The Complexity of Solving Multiobjective Optimization Problems and its Relation to Multivalued Functions

Krzysztof Fleszar [*] Christian Glaßer [*] Fabian Lipp [*] Christian Reitwießner [*] Maximilian Witek [*]

April 11, 2011

Abstract

Instances of optimization problems with multiple objectives can have several optimal solutions whose cost vectors are incomparable. This ambiguity leads to several reasonable notions for solving multiobjective problems. Each such notion defines a class of multivalued functions. We systematically investigate the computational complexity of these classes.

Some solution notions $\mathcal{S}$ turn out to be *equivalent* to NP in the sense that each function in $\mathcal{S}$ has a Turing-equivalent set in NP and each set in NP has a Turing-equivalent function in $\mathcal{S}$. Other solution notions are equivalent to the function class NPMV_g. We give evidence that certain solution notions are not equivalent to NP and NPMV_g. In particular, under suitable assumptions there are functions in NPMV_g that are Turing-inequivalent to all sets. It follows that the complexity of multiobjective problems is in general not expressible in terms of sets.

Moreover, we determine the possible combinations of complexities for every *fixed* multiobjective problem. In particular, for arbitrary $A, B, C \in \mathrm{NP}$ with $A \leq_{\mathrm{T}}^{\mathrm{p}} B \leq_{\mathrm{T}}^{\mathrm{p}} C$ there is a multiobjective problem where one solution notion is Turing-equivalent to A, another one is Turing-equivalent to B, and a third one is Turing-equivalent to C.

1 Introduction

Practical optimization problems often contain multiple objectives. A typical scheduling problem is to order given jobs in a way that minimizes both the lateness and the flow time. Here the quality of a solution is characterized by its cost vector, which is the pair that consists of the lateness and the flow time. This shows that two solutions of a multiobjective problem can have incomparable cost vectors and therefore, a given instance can have several optimal cost vectors. The set of optimal solutions (i.e., solutions with optimal cost vectors) is called the *Pareto set.* It shows the trade-offs between the optimal solutions of the current instance.

The multiple optimal costs make multiobjective problems fundamentally different from single-objective problems. In particular, they differ with respect to their optimization algorithms, their computational complexity, their notions of optimality, their theory of approximation, and the way

[*]Julius-Maximilians-Universität Würzburg, Germany.

solutions are presented to users. Hence single-objective problems cannot adequately represent multiobjective problems and therefore, multiobjective optimization is studied on its own. It has its origins in the 1980s and has become increasingly active since that time.

For a multiobjective problem $\mathcal{O}$ it is not immediately clear what it means to "solve the problem". There exist several reasonable notions. We group them into search notions (which ask for certain solutions) and value notions (which ask for certain cost vectors). For example, the arbitrary optimum search notion of $\mathcal{O}$ (in notation A-$\mathcal{O}$) asks for an arbitrary optimal solution. The specific optimum search notion of $\mathcal{O}$ (in notation S-$\mathcal{O}$) asks for an optimal solution that satisfies a given minimum quality. The corresponding value notion Val(A-$\mathcal{O}$) (resp., Val(S-$\mathcal{O}$)) asks for the cost vector of an arbitrary optimal solution (resp., the cost vector of an optimal solution satisfying a given minimum quality). We also consider the search notions D-$\mathcal{O}$, C-$\mathcal{O}$, L-$\mathcal{O}$, W-$\mathcal{O}$ and their corresponding value notions Val(D-$\mathcal{O}$), Val(C-$\mathcal{O}$), Val(L-$\mathcal{O}$), Val(W-$\mathcal{O}$), which are formally defined in Definition 2.6. On the technical side, all search notions that we consider are multivalued functions from $\mathbb{N}$ to $\mathbb{N}$ (the function maps to solutions). Similarly, all value notions are multivalued functions from $\mathbb{N}$ to $\mathbb{N}^k$ (the function maps to cost vectors). The computational complexity of multivalued functions was first studied by Selman [Sel92, Sel94, Sel96] and further developed by Fenner et al. [FHOS97, FGH$^+$99] and Hemaspaandra et al. [HNOS96].

In this paper we systematically investigate the complexity of multiobjective optimization problems $\mathcal{O}$, i.e., the complexity of the corresponding search and value notions. We determine all possible complexities and integrate them into the picture of existing classes like NP and NPMV$_g$ (the class of multivalued functions whose graph is in P). Our contribution consists of two parts, which will be explained in the following.

1. General complexity of value and search notions There are examples of multiobjective problems that are easy, i.e., solvable in polynomial time, while other multiobjective problems are NP-hard [GRSW10]. Here we investigate what intermediate complexities are possible. We use polynomial-time Turing reducibility to compare the complexity of solution notions of multiobjective problems with sets in NP and multivalued functions in NPMV$_g$. So two problems C and D have the same complexity if they are polynomial-time Turing equivalent (in notation $C \equiv^{\mathrm{p}}_{\mathrm{T}} D$). A complexity class $\mathcal{C}$ can be *embedded* in a complexity class $\mathcal{D}$ if for every $C \in \mathcal{C}$ there exists a $D \in \mathcal{D}$ such that $C \equiv^{\mathrm{p}}_{\mathrm{T}} D$. In this case $\mathcal{D}$ covers all complexities that appear in $\mathcal{C}$. The classes $\mathcal{C}$ and $\mathcal{D}$ are called *equivalent* if they can be embedded in each other. We investigate possible embeddings among the multiobjective solution notions, NP, and NPMV$_g$. In particular we show the following results, where {D-$\mathcal{O}$} is an abbreviation for {D-$\mathcal{O}$ | $\mathcal{O}$ is a multiobjective problem}:

- The following classes are equivalent: NP, $\max \cdot$ NP, {Val(D-$\mathcal{O}$)}, {Val(L-$\mathcal{O}$)}, {Val(W-$\mathcal{O}$)}.
- The following classes are equivalent: NPMV$_g$, {D-$\mathcal{O}$}, {L-$\mathcal{O}$}, {W-$\mathcal{O}$}.

This means that the complexities of the value notion {Val(L-$\mathcal{O}$)} coincide with the complexities of sets in NP, and hence both classes have the same degree structure. On the other hand, the complexities of the search notion {L-$\mathcal{O}$} coincide with the complexities of multivalued functions NPMV$_g$, and hence both classes have the same degree structure. Moreover, we give evidence that certain embeddings do not hold. For example we show:

- NP cannot be embedded in NPMV$_g$ unless EE = NEE.
- NPMV$_g$ cannot be embedded in any class of sets (hence not in NP) unless FewEEE = NEEE.

These results might be of interest on their own, independently of multiobjective problems. In particular, under the assumption FewEEE $\neq$ NEEE there exists a multivalued function $f \in \mathrm{NPMV}_g$ that is inequivalent to all sets (which implies that no partial function $g : \mathbb{N} \to \mathbb{N}$ that is a refinement of f is reducible to f). This shows that the complexity of functions in NPMV_g (resp., the complexity of multiobjective problems) is in general not expressible in terms of sets, unless FewEEE = NEEE. Figure 1 summarizes the obtained embedding results.

2. Complexity settings of value notions for fixed multiobjective problems For every fixed multiobjective problem $\mathcal{O}$ we compare the search and value notions of $\mathcal{O}$ with each other. For every combination we either prove that reducibility holds in general or we show that under a reasonable assumption it does not hold. Figure 2 gives a summary.

There exist examples of multiobjective problems $\mathcal{O}$ where one solution notion is polynomial-time solvable, while another notion is NP-hard [GRSW10]. We investigate this behavior for the value notions and determine the possible combinations of complexities.

- If $A, L, W \in \mathrm{NP}$ and $A \leq^{\mathrm{p}}_{\mathrm{T}} L \leq^{\mathrm{p}}_{\mathrm{T}} W$, then there is a multiobjective problem $\mathcal{O}$ such that $A \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{A-}\mathcal{O})$, $L \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{L-}\mathcal{O})$, and $W \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{W-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{D-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{C-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{S-}\mathcal{O})$.

As a consequence, there exists a multiobjective problem $\mathcal{O}$ such that $\mathcal{O}$'s arbitrary optimum value notion $\mathrm{Val}(\text{A-}\mathcal{O})$ is solvable in polynomial-time, $\mathcal{O}$'s lexicographic optimum value notion $\mathrm{Val}(\text{L-}\mathcal{O})$ is equivalent to the factorization problem of natural numbers, and $\mathcal{O}$'s constraint optimum value notion $\mathrm{Val}(\text{C-}\mathcal{O})$ is equivalent to SAT.

2 Preliminaries

2.1 Computational Complexity

Let $\mathbb{N}$ denote the set of non-negative integers. For $n \in \mathbb{N}$, $\mathrm{bin}(n)$ denotes the binary representation of n and $|n| = |\mathrm{bin}(n)|$. The logarithm to base 2 is denoted by log. For every $k \geq 1$ let $\langle \cdot, \cdot, \ldots, \cdot \rangle$ be a polynomial-time computable and polynomial-time invertible bijection from $\mathbb{N}^k$ to $\mathbb{N}$ that is monotone in each argument.

Let A and B be sets. A *multivalued function* from A to B is a total function $A \to 2^B$. For a multivalued function f from A to B, define $\mathrm{supp}(f) = \{x \mid f(x) \neq \emptyset\}$, $\mathrm{graph}(f) = \{(x, y) \mid y \in f(x)\}$, and $\mathrm{range}(f) = \bigcup_{x \in A} f(x)$. A multivalued function g is a *refinement* of a multivalued function f, if $\mathrm{supp}(g) = \mathrm{supp}(f)$ and for all x, $g(x) \subseteq f(x)$. A partial function g is a *refinement* of a multivalued function f, if for all x, $f(x) = \emptyset$ if g is not defined at x and $g(x) \in f(x)$ otherwise. The complexity classes used in this paper are defined in Figure 3. We denote the complement of a set $A \subseteq \mathbb{N}$ by $\overline{A} = \mathbb{N} - A$. Let $\mathcal{C}$ be a complexity class containing subsets of $\mathbb{N}$. The class of complements of $\mathcal{C}$ is denoted by $\mathrm{co}\mathcal{C} = \{\overline{A} \mid A \in \mathcal{C}\}$. An infinite and co-infinite set $L \subseteq \mathbb{N}$ is *$\mathcal{C}$-bi-immune* if neither L nor $\overline{L}$ has an infinite subset in $\mathcal{C}$ [BS85].

For reductions between multivalued functions we need the following definition by Fenner et al. [FHOS97] which describes how a deterministic Turing transducer M [BLS84] accesses a partial function g as oracle. For this, M contains a write-only oracle input tape, a separate read-only oracle

