathrm{NPSV} = \{ f \mid f \leq_{1\text{-}T}^{FP\text{-}strict} F_R \} = \{ f \mid f \leq_{tt}^{FP\text{-}strict} F_R \}$,

(ii) $\mathrm{FP}^{\mathrm{NPSV[1]}} = \{ f \mid f \leq_{1\text{-}T}^{FP} F_R \}$,

(iii) $\mathrm{FP}_{||}^{\mathrm{NPSV}} = \{ f \mid f \leq_{FP}^{tt} F_R \}$.

Proof. (i) Let f be in NPSV and let N be an NPSV machine for f. Consider the following NP-relation R_N. For $x, y \in \Sigma^*$, where $|y| \leq p(|x|)$ and p is some polynomial that bounds the the running time of N

$$xR_Ny \iff \begin{array}{l} y \text{ is a computation path of } N \text{ on } x \\ \text{on which } N \text{ produces an output.} \end{array}$$

Since R is a witness-preserving complete NP-relation, there exist two functions t_1, $t_2 \in \mathrm{FP}$ such that t_1 maps any x from the domain of R_N to the domain of R and for any solution y for $t_1(x)$, i.e., $t_1(x)Ry$ holds, $t_2(x, y)$ gives a solution for x, i.e., $xR_Nt_2(x, y)$ holds. Clearly, from $t_2(x, y)$ one can compute $f(x)$ in polynomial time. Furthermore, the reduction (t_1, t_2) is strict.

For the other direction, let f be a function that is $\leq_{tt}^{FP\text{-}strict}$-reducible to F_R via the functions t_1, $t_2 \in \mathrm{FP}$. Consider the following NP machine N on input x. First, N computes the queries $t_1(x) = (w_1, \ldots, w_k)$ and then guesses solutions $y_1, \ldots, y_k$ for them with respect to R. If w_iRy_i for $i = 1, \ldots, k$, then N outputs $t_2(x, y_1, \ldots, y_k)$.

Since the reduction is strict, there will be a path where N finds solutions for all $w_1, \ldots, w_k$. Furthermore, for every k-tuple of solutions $y_1, \ldots, y_k$, $t_2(x, y_1, \ldots, y_k)$ will give the same value, namely $f(x)$. Hence, N is single-valued and computes f.

The inclusion from left to right of (ii) and (iii) follows by an easy modification of the argument for (i). In fact, we get the more general result that $\mathrm{FP}_{\|}^{\mathrm{NPSV}[k]} \subseteq \{ f \mid f \leq_{FP}^{k\text{-}tt} F_R \}$, for every $k \in \mathrm{FP}$.

For the reverse inclusion of (ii), let f be a function that is $\leq_{1\text{-}T}^{FP}$-reducible to F_R via the functions t_1, $t_2 \in \mathrm{FP}$. Consider the following NP machine N on input x. First, N computes $t_1(x)$ and then guesses a solution y for it with respect to R. If $t_1(x)Ry$, then N outputs $t_2(x, y)$.

Clearly, N is a NPSV machine that outputs $f(x)$ if it is defined. Now a FP machine with N as an oracle can compute $f(x)$ by producing the same output as N on x when it is defined, and $t_2(x, \perp)$, otherwise.

For the reverse inclusion of (iii), let f be a function that is $\leq_{FP}^{tt}$-reducible to F_R via the functions t_1, $t_2 \in \mathrm{FP}$. We show how to compute f with parallel queries to NP. Let $x \in \Sigma^*$ be fixed and let $w_1, \ldots, w_k$ be the queries produced by $t_1(x)$. By asking the w_i's to D_R, we can find out which ones of them actually have a solution with respect to R. Suppose l of $w_1, \ldots, w_k$ are in D_R, where $0 \leq l \leq k$. Observe that an NP machine knowing l can actually compute (on some path) the w_i's in D_R together with some solution for them, and therefore, via t_2 also $f(x)$. Since there are only k possibilities for l, i.e. polynomially many, we can define an NP set that, for each l, refers to the bits of $f(x)$, similar as in the proof of Theorem 4.5. All together, we can compute $f(x)$ by asking polynomially many queries in parallel to D_R and the latter NP set. □

Corollary 5.6 *F_{sat} is strongly hard for* NPSV *and* $\mathrm{FP}^{\mathrm{NPSV}[1]}$*, but not complete for these classes for $\leq_{1\text{-}T}^{FP\text{-}strict}$- and $\leq_{1\text{-}T}^{FP}$-reduction, respectively, unless the polynomial-time hierarchy collapses.*

Watanabe and Toda [WT93] asked whether one can compute the leftmost satisfying assignment of a formula from any other satisfying assignment. Recall that f_{left} is a $\mathrm{FP}^{\mathrm{NP}}$-complete function. They showed that this is very unlikely to be true: if $f_{left} \leq_{tt}^{FP\text{-}strict} F_{sat}$, then NP = co-NP. However, by the characterizations obtained in Theorem 5.5, we have that the assumption made is equivalent to NPSV = $\mathrm{FP}^{\mathrm{NP}}$. Thus we can now strengthen the result of Watanabe and Toda [WT93] by weakening the assumption to NPSV = $\mathrm{FP}^{\mathrm{NPSV}[1]}$, which still leads to the same consequence.