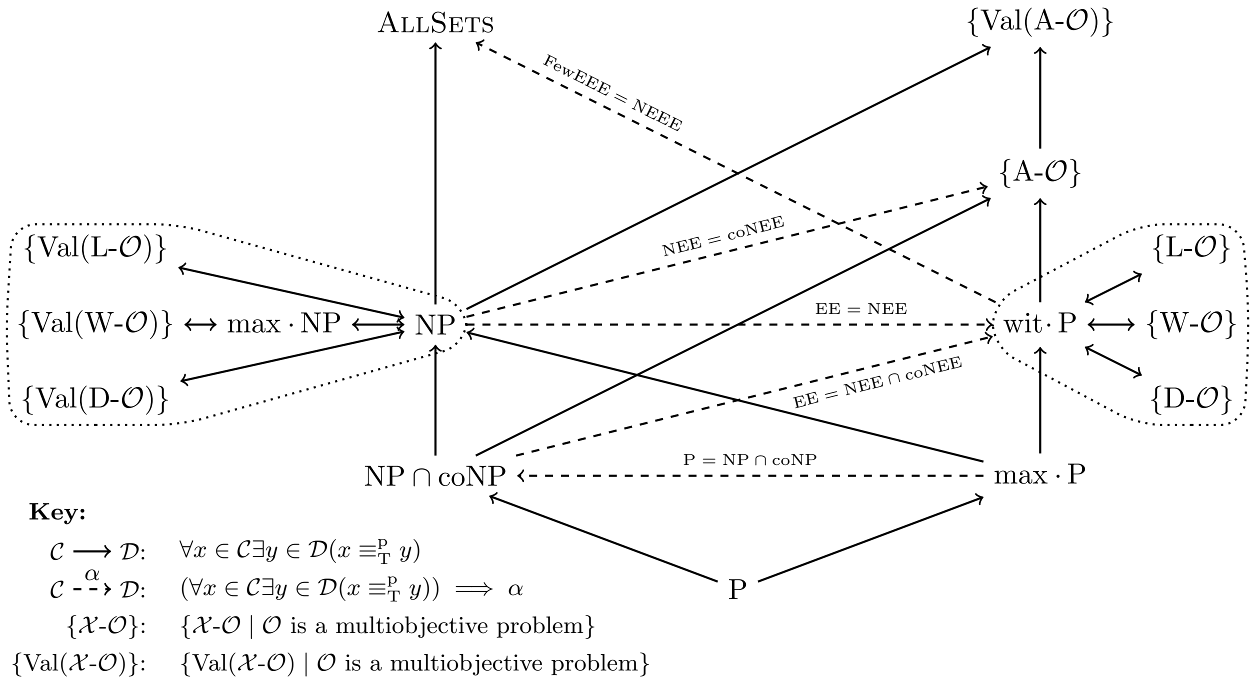

AllSets
{Val(A-$\mathcal{O}$)}
FewEEE = NEEE
{A-$\mathcal{O}$}
NEE = coNEE
{Val(L-$\mathcal{O}$)}
{L-$\mathcal{O}$}
{Val(W-$\mathcal{O}$)}
max · NP
NP
EE = NEE
wit· P
{W-$\mathcal{O}$}
{Val(D-$\mathcal{O}$)}
EE = NEE ∩ coNEE
{D-$\mathcal{O}$}
P = NP ∩ coNP
NP ∩ coNP
max · P
P
Key:
$\mathcal{C} \longrightarrow \mathcal{D}$: $\forall x \in \mathcal{C} \exists y \in \mathcal{D}(x \equiv^{\mathrm{p}}_{\mathrm{T}} y)$
$\mathcal{C} \overset{\alpha}{\dashrightarrow} \mathcal{D}$: $(\forall x \in \mathcal{C} \exists y \in \mathcal{D}(x \equiv^{\mathrm{p}}_{\mathrm{T}} y)) \implies \alpha$
$\{\mathcal{X}\text{-}\mathcal{O}\}$: $\{\mathcal{X}\text{-}\mathcal{O} \mid \mathcal{O}$ is a multiobjective problem$\}$
$\{\mathrm{Val}(\mathcal{X}\text{-}\mathcal{O})\}$: $\{\mathrm{Val}(\mathcal{X}\text{-}\mathcal{O}) \mid \mathcal{O}$ is a multiobjective problem$\}$

Figure 1: Summary of embeddings of complexity classes. A bold arrow from $\mathcal{C}$ to $\mathcal{D}$ shows that $\mathcal{C}$ can be embedded in $\mathcal{D}$. Dashed arrows give evidence against such an embedding. Observe that the embedding relation is reflexive and transitive and that evidence against an embedding propagates along bold lines (heads of dashed arrows can be moved downwards, tails can be moved upwards), and hence for each pair of classes $\mathcal{C}, \mathcal{D}$ in the diagram, we either show that $\mathcal{C}$ is embedded in $\mathcal{D}$ or give evidence against such an embedding. Note that wit$\cdot$ P = NPMV$_g$, max $\cdot$ NP = OptP (Krentel [Kre88]) and AllSets is the class of all decision problems.

output tape, and a special oracle call state q. When M enters the state q, if the oracle g is defined at the string x currently on the oracle input tape, then $g(x)$ appears on the oracle output tape. If it is not defined at this point, then the special symbol $\perp$ appears on the oracle output tape. Note that it is possible that M may read only a portion of the oracle's output if the oracle's output is too long to read with the resources of M. If M computes a partial function and the function is not defined on input x, M can either not halt at all or return the special symbol $\perp$. This allows deterministic polynomial-time Turing transducers to compute non-total functions. If g is a partial function and M is a deterministic oracle Turing transducer as just described, then let M^g denote the partial function computed by M with oracle g.

Definition 2.1 ([FHOS97])**.**

1. *Let f and g be partial functions.* f is polynomial-time Turing reducible to g, *$f \leq^{\mathrm{p}}_{\mathrm{T}} g$, if there exists a deterministic, polynomial-time oracle Turing transducer M such that $f = M^g$.*

2. *Let f and g be multivalued functions.* f is polynomial-time Turing reducible to g, *$f \leq^{\mathrm{p}}_{\mathrm{T}} g$, if there exists a deterministic, polynomial-time oracle Turing transducer M such that for every partial function g' that is a refinement of g it holds that the partial function $M^{g'}$ is a refinement of f.*

$$\begin{tikzcd}
 & \mathrm{SAT} & & \\
\text{W-}\mathcal{O} \arrow[ur] & & & \text{D-}\mathcal{O} \arrow[ull] \\
\mathrm{Val}(\text{W-}\mathcal{O}) \arrow[u] \arrow[urrr, dashed, "\mathrm{P} = \mathrm{NP}"] & & \text{L-}\mathcal{O} \arrow[ull] \arrow[ur] & \mathrm{Val}(\text{D-}\mathcal{O}) \arrow[u] \arrow[ulll, dashed, "\mathrm{P} = \mathrm{NP}"'] \\
 & \text{A-}\mathcal{O} \arrow[ur] \arrow[ul, dashed, "\mathrm{EE} = \mathrm{NEE} \wedge \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}"] \arrow[urr, dashed, "\mathrm{EE} = \mathrm{NEE} \wedge \mathrm{P} = \mathrm{NP} \cap \mathrm{coNP}"'] & \mathrm{Val}(\text{L-}\mathcal{O}) \arrow[u] \arrow[ull] \arrow[ur] \arrow[l, dashed, bend left, "\mathrm{P} = \mathrm{NP}"] & \\
 & \mathrm{Val}(\text{A-}\mathcal{O}) \arrow[u] \arrow[ur] & &
\end{tikzcd}$$

Key:

$X \longrightarrow Y$: $X \le_{\mathrm{T}}^{\mathrm{p}} Y$ for all $\mathcal{O}$

$X \overset{\alpha}{\dashrightarrow} Y$: $X \le_{\mathrm{T}}^{\mathrm{p}} Y$ for all $\mathcal{O} \implies \alpha$

Figure 2: A complete taxonomy of reductions among search and value notions. Bold arrows indicate reducibility for all problems $\mathcal{O}$ (reductions including weighted sum notions hold if all objectives are to be maximized or all objectives are to be minimized), whereas dashed arrows provide evidence against such a general reducibility. Observe that such evidence propagates along bold arrows (arrow heads backwards and arrow tails forwards) and we hence have evidence against all remaining possible reductions. Further note that D-$\mathcal{O} \equiv_{\mathrm{T}}^{\mathrm{p}}$ S-$\mathcal{O} \equiv_{\mathrm{T}}^{\mathrm{p}}$ $\mathrm{C_i}$-$\mathcal{O}$ and $\mathrm{Val}(\text{D-}\mathcal{O}) \equiv_{\mathrm{T}}^{\mathrm{p}} \mathrm{Val}(\text{S-}\mathcal{O}) \equiv_{\mathrm{T}}^{\mathrm{p}} \mathrm{Val}(\mathrm{C_i}\text{-}\mathcal{O})$ for $i \in \{1, \dots, k\}$.

It is important to note that the definition above is different from the one given by Selman [Sel94]. In Selman's definition, if the oracle g is a multivalued function and if some q with $g(q) = \emptyset$ is queried, then the oracle can give an arbitrary answer. Also note that the oracle model described above ensures that $\le_{\mathrm{T}}^{\mathrm{p}}$ is reflexive and transitive.

The decision problem of a set A is the computation of the characteristic function χ_A, which can be considered as a multivalued function. In this way, the polynomial-time Turing reducibility defined above also applies to decision problems.

A multivalued function g is called *polynomial-time solvable*, if there is a polynomial-time computable, partial function f such that f is a refinement of g. A multivalued function g is called NP-*hard*, if all problems in NP are polynomial-time Turing-reducible to g.

For a set $A \subseteq \mathbb{N}$ and a total function $p\colon \mathbb{N} \to \mathbb{N}$ we define the multivalued function $\mathrm{wit}_p\cdot A\colon \mathbb{N} \to 2^{\mathbb{N}}$, $x \mapsto \{y \mid \langle x, y\rangle \in A \text{ and } y < 2^{p(|x|)}\}$, the total function $\max_p \cdot A\colon \mathbb{N} \to \mathbb{N}$, $x \mapsto \max(\{0\} \cup \mathrm{wit}_p\cdot A(x))$, and the set $\exists_p\cdot A = \mathrm{supp}(\mathrm{wit}_p\cdot A)$. Moreover, let $\mathrm{wit}\cdot A = \{\mathrm{wit}_p\cdot A \mid p \text{ is a polynomial}\}$, $\max\cdot A = \{\max_p\cdot A \mid p \text{ is a polynomial}\}$, and $\exists\cdot A = \{\exists_p\cdot A \mid p \text{ is a polynomial}\}$. For a complexity class $\mathcal{C}$, define $\mathrm{wit}\cdot\mathcal{C} = \bigcup_{A\in\mathcal{C}} \mathrm{wit}\cdot A$, $\max\cdot\mathcal{C} = \bigcup_{A\in\mathcal{C}} \max\cdot A$, and $\exists\cdot\mathcal{C} = \bigcup_{A\in\mathcal{C}} \exists\cdot A$.

Classes like $\max\cdot\mathrm{P}$ and $\max\cdot\mathrm{NP}$ were systematically studied by Hempel and Wechsung [HW00]. Moreover, the classes $\mathrm{wit}\cdot\mathrm{P}$, $\mathrm{wit}\cdot\mathrm{NP}$, and $\mathrm{wit}\cdot\mathrm{coNP}$ were studied under the names $\mathrm{NPMV_g}$, NPMV,

$$
\begin{aligned}
\mathrm{PF} &= \{f \mid f\colon \mathbb{N}\to\mathbb{N} \text{ is a partial function that is polynomial-time computable}\}\\
\mathrm{NPMV} &= \{f \mid f \text{ multivalued function from } \mathbb{N} \text{ to } \mathbb{N},\ \mathrm{graph}(f)\in\mathrm{NP}, \text{ and } \exists \text{ polynomial } p,\ \forall (x,y)\in\mathrm{graph}(f)\ [y<2^{p(|x|)}]\}\\
\mathrm{coNPMV} &= \{f \mid f \text{ multivalued function from } \mathbb{N} \text{ to } \mathbb{N},\ \mathrm{graph}(f)\in\mathrm{coNP}, \text{ and } \exists \text{ polynomial } p,\ \forall (x,y)\in\mathrm{graph}(f)\ [y<2^{p(|x|)}]\}\\
\mathrm{NPMV_g} &= \{f\in\mathrm{NPMV} \mid \mathrm{graph}(f)\in\mathrm{P}\}\\
\mathrm{EE} &= \mathrm{DTIME}(2^{2^{O(n)}})\\
\mathrm{NEE} &= \mathrm{NTIME}(2^{2^{O(n)}})\\
\mathrm{NEEE} &= \mathrm{NTIME}(2^{2^{2^{O(n)}}})\\
\mathrm{UP} &= \{L\in\mathrm{NP} \mid L \text{ is accepted by a nondet. machine } N \text{ in time } n^{O(1)} \text{ s.t. } N \text{ on } x \text{ has} \le 1 \text{ accepting paths}\}\\
\mathrm{UEEE} &= \{L\in\mathrm{NEEE} \mid L \text{ is accepted by a nondet. machine } N \text{ in time } 2^{2^{2^{O(n)}}} \text{ s.t. } N \text{ on } x \text{ has} \le 1 \text{ accepting paths}\}\\
\mathrm{FewP} &= \{L\in\mathrm{NP} \mid L \text{ is accepted by a nondet. machine } N \text{ in time } n^{O(1)} \text{ s.t. } N \text{ on } x \text{ has} \le n^{O(1)} \text{ accepting paths}\}\\
\mathrm{FewEEE} &= \{L\in\mathrm{NEEE} \mid L \text{ is accepted by a nondet. machine } N \text{ in time } 2^{2^{2^{O(n)}}} \text{ s.t. } N \text{ on } x \text{ has} \le 2^{2^{2^{O(n)}}} \text{ accepting paths}\}
\end{aligned}
$$

Figure 3: Definitions of some complexity classes.

and coNPMV by Selman [Sel92, Sel94, Sel96], Fenner et al. [FHOS97, FGH$^+$99], and Hemaspaandra et al. [HNOS96].

Proposition 2.2.

1. $\mathrm{wit}\cdot\mathrm{P} = \mathrm{NPMV_g}$.
2. $\mathrm{wit}\cdot\mathrm{NP} = \mathrm{NPMV}$.
3. $\mathrm{wit}\cdot\mathrm{coNP} = \mathrm{coNPMV}$.

Proof. 1. "$\supseteq$": Let $f\in\mathrm{NPMV_g}$. So $\mathrm{graph}(f)\in\mathrm{P}$ and there exists a polynomial p such that for all $(x,y)\in\mathrm{graph}(f)$, $y<2^{p(|x|)}$. Hence the set $R=\{\langle x,y\rangle \mid (x,y)\in\mathrm{graph}(f)\}$ belongs to P. Moreover, $f=\mathrm{wit}_p\cdot R$.
"$\subseteq$": Let $f\in\mathrm{wit}\cdot\mathrm{P}$, i.e., there exists a polynomial p and an $R\in\mathrm{P}$ such that $f=\mathrm{wit}_p\cdot R$. In particular, for all $(x,y)\in\mathrm{graph}(f)$, $y<2^{p(|x|)}$. Note that $\mathrm{graph}(f)\in\mathrm{P}$. Hence $f\in\mathrm{NPMV_g}$.

2. and 3. follow immediately from the definitions of NPMV, wit$\cdot$NP and coNPMV, wit$\cdot$coNP. □

We show that NP and max $\cdot$ NP are equivalent. In particular, all sets in NP are equivalent to some function from max $\cdot$ NP. The latter might not be true for max $\cdot$ P (Corollary 4.11).

Proposition 2.3. *1. For every $g\in\max\cdot\mathrm{NP}$ there exists a $B\in\mathrm{NP}$ such that $g\equiv^{\mathrm{p}}_{\mathrm{T}} B$.*

2. For every $B\in\mathrm{NP}$ there exists a $g\in\max\cdot\mathrm{NP}$ such that $B\equiv^{\mathrm{p}}_{\mathrm{T}} g$.

Proof. 1. Choose a polynomial p and $R\in\mathrm{NP}$ such that $g=\max_p\cdot R$. Let $B=\{\langle x,y\rangle \mid g(x)\ge y\}$. Observe that $B\in\mathrm{NP}$ and $B\equiv^{\mathrm{p}}_{\mathrm{T}} g$. 2. Let $R=\{\langle x,1\rangle \mid x\in B\}$ and note that $R\in\mathrm{NP}$. Define $p(n)=1$ and $g=\max_p\cdot R$. So g is a total function $\mathbb{N}\to\{0,1\}$. It holds that $(g(x)=1 \iff x\in B)$ and hence $g\equiv^{\mathrm{p}}_{\mathrm{T}} B$. □

Under the assumption $\mathrm{P}\ne\mathrm{NP}\cap\mathrm{coNP}$, the class max $\cdot$ P cannot be embedded in $\mathrm{NP}\cap\mathrm{coNP}$.

Proposition 2.4. *If* $\mathrm{P} \neq \mathrm{NP} \cap \mathrm{coNP}$*, then there exists some* $g \in \max \cdot \mathrm{P}$ *such that for all* $L \in \mathrm{NP} \cap \mathrm{coNP}$ *we have* $g \not\leq^{\mathrm{p}}_{\mathrm{T}} L$*.*

Proof. Assume that for all $g \in \max \cdot P$ there is some $L' \in \mathrm{NP} \cap \mathrm{coNP}$ such that $g \leq^{\mathrm{p}}_{\mathrm{T}} L'$. Let $L \in \mathrm{NP}$, we show $L \in \mathrm{NP} \cap \mathrm{coNP}$: $L = \exists_p \cdot R$ for some polynomial p and $R \in \mathrm{P}$. Let $g = \max_p \cdot R$ and observe that $L \leq^{\mathrm{p}}_{\mathrm{T}} g$. By assumption, there is some $L' \in \mathrm{NP} \cap \mathrm{coNP}$ such that $g \leq^{\mathrm{p}}_{\mathrm{T}} L'$ and hence $L \leq^{\mathrm{p}}_{\mathrm{T}} L'$. So $L \in \mathrm{NP} \cap \mathrm{coNP}$, since $\mathrm{NP} \cap \mathrm{coNP}$ is closed under $\leq^{\mathrm{p}}_{\mathrm{T}}$. □

With standard padding techniques we construct several very sparse sets in NP under the assumption that certain super-exponential time classes do not coincide.

Proposition 2.5.

1. *If* $\mathrm{EE} \neq \mathrm{NEE}$*, then there exists a* $B \in \mathrm{NP} - \mathrm{P}$ *such that* $B \subseteq \{2^{2^{x^c}} \mid x \in \mathbb{N}\}$ *for some* $c \geq 1$*.*
2. *If* $\mathrm{EE} \neq \mathrm{NEE} \cap \mathrm{coNEE}$*, then there exists a* $B \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$ *such that* $B \subseteq \{2^{2^{x^c}} \mid x \in \mathbb{N}\}$ *for some* $c \geq 1$*.*
3. *If* $\mathrm{NEE} \neq \mathrm{coNEE}$*, then there exists a* $B \in \mathrm{NP} - \mathrm{coNP}$ *such that* $B \subseteq \{2^{2^{x^c}} \mid x \in \mathbb{N}\}$ *for some* $c \geq 1$*.*
4. *If* $\mathrm{FewEEE} \neq \mathrm{NEEE}$*, then there exists a* $B \in \mathrm{NP} - \mathrm{FewP}$ *such that* $B \subseteq \{t(c \cdot i) + k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$ *for some* $c \geq 1$ *and* $t(n) = 2^{2^{2^{2^n}}}$*.*
5. *If* $\mathrm{UEEE} \cap \mathrm{coUEEE} \neq \mathrm{NEEE} \cap \mathrm{coNEEE}$*, then there exists a* $B \in (\mathrm{NP} \cap \mathrm{coNP}) - (\mathrm{UP} \cap \mathrm{coUP})$ *such that* $B \subseteq \{t(c \cdot i) + k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$ *for some* $c \geq 1$ *and* $t(n) = 2^{2^{2^{2^n}}}$*.*

Proof. For $L \subseteq \mathbb{N}$ and $c \in \mathbb{N} - \{0\}$ let $B(L, c) = \{2^{2^{x^c}} \mid x \in L\}$. We claim:

$$L \in \mathrm{DTIME}(2^{2^{c \cdot n}}) \iff B(L,c) \in \mathrm{P} \tag{1}$$

$$L \in \mathrm{NTIME}(2^{2^{c \cdot n}}) \iff B(L,c) \in \mathrm{NP} \tag{2}$$

$$L \in \mathrm{coNTIME}(2^{2^{c \cdot n}}) \iff B(L,c) \in \mathrm{coNP} \tag{3}$$

If $L \in \mathrm{DTIME}(2^{2^{c \cdot n}})$, then $B(L,c) \in \mathrm{P}$ by the algorithm that on input $y = 2^{2^{x^c}}$ simulates N on x in deterministic polynomial time in $|y|$ (N on x needs time $2^{2^{c \cdot |x|}} \leq 2^{2^{c \cdot (1 + \log x)}} = 2^{2^c \cdot x^c} = (\log y)^{2^c} \leq |y|^{2^c} \leq |y|^{c'}$). If $B(L,c) \in \mathrm{P}$, then $L \in \mathrm{DTIME}(2^{2^{c \cdot n}})$ by the algorithm that on input x simulates the deterministic polynomial-time algorithm for $B(L,c)$ on input $y = 2^{2^{x^c}}$ (the simulation needs time $|y|^{c''} \leq (\log y)^{c''+1} = (2^{x^c})^{c''+1} \leq 2^{x^d} = 2^{2^{d \log x}} \leq 2^{2^{d|x|}}$). Analogously one shows (2) and (3).

1. If $\mathrm{EE} \neq \mathrm{NEE}$, then let $L \in \mathrm{NEE} - \mathrm{EE}$. Choose $c \geq 1$ such that $L \in \mathrm{NTIME}(2^{2^{c \cdot n}})$. By (1) and (2), $B(L,c) \in \mathrm{NP} - \mathrm{P}$.

2. If $\mathrm{EE} \neq \mathrm{NEE} \cap \mathrm{coNEE}$, then let $L \in (\mathrm{NEE} \cap \mathrm{coNEE}) - \mathrm{EE}$. Choose $c \geq 1$ such that $L, \overline{L} \in \mathrm{NTIME}(2^{2^{c \cdot n}})$. By (1)–(3), $B(L,c) \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$.

3. If $\mathrm{NEE} \neq \mathrm{coNEE}$, then let $L \in \mathrm{NEE} - \mathrm{coNEE}$. Choose $c \geq 1$ such that $L \in \mathrm{NTIME}(2^{2^{c \cdot n}})$. By (2) and (3), $B(L,c) \in \mathrm{NP} - \mathrm{coNP}$.

4. Let $L \in \mathrm{NEEE} - \mathrm{FewEEE}$ and choose some $c \in \mathbb{N} - \{0\}$ such that L is decidable by a nondeterministic machine N that works in time $2^{2^{2^{c \cdot n}}}$. Let $B = \{t(c \cdot |x|) + x \mid x \in L\}$ and note that $B \subseteq \{t(c \cdot i) + k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$. We show $B \in \mathrm{NP} - \mathrm{FewP}$: $B \in \mathrm{NP}$ by the algorithm that on input $y = t(c \cdot |x|) + x$ simulates N on x in nondeterministic polynomial time in $|y|$ (N on x needs time $2^{2^{2^{c \cdot |x|}}} = \log t(c \cdot |x|) \leq \log y \leq |y|$). $B \notin \mathrm{FewP}$, since otherwise $L \in \mathrm{FewEEE}$ by the algorithm that on input x simulates the FewP-algorithm for B on input $y = t(c \cdot |x|) + x$ (the simulation works in time $|y|^{c'} \leq (1 + \log y)^{c'} \leq (\log y)^{c'+1} \leq (\log 2t(c \cdot |x|))^{c'+1} = (2^{2^{2^{c \cdot |x|}}} + 1)^{c'+1} \leq (2^{2^{2^{c \cdot |x|}}})^{c'+2} \leq 2^{2^{2^{c'' \cdot |x|}}}$ and similarly we see that the number of accepting paths is lower equal $|y|^{d'} \leq 2^{2^{2^{d'' \cdot |x|}}}$).