Corollary 5.7 *Let R be a witness-preserving complete* NP*-relation.*

(i) If $\mathrm{FP}^{\mathrm{NPSV}[1]} \leq_{tt}^{FP\text{-}strict} F_R$*, then* NP = co-NP.

(ii) For any $k > 1$, if $\mathrm{FP}_{||}^{\mathrm{NPSV}[k]} \leq_{1\text{-}T}^{FP} F_R$, then $\mathrm{P}_{||}^{\mathrm{NP}[l]} = \mathrm{P}^{\mathrm{NP}[1]}$ for any $l \geq 1$, and hence the polynomial-time hierarchy collapses.

(iii) Let (R', c) be a universal NPbOpt. *If $\mathrm{Opt}_{R',c} \leq_{1\text{-}T}^{FP} F_R$, then $\mathrm{P}_{||}^{\mathrm{NP}} = \mathrm{P}^{\mathrm{NP}[1]}$, and hence the polynomial-time hierarchy collapses.*

Proof. (i) From the assumption together with Theorem 5.5 (i), we conclude that $\mathrm{FP}^{\mathrm{NPSV}[1]} = \mathrm{NPSV}$. As a special case, when considering only characteristic functions, it follows that $\mathrm{P}^{\mathrm{NPSV}[1]} = \mathrm{NP}$. Now, observe that co-NP $\subseteq \mathrm{P}^{\mathrm{NPSV}[1]}$.

(ii) From the assumption together with Theorem 5.5 (ii), we conclude that $\mathrm{FP}_{||}^{\mathrm{NPSV}[k]} = \mathrm{FP}^{\mathrm{NPSV}[1]}$, and hence $\mathrm{P}^{\mathrm{NPSV}[k]} = \mathrm{P}^{\mathrm{NPSV}[1]}$. Now, the claim follows since $\mathrm{P}_{||}^{\mathrm{NPSV}[l]} = \mathrm{P}_{||}^{\mathrm{NP}[l]}$ for any $l \geq 0$ [FHOS93].

(iii) Follows from a similar argument as in (ii) together with Theorem 3.1 and the transitivity of the $\leq_{1\text{-}T}^{FP}$-reduction. □

Acknowledgements

We want to thank Manindra Agrawal and Somenath Biswas for helpful discussions. The referee comments helped a lot to improve the representation of the paper.

References

[AB92] M. Agrawal, S. Biswas. Universal Relations. In *Proc. 7th Structure in Complexity Theory Conference*, pages 207–220, 1992. To appear in *Information and Computation.*

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 1988.

[BLS84] R. Book, T. Long, A. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing* 13(3):461-487, 1984.

[Co71] S. Cook. The Complexity of Theorem-Proving Procedures. In *Proc. 3rd ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[CT91] Z. Chen and S. Toda. On the Complexity of Computing Optimal Solutions. In *International Journal of Foundations of Computer Science* 2:207-220, 1991.

[CT93] Z. Chen and S. Toda. An Exact Characterization of $\mathrm{FP}_{||}^{\mathrm{NP}}$. Manuscript, 1993.

[FHOS93] S. Fenner, S. Homer, M. Ogiwara, and A. Selman. On Using Oracles That Compute values. In Proc. 10th Annual Symposium on Theoretical Aspects of Computer Science (STACS), 398–407, 1993. To appear in *SIAM Journal on Computing*.

[GS88] J. Grollmann and A. Selman. Complexity Measures for Public-Key Cryptosystems. *SIAM Journal on Computing* 17:309–335, 1988.

[HNOS94] L. Hemaspaandra, A. Naik, M. Ogiwara, and A. Selman. Computing Solutions Uniquely Collapes the Polynomial Hierarchy. In Algorithms and Computaation, International Symposium ISAAC '94, Springer Verlag LNCS 834, pages 56–64, 1994. To appear in *SIAM Journal on Computing*.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[JT93] B. Jenner and J. Torán. Computing Functions With Parallel Queries to NP. *Theoretical Computer Science* 141, 175–193, 1995.

[Kr88] M. Krentel. The Complexity of Optimization Problems. *Journal of Computer and System Sciences* 36(3):761–767, 1988.

[Og95] M. Ogihara. Functions Computable with Limited Access to NP. *Information Processing Letters* 58:35-38, 1996.

[Se94] A. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Science* 48:357–381, 1994.

[To90] S. Toda. The complexity of finding medians. Proc. 31st IEEE Annual Symposium on Foundations of Computer Science, 778–787, 1990.

[Va76] L. Valiant. The Relative Complexity of Checking and Evaluating. *Information Processing Letters* 5:20–23, 1976.

[WT93] O. Watanabe and S. Toda. Structural Analysis on the Complexity of Inverse Functions. *Mathematical Systems Theory* 26:203–214, 1993.