5. Let $L \in (\mathrm{NEEE} \cap \mathrm{coNEEE}) - (\mathrm{UEEE} \cap \mathrm{coUEEE})$ and choose some $c \in \mathbb{N} - \{0\}$ such that L (resp., $\overline{L}$) is decidable by a nondeterministic machine N (resp, $\overline{N}$) that works in time $2^{2^{2^{c \cdot n}}}$. Let $B = \{t(c \cdot |x|) + x \mid x \in L\}$ and note that $B \subseteq \{t(c \cdot i) + k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$. We show $B \in (\mathrm{NP} \cap \mathrm{coNP}) - (\mathrm{UP} \cap \mathrm{coUP})$: $B \in \mathrm{NP}$ by the algorithm that on input $y = t(c \cdot |x|) + x$ simulates N on x in nondeterministic polynomial time in $|y|$ (N on x needs time $2^{2^{2^{c \cdot |x|}}} = \log t(c \cdot |x|) \leq \log y \leq |y|$). Similarly, $\overline{B} \in \mathrm{NP}$ by the algorithm that on input $y = t(c \cdot |x|) + x$ simulates N' on x in nondeterministic polynomial time in $|y|$. $B \notin (\mathrm{UP} \cap \mathrm{coUP})$, since otherwise $L \in \mathrm{UEEE} \cap \mathrm{coUEEE}$ by the algorithm that on input x simulates the $(\mathrm{UP} \cap \mathrm{coUP})$-algorithm for B on input $y = t(c \cdot |x|) + x$ (the simulation works in time $|y|^{c'} \leq (1 + \log y)^{c'} \leq (\log y)^{c'+1} \leq (\log 2t(c \cdot |x|))^{c'+1} = (2^{2^{2^{c \cdot |x|}}} + 1)^{c'+1} \leq (2^{2^{2^{c \cdot |x|}}})^{c'+2} \leq 2^{2^{2^{c'' \cdot |x|}}}$). $\square$

2.2 Multiobjective Optimization Problems

Let $k \geq 1$. A *k-objective* NP *optimization problem* (*k-objective problem*, for short) is a tuple $(S, f, \leftarrow)$ where

- $S \colon \mathbb{N} \to 2^{\mathbb{N}}$ maps an instance $x \in \mathbb{N}$ to the set of feasible solutions for this instance, denoted as $S^x = S(x) \subseteq \mathbb{N}$. There must be some polynomial p such that for every $x \in \mathbb{N}$ and every $s \in S^x$ it holds that $|s| \leq p(|x|)$ and the set $\{\langle x, s \rangle \mid x \in \mathbb{N}, s \in S^x\}$ must be polynomial-time decidable, i.e., $S \in \mathrm{wit}\cdot\mathrm{P}$.
- $f \colon \{\langle x, s \rangle \mid x \in \mathbb{N}, s \in S^x\} \to \mathbb{N}^k$ maps an instance $x \in \mathbb{N}$ and a solution $s \in S^x$ to its value, denoted by $f^x(s) \in \mathbb{N}^k$. The function f must be polynomial-time computable.
- $\leftarrow \subseteq \mathbb{N}^k \times \mathbb{N}^k$ is a partial order on the values of solutions. It must hold that $(a_1, \ldots, a_k) \leftarrow (b_1, \ldots, b_k) \iff a_1 \leftarrow_1 b_1 \wedge \cdots \wedge a_k \leftarrow_k b_k$, where $\leftarrow_i$ is $\leq$ if the i-th objective is minimized, and $\leftarrow_i$ is $\geq$ if the i-th objective is maximized.

We also use $\leq$ as the partial order $\leftarrow$ where $\leftarrow_i = \leq$ for all i and $\geq$ is used analogously.

The superscript x of f and S can be omitted if it is clear from context. The projection of f^x to the i-th component is denoted as f_i^x where $f_i^x(s) = v_i$ if $f^x(s) = (v_1, \ldots, v_k)$. If $a \leftarrow b$ we say that *a weakly dominates b* (i.e., a is at least as good as b). If $a \leftarrow b$ and $a \neq b$ we say that *a dominates b*. Note that $\leftarrow$ always points in the direction of the better value. If f and x are clear from the context, then we extend $\leftarrow$ to combinations of values and solutions. So we can talk about weak dominance between solutions, and we write $s \leftarrow t$ if $f^x(s) \leftarrow f^x(t)$, $s \leftarrow c$ if $f^x(s) \leftarrow c$, and so on, where $s, t \in S^x$ and

$c \in \mathbb{N}^k$. Furthermore, we define $\mathrm{opt}_{\leftarrow} \colon 2^{\mathbb{N}^k} \to 2^{\mathbb{N}^k}$, $\mathrm{opt}_{\leftarrow}(M) = \{y \in M \mid \forall z \in M[z \leftarrow y \Rightarrow z = y]\}$ as a function that maps sets of values to sets of optimal values. The operator $\mathrm{opt}_{\leftarrow}$ is also applied to sets of solutions $S' \subseteq S^x$ as $\mathrm{opt}_{\leftarrow}(S') = \{s \in S' \mid f^x(s) \in \mathrm{opt}_{\leftarrow}(f^x(S'))\}$. If even $\leftarrow$ is clear from the context, we write $S^x_{\mathrm{opt}} = \mathrm{opt}_{\leftarrow}(S^x)$ and $\mathrm{opt}_i(S') = \{s \in S' \mid f^x_i(s) \in \mathrm{opt}_{\leftarrow_i}(f^x_i(S'))\}$.

Definition 2.6. *For every k-objective* NP *optimization problem $\mathcal{O} = (S, f, \leftarrow)$ where $k \geq 1$ and all $1 \leq i \leq k$ we define the search notions* arbitrary optimum (A-$\mathcal{O}$), dominating solution (D-$\mathcal{O}$), specific optimum (S-$\mathcal{O}$), constraint optimum (C$_\mathrm{i}$-$\mathcal{O}$), lexicographic optimum (L-$\mathcal{O}$)*, and* weighted sum optimum (W-$\mathcal{O}$) *as multivalued functions from $\mathbb{N}$ to $\mathbb{N}$, where*

$$\begin{aligned}
\text{A-}\mathcal{O}(x) &= S^x_{\mathrm{opt}} \\
\text{D-}\mathcal{O}(\langle x, \langle c\rangle\rangle) &= \{y \in S^x \mid y \leftarrow c\} \\
\text{S-}\mathcal{O}(\langle x, \langle c\rangle\rangle) &= \left\{y \in S^x_{\mathrm{opt}} \mid y \leftarrow c\right\} \\
\text{C}_\text{i}\text{-}\mathcal{O}(\langle x, \langle c\rangle\rangle) &= \mathrm{opt}_i\left(\left\{s \in S^x \mid f^x_j(s) \leftarrow_j c_j \text{ for all } j \neq i\right\}\right) \\
\text{L-}\mathcal{O}(x) &= \mathrm{opt}_k(\dots(\mathrm{opt}_2(\mathrm{opt}_1(S^x)))\dots) \\
\text{W-}\mathcal{O}(\langle x, \langle \omega\rangle\rangle) &= \{y \in S^x \mid \forall s \in S^x\,[w^x_\omega(y) \leftarrow_1 w^x_\omega(s)]\}
\end{aligned}$$

for all $x \in \mathbb{N}$ and $c, \omega \in \mathbb{N}^k$, where $w^x_\omega(y) = \sum_{j=1}^k \omega_j f^x_j(y)$ for all $y \in S^x$. For the weighted sum optimum notion, we assume that all objectives are to be maximized or all objectives are to be minimized.

The arbitrary optimum notion of $\mathcal{O}$ maps input instances to all optimal solutions and hence is polynomial-time solvable if for all input instances $x \in \mathbb{N}$ we can decide if $S^x \neq \emptyset$ and further find some arbitrary optimal solution $s \in S^x_{\mathrm{opt}}$ in polynomial time. Analogously, the specific optimum notion searches for optimal solutions that are restricted to be at least as good as the constraint vector $c \in \mathbb{N}^k$, whereas the dominating solution notion does not require the solutions to be optimal. The constraint optimum notion for the i-th objective searches solutions that are at least as good as c for all objectives $j \neq i$ and optimal for objective i, while the lexicographical optimum notion searches for solutions that are optimal according to some fixed order of objectives (here: $1, 2, \dots, k$). Finally, the weighted sum notion searches for solutions such that the sum of all objectives weighted with the weight vector $\omega \in \mathbb{N}^k$ is optimal.

Note that the weighted sum notion takes $\leftarrow_1$ as the partial order of the weighted sum of values of solutions, since optimizing the weighted sum only makes sense if all objectives are to be minimized or all objectives are to be maximized. This notion plays a special role as it combines multiple objectives into a single function and thus turns out to be equivalent to a single-objective problem.

Proposition 2.7. *For every k-objective* NP *optimization problem $\mathcal{O} = (S, f, \leftarrow)$ where all objectives are to be maximized (resp., minimized) there is a single-objective problem $\mathcal{O}'$ such that* W-$\mathcal{O}$ = A-$\mathcal{O}'$.

Proof. Let $\mathcal{O}' = (S', f', \leftarrow_1)$ with $S'^{\langle x, \langle \omega_1, \dots, \omega_k\rangle\rangle} = S^x$ and $f'^{\langle x, \langle \omega_1, \dots, \omega_k\rangle\rangle}(s) = \sum_{i=1}^k \omega_i f^x_i(s)$. □

We refer to [GRSW10] for a more detailed introduction to solution notions of multiobjective problems.

Each search notion maps to sets of solutions, which, in turn, map to values in $\mathbb{N}^k$ via f. Hence, each search notion naturally motivates a *value notion* for the problem.

Definition 2.8. *For every k-objective* NP *optimization problem $\mathcal{O} = (S, f, \leftarrow)$ we define the* value notion $\mathrm{Val}(\mathcal{X}\text{-}\mathcal{O})$ *as a multivalued function from $\mathbb{N}$ to $\mathbb{N}^k$, where*

$$\mathrm{Val}(\mathcal{X}\text{-}\mathcal{O})(\varphi) \quad = \quad f^x(\mathcal{X}\text{-}\mathcal{O}(\varphi))$$

for all $\varphi \in \mathbb{N}$ and $\mathcal{X} \in \{\mathrm{A}, \mathrm{D}, \mathrm{S}, \mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}, \mathrm{L}, \mathrm{W}\}$, where x is the problem instance encoded in φ, and $\mathcal{X} = W$ only if all objectives are to be maximized (resp., minimized).

We show that we can restrict to multiobjective problems whose objectives are all to be maximized.

Proposition 2.9. *For every k-objective* NP *optimization problem $\mathcal{O} = (S, f, \leftarrow)$ there is a k-objective* NP *optimization problem $\mathcal{O}' = (S, f', \geq)$ such that for all $\mathcal{X} \in \{\mathrm{A}, \mathrm{D}, \mathrm{S}, \mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}, \mathrm{L}, \mathrm{W}\}$*

$$\mathcal{X}\text{-}\mathcal{O} \equiv^{\mathrm{p}}_{\mathrm{T}} \mathcal{X}\text{-}\mathcal{O}' \qquad \textit{and} \qquad \mathrm{Val}(\mathcal{X}\text{-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\mathcal{X}\text{-}\mathcal{O}')$$

(where $\mathcal{X} = W$ is only considered for $\leftarrow \in \{\leq, \geq\}$).

Proof. Since f must be polynomial-time computable, there is a polynomial p such that for every $i \in \{1, \ldots, k\}$, $f_i^x(s) \leq 2^{p(|x|)}$. For every i such that $\leftarrow_i = \leq$, let $f'^x_i(s) = 2^{p(|x|)} - f_i^x(s)$ and $f'^x_i(s) = f_i^x(s)$ for all other i. Observe that the assertions hold. □

We obtain the following upper bounds for the search and value notions.

Proposition 2.10. *Let $\mathcal{O} = (S, f, \leftarrow)$ be a k-objective* NP *optimization problem.*

1. $\mathcal{X} \in \{\mathrm{A}, \mathrm{S}, \mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}, \mathrm{L}, \mathrm{W}\} \implies \mathcal{X}\text{-}\mathcal{O} \in \mathrm{coNPMV}$ *and* $\mathrm{Val}(\mathcal{X}\text{-}\mathcal{O}) \in \mathrm{coNPMV}$.
2. $\mathrm{D}\text{-}\mathcal{O} \in \mathrm{NPMV_g}$ *and* $\mathrm{Val}(\mathrm{D}\text{-}\mathcal{O}) \in \mathrm{NPMV}$

Proof.
1. Let $\mathcal{X} \in \{\mathrm{A}, \mathrm{S}, \mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}, \mathrm{L}, \mathrm{W}\}$. By definition of multiobjective problems and search notions, $(x, y) \in \mathrm{graph}(\mathcal{X}\text{-}\mathcal{O})$ implies that y is polynomially bounded in its length, and the same holds for the value of y in particular. Further observe that $\mathrm{graph}(\mathcal{X}\text{-}\mathcal{O}) \in \mathrm{coNP}$ and $\mathrm{graph}(\mathrm{Val}(\mathcal{X}\text{-}\mathcal{O})) \in \mathrm{coNP}$ by checking some P-predicate for all possible solutions and hence we obtain $\mathcal{X}\text{-}\mathcal{O} \in \mathrm{coNPMV}$ and $\mathrm{Val}(\mathcal{X}\text{-}\mathcal{O}) \in \mathrm{coNPMV}$.
2. Again, solutions are polynomially bounded. Further observe that $\mathrm{graph}(\mathrm{D}\text{-}\mathcal{O}) \in \mathrm{P}$ and $\mathrm{graph}(\mathrm{Val}(\mathrm{D}\text{-}\mathcal{O})) \in \mathrm{NP}$, because $(\langle x, c\rangle, s) \in \mathrm{graph}(\mathrm{D}\text{-}\mathcal{O}) \iff s \in S^x$ and $y \leftarrow c$, which can be tested in polynomial time, whereas $(\langle x, c\rangle, y) \in \mathrm{graph}(\mathrm{Val}(\mathrm{D}\text{-}\mathcal{O}))$ needs to further check if a solution $s \in S^x$ with $f^x(s) = y$ exists. □

3 Reducibility Structure

We investigate the reducibility among search and value notions for multiobjective problems. More specifically, for every possible combination we either show that reducibility holds for all multiobjective problems (Theorem 3.1, Theorem 3.2) or we give evidence for the existence of a counter example (Theorem 3.3, Corollary 3.5).

Glaßer et al. [GRSW10] show reductions among search notions that generally hold for all multiobjective optimization problems.

Theorem 3.1 ([GRSW10, Theorem 1]). *Let $\mathcal{O} = (S, f, \geq)$ be a k-objective* NP *optimization problem.*

1. $\text{A-}\mathcal{O} \leq_T^p \text{L-}\mathcal{O} \leq_T^p \text{S-}\mathcal{O}$
2. $\text{S-}\mathcal{O} \equiv_T^p \text{D-}\mathcal{O} \equiv_T^p \text{C}_1\text{-}\mathcal{O} \equiv_T^p \text{C}_2\text{-}\mathcal{O} \equiv_T^p \dots \equiv_T^p \text{C}_\text{k}\text{-}\mathcal{O}$
3. $\text{L-}\mathcal{O} \leq_T^p \text{W-}\mathcal{O}$
4. $\text{W-}\mathcal{O} \leq_T^p \text{SAT}$ *and* $\text{D-}\mathcal{O} \leq_T^p \text{SAT}$

Let us first analyze analogous reductions among value notions and relate them to the search notions. After that we will give evidence that these are indeed the only reductions that hold in general.

Theorem 3.2. *Let $\mathcal{O} = (S, f, \geq)$ be a k-objective* NP *optimization problem.*

1. $\text{Val}(\mathcal{X}\text{-}\mathcal{O}) \leq_T^p \mathcal{X}\text{-}\mathcal{O}$ *for* $\mathcal{X} \in \{\text{A}, \text{L}, \text{S}, \text{D}, \text{C}_1, \text{C}_2, \dots, \text{C}_\text{k}, \text{W}\}$
2. $\text{Val}(\text{A-}\mathcal{O}) \leq_T^p \text{Val}(\text{L-}\mathcal{O}) \leq_T^p \text{Val}(\text{S-}\mathcal{O})$
3. $\text{Val}(\text{D-}\mathcal{O}) \equiv_T^p \text{Val}(\text{S-}\mathcal{O}) \equiv_T^p \text{Val}(\text{C}_\text{i}\text{-}\mathcal{O})$ *for* $i \in \{1, \dots, k\}$
4. $\text{Val}(\text{L-}\mathcal{O}) \leq_T^p \text{Val}(\text{W-}\mathcal{O})$

Proof. 1. We can compute a refinement of $\text{Val}(\mathcal{X}\text{-}\mathcal{O})$ by applying f on a refinement of $\mathcal{X}\text{-}\mathcal{O}$.

2. $\text{Val}(\text{L-}\mathcal{O})$ maps to values of optimal solutions, hence every refinement of $\text{Val}(\text{L-}\mathcal{O})$ is a refinement of $\text{Val}(\text{A-}\mathcal{O})$ and we obtain $\text{Val}(\text{A-}\mathcal{O}) \leq_T^p \text{Val}(\text{L-}\mathcal{O})$.

   To show $\text{Val}(\text{L-}\mathcal{O}) \leq_T^p \text{Val}(\text{S-}\mathcal{O})$ we perform a binary search using $\text{Val}(\text{S-}\mathcal{O})$ where we first optimize the objective with the highest priority and go on with the other objectives.

3. It holds that $\text{Val}(\text{D-}\mathcal{O}) \leq_T^p \text{Val}(\text{S-}\mathcal{O})$, because every refinement of $\text{Val}(\text{S-}\mathcal{O})$ is also a refinement of $\text{Val}(\text{D-}\mathcal{O})$. On the other hand we can compute a refinement of $\text{Val}(\text{S-}\mathcal{O})$ by a binary search using any refinement of $\text{Val}(\text{D-}\mathcal{O})$ and hence obtain $\text{Val}(\text{S-}\mathcal{O}) \equiv_T^p \text{Val}(\text{D-}\mathcal{O})$.

   We can compute a refinement of $\text{Val}(\text{D-}\mathcal{O})$ by using any refinement of $\text{Val}(\text{C}_\text{i}\text{-}\mathcal{O})$ with the input cost vector $c = (c_1, \dots, c_k)$ given to $\text{Val}(\text{D-}\mathcal{O})$: If the refinement of $\text{Val}(\text{C}_\text{i}\text{-}\mathcal{O})$ returns a value $y = (y_1, \dots, y_k)$, we return y as a value of the refinement of $\text{Val}(\text{D-}\mathcal{O})$ iff $y_i \geq c_i$. To compute a refinement of $\text{Val}(\text{C}_\text{i}\text{-}\mathcal{O})$ we perform a binary search to optimize i using any partial function that is a refinement of $\text{Val}(\text{S-}\mathcal{O})$ with the constraints as cost vector.

4. Because f is computable in polynomial time, there is a polynomial q with $|f_i^x(s)| \leq q(|x|)$ for every instance $x \in \mathbb{N}$, every solution $s \in S^x$ and every $i \in \{1, \dots, k\}$. Let the order of objectives for $\text{Val}(\text{L-}\mathcal{O})$ be $1, 2, \dots, k$ and define $\omega_i = 2^{(k-i)q(|x|)}$ for $i \in \{1, \dots, k\}$. Then any refinement of $\text{Val}(\text{W-}\mathcal{O})$ with weight vector $\omega = (\omega_1, \dots, \omega_k)$ is a refinement of $\text{Val}(\text{L-}\mathcal{O})$, and we have $\text{Val}(\text{L-}\mathcal{O}) \leq_T^p \text{Val}(\text{W-}\mathcal{O})$. ☐

Theorem 3.3. *If* $\text{P} \neq \text{NP}$, *then there exist two-objective* NP *optimization problems $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ such that:*

1. $\text{Val}(\text{L-}\mathcal{O}_1) \not\leq_T^p \text{A-}\mathcal{O}_1$
2. $\text{Val}(\text{W-}\mathcal{O}_2) \not\leq_T^p \text{D-}\mathcal{O}_2$
3. $\text{Val}(\text{D-}\mathcal{O}_3) \not\leq_T^p \text{W-}\mathcal{O}_3$

Proof. We avoid artificial constructions and use natural optimization problems to show the results.

1. We consider the two-objective minimum lateness and weighted flow time scheduling problem (we assume that objects, such as lists and permutations, are implicitly encoded as non-negative integers)

   2-LWF $= (S, f, \leq)$, where instances are triples (P, D, W) such that
   - $P = (p_1, \dots, p_n) \in \mathbb{N}^n$ are processing times,
   - $D = (d_1, \dots, d_n) \in \mathbb{N}^n$ are due dates,
   - $W = (w_1, \dots, w_n) \in \mathbb{N}^n$ are weights,
   - $S^{(P,D,W)} = \{\pi \mid \pi \text{ is a permutation representing the schedule } p_{\pi(1)}, \dots, p_{\pi(n)}\}$,
   - $f^{(P,D,W)}(\pi) = (L_{\max}, \sum_{j=1}^{n} w_j C_j)$ where
     - the *completion time* of job j is $C_j = \sum_{i:\pi(i)\leq\pi(j)} p_i$,
     - the *maximum lateness* is $L_{\max} = \max\{C_j - d_j \mid 1 \leq j \leq n\}$,
     - the *weighted flow time* is $\sum_{j=1}^{n} w_j C_j$,

   and let $\mathcal{O}_1 =$ 2-LWF. Note that 2-LWF does not strictly conform to the definition of multiobjective optimization problems since f can have negative values. Nonetheless, since f is polynomial-time computable, one can easily construct an equivalent problem where the solutions only have non-negative values by adding an appropriate number.

   Define L$_i$-$\mathcal{O}_1$ for $i = 1, 2$ as the search notion in which the i-th objective has the higher priority. It holds that L$_1$-$\mathcal{O}_1$ is NP-hard and L$_2$-$\mathcal{O}_1$ is polynomial-time solvable [GRSW10]. A-$\mathcal{O}_1$ is polynomial-time solvable as it can be reduced to L$_2$-$\mathcal{O}_1$. We now show L$_1$-$\mathcal{O}_1 \leq_{\mathrm{T}}^{\mathrm{p}}$ Val(L$_1$-$\mathcal{O}_1$):

   Let (P, D, W) with $D = (d_1, \dots, d_n)$ be the input. We query Val(L$_1$-$\mathcal{O}_1$) and obtain $(L_{\max}, \Sigma)$, which is lexicographically optimal over all schedules. Let $D' = (d'_1, \dots, d'_n)$, where $d'_i = d_i + L_{\max}$ (note that $d'_i \geq 0$ even if $L_{\max} < 0$). Observe that for all schedules π we have $f^{(P,D,W)}(\pi) = (x, y) \iff f^{(P,D',W)}(\pi) = (x - L_{\max}, y)$. It hence remains to find a schedule for (P, D', W) with value $(0, \Sigma)$.

   Let $D^* \in \mathbb{N}^n$ with $D^* \leq D'$. As $(0, \Sigma)$ is lexicographically optimal for (P, D', W), for all schedules π the value $f^{(P,D^*,W)}(\pi)$ cannot be lexicographically better than $(0, \Sigma)$. Let d_1^* be the smallest due date for job 1 such that a schedule with value $(0, \Sigma)$ still exists. This existence can be tested by querying Val(L$_1$-$\mathcal{O}_1$), and hence d_1^* can be found in polynomial time by binary search. Fix this due date for job 1 and observe that now, in every schedule with value $(0, \Sigma)$, job 1 has completion time $C_1 = d_1^*$ and hence we know its exact start and completion time in such a schedule. We proceed with the remaining jobs in the same way and hence find a schedule with value $(0, \Sigma)$ in polynomial time. It is easy to see that this schedule has value $(0, \Sigma)$ for the instance (P, D', W) and hence has the value $(L_{\max}, \Sigma)$ for (P, D, W). This shows L$_1$-$\mathcal{O}_1 \leq_{\mathrm{T}}^{\mathrm{p}}$ Val(L$_1$-$\mathcal{O}_1$), hence Val(L$_1$-$\mathcal{O}_1$) is NP-hard.

2. We consider the two-objective minimum quadratic diophantine equations problem

   2-QDE $= (S, f, \leq)$, where instances are $\langle a, b, c\rangle$ with $a, b, c \in \mathbb{N}$,
   $S^{\langle a,b,c\rangle} = \{\langle x, y\rangle \mid ax^2 + by^2 - c \geq 0\}$, and $f^{\langle a,b,c\rangle}(\langle x, y\rangle) = (x^2, y^2)$,

   and let $\mathcal{O}_2 =$ 2-QDE. It holds that D-$\mathcal{O}_2$ is polynomial-time solvable [GRSW10]. The set QDE $= \{(a, b, c) \in \mathbb{N} \mid \exists x, y \in \mathbb{N} : [ax^2 + by^2 - c = 0]\}$ is NP-complete [MA78]. Now we show that $\mathrm{Val}(\text{W-}\mathcal{O}_2)$ is NP-hard by reducing QDE to it. For given $\langle a,b,c\rangle$ solve $\mathrm{Val}(\text{W-}\mathcal{O}_2)$ with the weight vector $w=(a,b)$. If $\mathrm{Val}(\text{W-}\mathcal{O}_2)$ reports that $S^{\langle a,b,c\rangle}=\emptyset$ then $(a,b,c)\notin$ QDE. If otherwise there is some $(x',y')\in \mathrm{Val}(\text{W-}\mathcal{O}_2)(\langle\langle a,b,c\rangle,\langle w\rangle\rangle)$, i.e., there exist $x',y'\in\mathbb{N}$ with $ax'+by'-c\geq 0$ and minimal $ax'+by'$, then $(a,b,c)\in$ QDE if and only if $ax'+by'-c=0$. So it holds that QDE $\leq^{\mathrm{p}}_{\mathrm{T}}$ $\mathrm{Val}(\text{W-}\mathcal{O}_2)$ and therefore $\mathrm{Val}(\text{W-}\mathcal{O}_2)$ is NP-hard.

3. We consider the two-objective minimum spanning tree problem (again, assume that graphs and trees are encoded as non-negative integers)

   2-MST $=(S,f,\leq)$, where instances are $\mathbb{N}^2$-edge-labeled graphs $G=(V,E,l)$,
   $S^G=\{T\subseteq E\mid T \text{ is a spanning tree of } G\}$, and $f^G(T)=\sum_{e\in T} l(e)$,

   and let $\mathcal{O}_3=$ 2-MST. It is known that W-$\mathcal{O}_3$ is polynomial-time solvable, while D-$\mathcal{O}_3$ is NP-hard [GRSW10, PY82]. We show D-$\mathcal{O}_3\leq^{\mathrm{p}}_{\mathrm{T}}\mathrm{Val}(\text{D-}\mathcal{O}_3)$.

   Given an $\mathbb{N}^2$-edge-labeled input graph $G=(V,E,l)$ and a cost vector $c\in\mathbb{N}^2$, suppose there exists a spanning tree that weakly dominates c. Since every spanning tree consists of exactly $|V|-1$ edges, if $|E|>|V|-1$ then there must be some edge that we can delete from the graph such that the resulting graph still contains a spanning tree that weakly dominates c. To find such an edge we loop over all $e\in E$ and ask $\mathrm{Val}(\text{D-}\mathcal{O}_3)$ whether the graph with edges $E\setminus\{e\}$ contains a spanning tree that weakly dominates c. We remove the edge we found and repeat with the altered graph until $|E|=|V|-1$. Clearly, this process terminates after polynomially many iterations and the resulting graph is a spanning tree that weakly dominates c. Hence $\mathrm{Val}(\text{D-}\mathcal{O}_3)$ is NP-hard, and $\mathrm{Val}(\text{D-}\mathcal{O}_3)\not\leq^{\mathrm{p}}_{\mathrm{T}}$ W-$\mathcal{O}_3$, unless P $=$ NP. □

The question of whether A-$\mathcal{O}\leq^{\mathrm{p}}_{\mathrm{T}}\mathrm{Val}(\text{W-}\mathcal{O})$ is related to the study of search versus decision [BD76, Bal89, BBFG91], more precisely to the notion of functional self-reducibility, which was introduced by Borodin and Demers [BD76]. A problem is *functionally self-reducible* if it belongs to the following set (whose name indicates that functional self-reducibility is a universal variant of the notion of search reduces to decision).

$$\mathrm{SRD}_\forall=\{L\in\mathrm{NP}\mid \text{for all polynomials } p \text{ and all } R\in\mathrm{P} \text{ it holds that } (L=\exists_p\cdot R\Rightarrow \mathrm{wit}_p\cdot R\leq^{\mathrm{p}}_{\mathrm{T}} L)\}$$

The statement 1 in the following theorem is equivalent to the statement NP $\neq\mathrm{SRD}_\forall$. Moreover, if there exists an $L\in$ NP for which search does not reduce to decision (as shown by Beigel et al. [BBFG91] under the assumption EE $\neq$ NEE), then statement 1 holds.

Theorem 3.4. *The following statements are equivalent:*

1. *There exists a polynomial p and $R\in\mathrm{P}$ such that* $\mathrm{wit}_p\cdot R\not\leq^{\mathrm{p}}_{\mathrm{T}}\exists_p\cdot R$.

2. *There exists a multiobjective* NP *optimization problem* $\mathcal{O}=(S,f,\geq)$ *such that* A-$\mathcal{O}\not\leq^{\mathrm{p}}_{\mathrm{T}}$ $\mathrm{Val}(\text{W-}\mathcal{O})\equiv^{\mathrm{p}}_{\mathrm{T}}\mathrm{Val}(\text{D-}\mathcal{O})$ *and* $|\mathrm{range}(f)|=1$.

Proof. "1 $\Rightarrow$ 2": Define $\mathcal{O}=(S,f,\geq)$ by $S^x=\mathrm{wit}_p\cdot R(x)$ and $f(\langle x,y\rangle)=1$ for $y\in S^x$. So $(x\in\exists_p\cdot R\iff S^x\neq\emptyset)$ and hence $\exists_p\cdot R\equiv^{\mathrm{p}}_{\mathrm{T}}\mathrm{Val}(\text{W-}\mathcal{O})\equiv^{\mathrm{p}}_{\mathrm{T}}\mathrm{Val}(\text{D-}\mathcal{O})$. The implication follows, since A-$\mathcal{O}=\mathrm{wit}_p\cdot R\not\leq^{\mathrm{p}}_{\mathrm{T}}\exists_p\cdot R$.

"$2 \Rightarrow 1$": From $|\mathrm{range}(f)| = 1$ it follows that each $y \in S^x$ is optimal. Choose a polynomial p such that $y < 2^{p(|x|)}$ for all $y \in S^x$. Let $R = \{\langle x, y\rangle \mid y \in S^x\}$ and note that $R \in \mathrm{P}$ (by the definition of multiobjective problems). Observe that A-$\mathcal{O} = \mathrm{wit}_p\cdot R$. Moreover, $x \in \exists_p\cdot R \iff \mathrm{Val}(\text{W-}\mathcal{O})(\langle x, 0\rangle) \neq \emptyset$ and hence $\exists_p\cdot R \leq^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{W-}\mathcal{O})$. Therefore, $\mathrm{wit}_p\cdot R \not\leq^{\mathrm{p}}_{\mathrm{T}} \exists_p\cdot R$, since otherwise A-$\mathcal{O} \leq^{\mathrm{p}}_{\mathrm{T}} \exists_p\cdot R \leq^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{W-}\mathcal{O})$. □

Corollary 3.5. *If* $\mathrm{P} \neq \mathrm{NP}\cap\mathrm{coNP}$ *or* $\mathrm{EE} \neq \mathrm{NEE}$*, then there exists a multiobjective* NP *optimization problem* $\mathcal{O} = (S, f, \geq)$ *such that* A-$\mathcal{O} \not\leq^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{W-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{D-}\mathcal{O})$.

Proof. Valiant [Val76] shows that $\mathrm{P} \neq \mathrm{NP} \cap \mathrm{coNP}$ implies statement 1 in Theorem 3.4. Beigel et al. [BBFG91] show that $\mathrm{EE} \neq \mathrm{NEE}$ implies the same statement (cf. Theorem 4.10). □

The results of this section are summarized in Figure 2.

4 Complexity of Value Notions

This section addresses the following questions concerning the complexities of value notions Val(A-$\mathcal{O}$), Val(L-$\mathcal{O}$), Val(D-$\mathcal{O}$), and Val(W-$\mathcal{O}$).

Q1: What complexities can appear?

Q2: What settings of complexities for Val(A-$\mathcal{O}$), Val(L-$\mathcal{O}$), Val(D-$\mathcal{O}$), and Val(W-$\mathcal{O}$) are possible for fixed multiobjective problems $\mathcal{O}$?

It turns out that Val(L-$\mathcal{O}$), Val(D-$\mathcal{O}$), and Val(W-$\mathcal{O}$) can be embedded in NP, while we give evidence that this does not hold for Val(A-$\mathcal{O}$). Moreover, NP can be embedded in Val(A-$\mathcal{O}$), Val(L-$\mathcal{O}$), Val(D-$\mathcal{O}$), and Val(W-$\mathcal{O}$), which answers Q1. Regarding Q2, we show that the following settings of complexities are possible: For all sets $A, L, D, W \in \mathrm{NP}$ that satisfy the following moderate requirements there exist multiobjective NP optimization problems $\mathcal{O}$ whose value notions are equivalent to A, L, D, W.

- Requirement 1: $A \leq^{\mathrm{p}}_{\mathrm{T}} L \leq^{\mathrm{p}}_{\mathrm{T}} D$ and $L \leq^{\mathrm{p}}_{\mathrm{T}} W$
- Requirement 2: $W \equiv^{\mathrm{p}}_{\mathrm{T}} g$ for some $g \in \max\cdot D$

The first requirement is necessary, since by Theorem 3.2 these reducibilities hold for all multiobjective NP optimization problems. The necessity of the second requirement is shown by Proposition 4.2.

Theorem 4.1. *Let* A*,* L*,* D*,* $W \in \mathrm{NP}$ *such that* $A \leq^{\mathrm{p}}_{\mathrm{T}} L \leq^{\mathrm{p}}_{\mathrm{T}} D$ *and* $L \leq^{\mathrm{p}}_{\mathrm{T}} W \equiv^{\mathrm{p}}_{\mathrm{T}} g$ *for some* $g \in \max\cdot D$*. Then there exists a two-objective* NP *optimization problem* $\mathcal{O} = (S, f, \geq)$ *such that*

1. Val(A-$\mathcal{O}$) $\equiv^{\mathrm{p}}_{\mathrm{T}} A$

2. Val(L-$\mathcal{O}$) $\equiv^{\mathrm{p}}_{\mathrm{T}} L$

3. Val(D-$\mathcal{O}$) $\equiv^{\mathrm{p}}_{\mathrm{T}} D$

4. Val(W-$\mathcal{O}$) $\equiv^{\mathrm{p}}_{\mathrm{T}} W$

Proof. Let A, L, D, $W \in \mathrm{NP}$, $g \in \max \cdot D$ with reduction relations as required in the statement of the theorem and let A_w, L_w, $D_w \in \mathrm{P}$ be corresponding witness sets. For the order of objectives with regard to Val(L-$\mathcal{O}$) we choose to give priority to the first objective.

We first show that we can demand the following without loss of generality:

Claim 4.1.1. *By replacing all sets and functions in the theorem with equivalent sets and functions, it can be assumed that p is a polynomial such that for any x the following holds:*

1. *for any $L \in \{A_w, L_w, D_w\}$ and any y such that $\langle x, y\rangle \in L$ it holds that $y < 2^{p(|x|)}$*
2. *for all y where $\langle x, y\rangle \in D$ it holds that $0 < y < 2^{p(|x|)} - 1$ and $g = \max_p \cdot D$*
3. *there is at least one y such that $\langle x, y\rangle \in D$*
4. *for all y it holds that $\langle x, y\rangle \in D \iff \langle x, 2^{p(|x|)} - 1 - y\rangle \in D$*

Proof. Statement 1 can be fulfilled by using a large enough polynomial and removing witnesses from the witness set that are too large. Note that 1 remains fulfilled for larger polynomials.

For an arbitrary $D_0 \in \mathrm{NP}$ and $g_0 = \max_{p_0} \cdot D_0$ (with $p_0 > 0$), we now construct $D \equiv^{\mathrm{p}}_{\mathrm{T}} D_0$ and $g = \max_p \cdot D \equiv^{\mathrm{p}}_{\mathrm{T}} g_0$ for some polynomial p that fulfill the assertions. Consider the set

$$D' \overset{df}{=} \{\langle\langle x, 0\rangle, y\rangle \mid \langle x, y\rangle \in D_0 \text{ and } y < 2^{p_0(|x|)}\} \cup \{\langle\langle x, 1 + y\rangle, a\rangle \mid a = 1 \vee (a = 0 \wedge \langle x, y\rangle \in D_0)\}.$$

Observe that $D_0 \equiv^{\mathrm{p}}_{\mathrm{T}} D'$, $g_0 \equiv^{\mathrm{p}}_{\mathrm{T}} g' \overset{df}{=} \max_{p_0} \cdot D'$ and for all $\langle x, y\rangle \in D'$ it holds that $y < 2^{p_0(|x|)}$. Choose some polynomial p such that $p > p_0 + 3$ and p is large enough for assertion 1. Observe that for

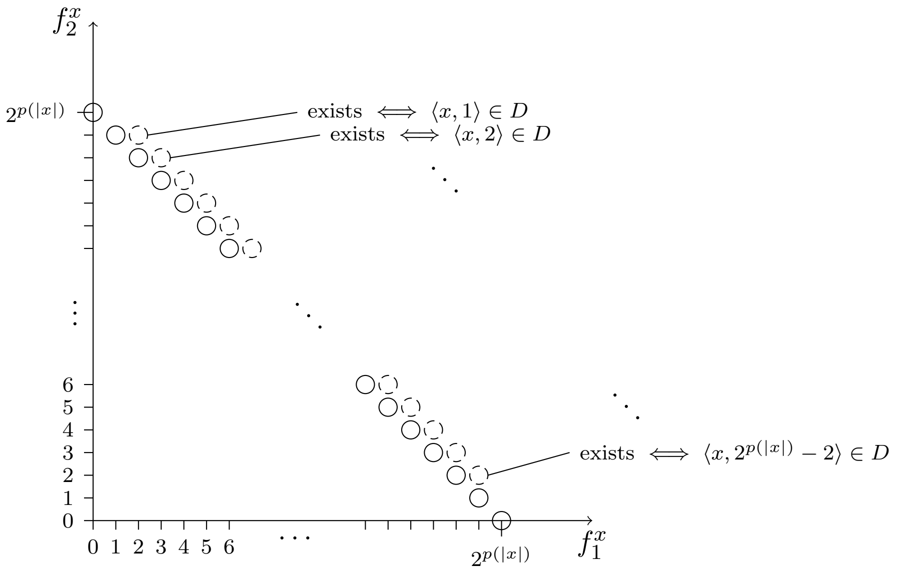
$$\begin{aligned} D \overset{df}{=} & \{\langle x, 2^{p(|x|)-1} + y\rangle \mid \langle x, y\rangle \in D'\} \cup \\ & \{\langle x, 2^{p(|x|)-1} - 1\rangle \mid x \in \mathbb{N}\} \cup \\ & \{\langle x, 2^{p(|x|)-1}\rangle \mid x \in \mathbb{N}\} \cup \\ & \{\langle x, 2^{p(|x|)-1} - 1 - y\rangle \mid \langle x, y\rangle \in D'\} \end{aligned}$$

it holds that $D \equiv^{\mathrm{p}}_{\mathrm{T}} D'$ and $g \overset{df}{=} \max_p \cdot D \equiv^{\mathrm{p}}_{\mathrm{T}} g_0$. Moreover, D and g fulfill the remaining assertions. □

We define the 2-objective maximization problem $\mathcal{O} = (S, f, \geq)$ by

$$\begin{aligned} S^{3x} &= \{\langle 0, 0, y\rangle \mid \langle x, y\rangle \in A_w\} && \text{(stage for } A) \\ S^{3x+1} &= \{\langle 0, 0, 0\rangle\} \cup \{\langle 0, 1, y\rangle \mid \langle x, y\rangle \in L_w\} && \text{(stage for } L) \\ S^{3x+2} &= \{\langle 0, i, 0\rangle \mid i \leq 2^{p(|x|)}\} \cup && \text{(stage for } W \text{ and } D) \\ & \quad \{\langle 1, y, z\rangle \mid y < 2^{p(|x|)} \text{ and } \langle\langle x, y\rangle, z\rangle \in D_w\} \\ f^{3x+r}(\langle a, i, z\rangle) &= (i + a, j) \text{ such that } i + j = 2^{p(|x|)} \end{aligned}$$

The lengths of valid solutions are obviously polynomially bounded and S is in P, because $\langle a, i, z\rangle \in S^{3x+r}$ can always be checked by simple arithmetic and optionally some query to a witness set in P. The objective function f is computable in polynomial time.



f_2^x
$2^{p(|x|)}$
exists $\iff \langle x,1\rangle \in D$
exists $\iff \langle x,2\rangle \in D$
6
5
4
3
2
1
0
exists $\iff \langle x, 2^{p(|x|)}-2\rangle \in D$
0 1 2 3 4 5 6
$2^{p(|x|)}$
f_1^x

Figure 4: Illustration of $f(S^{3x+2})$.

1. $\mathrm{Val}(\text{A-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} A$: Note that the value $(0, 2^{p(|x|)})$ is always optimal for instances of the form $3x+1$ or $3x+2$, so the reduction algorithm can output it without querying A. For instances of the form $3x$ it queries A for x and outputs $(0, 2^{p(|x|)})$ if the answer is yes and $\perp$ otherwise.

   $A \leq_{\mathrm{T}}^{\mathrm{p}} \mathrm{Val}(\text{A-}\mathcal{O})$: Here, on input x the reduction is done by a query for $\mathrm{Val}(\text{A-}\mathcal{O})(3x)$ with output "no" if and only if the answer is $\perp$.

2. $\mathrm{Val}(\text{L-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} L$: Note that for instances of the form $3x+2$, the values $(0, 2^{p(|x|)})$ and $(2^{p(|x|)}, 0)$ are always optimal, so the reduction algorithm can output a lexicographically optimal solution without querying L. Instances of the form $3x$ can be solved by a query to $\mathrm{Val}(\text{A-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} A \leq_{\mathrm{T}}^{\mathrm{p}} L$. Let now the instance be $3x+1$. Note that $\mathrm{Val}(\text{L-}\mathcal{O})$ has to output $(0, 2^{p(|x|)})$ if $x \notin L$ and $(1, 2^{p(|x|)} - 1)$ otherwise, which can be checked by a simple query to L.

   $L \leq_{\mathrm{T}}^{\mathrm{p}} \mathrm{Val}(\text{L-}\mathcal{O})$: Similar to the case for A, the reduction is a simple query to $\mathrm{Val}(\text{L-}\mathcal{O})(3x+1)$.

3. $\mathrm{Val}(\text{D-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} D$: Instances not of the form $3x+2$ can be handled by queries to $\mathrm{Val}(\text{A-}\mathcal{O})$ or $\mathrm{Val}(\text{L-}\mathcal{O})$ since $\mathrm{Val}(\text{A-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} A \leq_{\mathrm{T}}^{\mathrm{p}} D$ and $\mathrm{Val}(\text{L-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} L \leq_{\mathrm{T}}^{\mathrm{p}} D$. Let now $\langle 3x+2, \langle i,j\rangle\rangle$ be the input.

   If $i + j \leq 2^{p(|x|)}$, output $f^{3x+2}(\langle 0, i, 0\rangle) = (i, 2^{p(|x|)} - i)$, which is always the value of some solution. If $i+j > 2^{p(|x|)} + 1$, there is no solution that (weakly) dominates this value, so output $\perp$. For the last case, $i + j = 2^{p(|x|)} + 1$, note that the only solutions that can possibly (weakly) dominate the value (i,j) are those of type $\langle 1, y, z\rangle$ for $y < 2^{p(|x|)}$ and $\langle\langle x,y\rangle, z\rangle \in D_w$ which also have the value (i,j). This means that $y = i-1$, so we can return (i,j) if $\langle x, i-1\rangle \in D$ and $\perp$ otherwise.

   $D \leq_{\mathrm{T}}^{\mathrm{p}} \mathrm{Val}(\text{D-}\mathcal{O})$: On input $\langle x,y\rangle$, $\mathrm{Val}(\text{D-}\mathcal{O})(\langle 3x+2, \langle i,j\rangle\rangle)$ with $i = y+1$ and $j = 2^{p(|x|)} - y$ is queried. As shown in the previous paragraph, the result of this query tells whether or not $\langle x,y\rangle \in D$.

4. Val(W-$\mathcal{O}$) $\leq^{\mathrm{p}}_{\mathrm{T}}$ W: As in the case of Val(D-$\mathcal{O}$), instances not of the form $3x+2$ can be handled by indirect reductions. For instances of the form $3x+2$ we show Val(W-$\mathcal{O}$) $\leq^{\mathrm{p}}_{\mathrm{T}}$ g:

   It obviously suffices to return values from the border of the convex hull of all solution values. It even suffices to consider only corner points of the convex hull. These corner points are $(0, 2^{p(|x|)}), (2^{p(|x|)}, 0), (1 + y_{\min}, 2^{p(|x|)} - y_{\min})$ and $(1 + y_{\max}, 2^{p(|x|)} - y_{\max})$ where $y_{\min}$ and $y_{\max}$ are the minimal and maximal values for y such that $\langle x, y\rangle \in D$. Since we required that $\langle x, y\rangle \in D \iff \langle x, 2^{p(|x|)} - 1 - y\rangle \in D$, we only need to determine $y_{\max}$ and this can obviously be done by a query to $g(x)$ (note that we also required that there is at least one y such that $\langle x, y\rangle \in D$).

5. $W \leq^{\mathrm{p}}_{\mathrm{T}}$ Val(W-$\mathcal{O}$): The reduction $g \leq^{\mathrm{p}}_{\mathrm{T}}$ Val(W-$\mathcal{O}$) holds as follows: On input x, Val(W-$\mathcal{O}$)$(\langle 3x+2, \langle w, w-1\rangle\rangle)$ for $w = 2^{p(|x|)} + 1$ is queried. The weighted sum of the value of a solution $s = \langle a, i, z\rangle$ is

$$\begin{aligned} w f_1^{3x+2}(\langle a, i, z\rangle) + (w-1) f_2^{3x+2}(\langle a, i, z\rangle) &= w(i+a) + (w-1)(2^{p(|x|)} - i) \\ &= i + wa + (w-1)2^{p(|x|)}. \end{aligned}$$

   Since every possible value for i is at most $2^{p(|x|)} < w$ and we required that there is at least one y such that $\langle x, y\rangle \in D$, the function Val(W-$\mathcal{O}$) returns the value of a solution of type $\langle 1, y, z\rangle$ with maximal y, which is exactly $g(x)$. □

We now show that in Theorem 4.1 it is necessary to restrict the relationship between D and W such that $W \equiv^{\mathrm{p}}_{\mathrm{T}} g$ for some $g \in \max \cdot D$. As a consequence, the complexities for Val(A-$\mathcal{O}$), Val(L-$\mathcal{O}$), Val(D-$\mathcal{O}$), and Val(W-$\mathcal{O}$) provided by Theorem 4.1 are indeed all possible complexities for the value notions that can be described in terms of sets (cf. Corollary 4.3).

Proposition 4.2. *For every multiobjective* NP *optimization problem* $\mathcal{O} = (S, f, \geq)$ *there is some* $A \in \mathrm{NP}$ *and* $g \in \max \cdot A$ *such that*

$$\mathrm{Val}(\text{D-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} A \text{ and } \mathrm{Val}(\text{W-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} g.$$

Proof. For the k-objective problem $\mathcal{O} = (S, f, \geq)$ let

$$\begin{aligned} A := \{\langle\langle x, \langle w\rangle\rangle, \langle z, y_k, \ldots, y_1\rangle'\rangle \mid w \in \mathbb{N}^k, y_i < 2^{p(|x|)}, z = \sum_{i=1}^{k} w_i y_i, \text{ and} \\ \text{there is some } s \in S^x \text{ such that } f^x(s) \geq (y_1, \ldots, y_k)\} \end{aligned}$$

where p is a polynomial upper bound for all polynomials in the definition of $\mathcal{O}$ and $\langle z, y_k, \ldots, y_1\rangle' \overset{df}{=} 1 + z \cdot 2^{k \cdot p(|x|)} + \sum_{i=1}^{k} y_i \cdot 2^{(i-1)\cdot p(|x|)}$ for $z \in \mathbb{N}$ and $0 \leq y_i < 2^{p(|x|)}$. This means that $\langle\cdot\rangle'$ is a bijection between $\mathbb{N} \times \{0, \ldots, 2^{p(|x|)} - 1\}^k$ and $\mathbb{N}^+$ that transfers the lexicographical order on $\mathbb{N} \times \{0, \ldots, 2^{p(|x|)} - 1\}^k$ to the natural order on $\mathbb{N}^+$. Furthermore, for all $\langle\langle x, \langle w\rangle\rangle, \langle z, y_k, \ldots, y_1\rangle'\rangle \in A$ it holds that $\langle z, y_k, \ldots, y_1\rangle' < 2^{q(|\langle x, \langle w\rangle\rangle|)}$ for some polynomial q. Since $\{\langle x, s\rangle \mid x \in \mathbb{N}, s \in S^x\} \in \mathrm{P}$ and $f \in \mathrm{PF}$ we have $A \in \mathrm{NP}$. Let $g = \max_q \cdot A$. We will show Val(D-$\mathcal{O}$) $\equiv^{\mathrm{p}}_{\mathrm{T}}$ A and Val(W-$\mathcal{O}$) $\equiv^{\mathrm{p}}_{\mathrm{T}}$ g.

1. $\mathrm{Val}(\text{D-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} A$: On input $\langle x, \langle c \rangle \rangle$, we query $x' := \langle \langle x, \langle 0, 0, \dots, 0 \rangle \rangle, \langle 0, c_k, \dots, c_1 \rangle' \rangle \in A$. If $x' \notin A$, then there is no $s \in S^x$ with $f^x(s) \geq (c_1, \dots, c_k)$, and we return $\perp$. Otherwise there is some $s \in S^x_{\mathrm{opt}}$ with $f^x(s) = (c'_1, \dots, c'_k) \geq (c_1, \dots, c_k)$. We find $(c'_1, \dots, c'_k)$ by a binary search using queries similar to x' and return $(c'_1, \dots, c'_k)$.

2. $A \leq_{\mathrm{T}}^{\mathrm{p}} \mathrm{Val}(\text{D-}\mathcal{O})$: On input $\langle \langle x, \langle w \rangle \rangle, \langle z, y_k, \dots, y_1 \rangle' \rangle$, we reject if $z \neq \sum_{i=1}^k w_i y_i$. Otherwise we accept if and only if there is some $s \in S^x$ with $f^x(s) \geq (y_1, \dots, y_k)$, which can be determined by a query to $\mathrm{Val}(\text{D-}\mathcal{O})$ on $\langle x, \langle y_1, \dots, y_k \rangle \rangle$.

3. $\mathrm{Val}(\text{W-}\mathcal{O}) \leq_{\mathrm{T}}^{\mathrm{p}} g$: On input $\langle x, \langle w_1, \dots, w_k \rangle \rangle$, we obtain $r := g(\langle x, \langle w_1, \dots, w_k \rangle \rangle)$ by a query to the oracle. If $r = 0$, there are no $z, y_1, \dots, y_k \in \mathbb{N}$ with $\langle \langle x, \langle w_1, \dots, w_k \rangle \rangle, \langle z, y_k, \dots, y_1 \rangle' \rangle \in A$, and thus $S^x = \emptyset$ and we return $\perp$. Otherwise, let $z, y_1, \dots, y_k \in \mathbb{N}$ with $\langle z, y_k, \dots, y_1 \rangle' = r$. Hence we have $z = \sum_{i=1}^k w_i y_i$ and $f^x(s) \geq (y_1, \dots, y_k)$ for some $s \in S^x$.

   Assume there is some $s' \in S^x_{\mathrm{opt}}$ such that $z' \overset{\mathit{df}}{=} \sum_{i=1}^k w_i f_i^x(s') > \sum_{i=1}^k w_i f_i^x(s) \geq \sum_{i=1}^k w_i y_i$. Then $\langle z', f_k^x(s'), \dots, f_1^x(s') \rangle' > \langle z, y_k, \dots, y_1 \rangle' = r$ because of the lexicographic ordering induced by $\langle \cdot \rangle'$ and thus r is not maximal, which is a contradiction.

   It remains to show that $(y_1, \dots, y_k)$ is the value of some solution. Let s be the previously mentioned solution and assume that $f_i^x(s) > y_i$ for some i. Let $z' \overset{\mathit{df}}{=} \sum_{i=1}^k w_i f_i^x(s)$. If $z' > z$, then $\langle z', f_k^x(s), \dots, f_1^x(s) \rangle' > r$, which is impossible. Otherwise $z' = z$ (and $w_i = 0$) and hence $\langle z', f_k^x(s), \dots, f_1^x(s) \rangle' > r$, which is impossible again. Thus we have $f^x(s) = (y_1, \dots, y_k)$, which is a valid answer for the input.

4. $g \leq_{\mathrm{T}}^{\mathrm{p}} \mathrm{Val}(\text{W-}\mathcal{O})$: On input $\langle x, \langle w_1, \dots, w_k \rangle \rangle$, let $\tilde{w_i} := w_i \cdot 2^{k \cdot p(|x|)} + 2^{(i-1) \cdot p(|x|)}$ for all i and query $\mathrm{Val}(\text{W-}\mathcal{O})$ on $\langle x, \langle \tilde{w_1}, \dots, \tilde{w_k} \rangle \rangle$. On answer $\perp$ we have $S^x = \emptyset$ and return 0, which is obviously the correct value. Otherwise, if $(y_1, \dots, y_k)$ is the obtained answer, let the reduction function return $1 + \sum_{i=1}^k \tilde{w_i} y_i = 1 + \sum_{i=1}^k w_i y_i 2^{k \cdot p(|x|)} + \sum_{i=1}^k y_i 2^{(i-1) \cdot p(|x|)} = \langle z, y_k, \dots, y_1 \rangle'$ for $z = \sum_{i=1}^k w_i y_i$. Because we got $(y_1, \dots, y_k)$ from a query to $\mathrm{Val}(\text{W-}\mathcal{O})$, there is some $s \in S^x$ such that $f^x(s) \geq (y_1, \dots, y_k)$ and thus, the returned value is in $\mathrm{wit}_q \cdot A(\langle x, \langle w_1, \dots, w_k \rangle \rangle)$. To see that it is indeed maximal, assume there is some $\langle z', y'_1, \dots, y'_k \rangle' \in \mathrm{wit}_q \cdot A(\langle x, \langle w_1, \dots, w_k \rangle \rangle)$ that is strictly larger. Here we get

$$\begin{aligned}\sum_{i=1}^k \tilde{w_i} y'_i &= \sum_{i=1}^k w_i y'_i 2^{k \cdot p(|x|)} + \sum_{i=1}^k y'_i 2^{(i-1) \cdot p(|x|)} \\ &= \langle z', y'_k, \dots, y'_1 \rangle' - 1 > \langle z, y_k, \dots, y_1 \rangle' - 1 = \sum_{i=1}^k \tilde{w_i} y_i,\end{aligned}$$

   which contradicts the fact that $\mathrm{Val}(\text{W-}\mathcal{O})$ returns a value that is optimal with respect to the sum weighted by $(\tilde{w_1}, \dots, \tilde{w_k})$. $\square$

Corollary 4.3. *Let $A, L, D, W \in \mathrm{NP}$. The following statements are equivalent:*

1. *There exists a multiobjective* NP *optimization problem $\mathcal{O} = (S^x, f, \geq)$ such that*
$$\begin{aligned} A &\equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{A-}\mathcal{O}), \\ L &\equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{L-}\mathcal{O}), \\ D &\equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{D-}\mathcal{O}), \\ W &\equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{W-}\mathcal{O}). \end{aligned}$$

2. *$A \leq^{\mathrm{p}}_{\mathrm{T}} L \leq^{\mathrm{p}}_{\mathrm{T}} D, W$ and W is $\leq^{\mathrm{p}}_{\mathrm{T}}$-equivalent to some function in $\max \cdot D'$ for some $D' \in \mathrm{NP}$ such that $D' \equiv^{\mathrm{p}}_{\mathrm{T}} D$.*

Proof. "2 $\Rightarrow$ 1" follows from Theorem 4.1 applied to A, L, D', W and "1 $\Rightarrow$ 2" follows from Proposition 4.2 and Theorem 3.2. ☐

Corollary 4.4. *If $A, L, W \in \mathrm{NP}$ such that $A \leq^{\mathrm{p}}_{\mathrm{T}} L \leq^{\mathrm{p}}_{\mathrm{T}} W$, then there exists a multiobjective* NP *optimization problem $\mathcal{O}$ such that $A \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{A-}\mathcal{O})$, $L \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{L-}\mathcal{O})$, and $W \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{W-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{D-}\mathcal{O})$.*

Proof. Let $D = D' = \{\langle x, 1\rangle \mid x \in W\}$ and $p(n) = 1$. Note that $D, D' \in \mathrm{NP}$, $D' \equiv^{\mathrm{p}}_{\mathrm{T}} D \equiv^{\mathrm{p}}_{\mathrm{T}} W$, and $\max_p \cdot D' \equiv^{\mathrm{p}}_{\mathrm{T}} W$. So we can apply Corollary 4.3, which finishes the proof. ☐

From the results in this section it follows that $\mathrm{Val}(\text{L-}\mathcal{O})$, $\mathrm{Val}(\text{D-}\mathcal{O})$, and $\mathrm{Val}(\text{W-}\mathcal{O})$ are always equivalent to sets in NP, which is probably not true for $\mathrm{Val}(\text{A-}\mathcal{O})$.

Corollary 4.5. *For every multiobjective* NP *optimization problem $\mathcal{O}$ the following holds.*

1. $\mathrm{Val}(\text{L-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} B$ *for some $B \in \mathrm{NP}$.*
2. $\mathrm{Val}(\text{D-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} B$ *for some $B \in \mathrm{NP}$.*
3. $\mathrm{Val}(\text{W-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} B$ *for some $B \in \mathrm{NP}$.*

Proof. 1. Let $1, 2, \dots, k$ be the order of objectives for $\mathrm{Val}(\text{L-}\mathcal{O})$. For the k-objective problem $\mathcal{O} = (S, f, \leftarrow)$, let p be a polynomial upper bound for all values of f. Let

$$\begin{aligned} B = \{\langle x, \langle y_1, \dots, y_k\rangle\rangle \mid x, y_1, \dots, y_k \in \mathbb{N} \text{ and there is some } s \in S^x \text{ such that } f_1(s) \leftarrow_1 y_1 \\ \wedge f_1(s) = y_1 \implies (f_2(s) \leftarrow_2 y_2 \\ \wedge f_2(s) = y_2 \implies (f_3(s) \leftarrow_3 y_3 \\ \cdots \\ \wedge f_{k-1}(s) = y_{k-1} \implies f_k(s) \leftarrow_k y_k \dots))\} \end{aligned}$$

and observe that $B \in \mathrm{NP}$. We have $\mathrm{Val}(\text{L-}\mathcal{O}) \leq^{\mathrm{p}}_{\mathrm{T}} B$ by a binary search over k stages: suppose $(y_1^*, \dots, y_k^*) \in \mathrm{Val}(\text{L-}\mathcal{O})(x)$. In the i-th stage of the binary search, we ask queries of the form $\langle x, \langle y_1^*, \dots, y_{i-1}^*, y_i, z_{i+1}, \dots, z_k\rangle\rangle \in B$, where $z_j = 0$ if the j-th objective is maximized, and $z_j = 2^{p(|x|)}$ otherwise. This way we find y_i^* in polynomial time. On the other hand, given the value of $\mathrm{Val}(\text{L-}\mathcal{O})(x)$, it is easy to determine whether or not $\langle x, \langle y_1, \dots, y_k\rangle\rangle \in B$, hence we also have $B \leq^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{L-}\mathcal{O})$.

2. Follows from Proposition 4.2.

3. By Proposition 4.2, there exists a $g \in \max \cdot \mathrm{NP}$ such that $\mathrm{Val}(\text{W-}\mathcal{O}) \equiv^{\mathrm{p}}_{\mathrm{T}} g$. By Proposition 2.3, $g \equiv^{\mathrm{p}}_{\mathrm{T}} B$ for some $B \in \mathrm{NP}$. $\square$

The absence of $\mathrm{Val}(\text{A-}\mathcal{O})$ in Corollary 4.5 can be explained: Below we show that each function in $\mathrm{wit}\cdot \mathrm{P}$ is equivalent to some $\mathrm{Val}(\text{A-}\mathcal{O})$ (we will later show the stronger statement that each function in $\mathrm{wit}\cdot \mathrm{P}$ is equivalent to some $\text{A-}\mathcal{O}$ (Proposition 5.2) and each $\text{A-}\mathcal{O}$ is equivalent to some $\mathrm{Val}(\text{A-}\mathcal{O}')$ (Proposition 5.8)). Then in Corollary 4.8 we give evidence for the existence of functions in $\mathrm{wit}\cdot \mathrm{P}$ that are inequivalent to all sets. Hence this is an evidence for the existence of multiobjective NP optimization problems whose arbitrary optimum search and value notions are inequivalent to all sets.

Proposition 4.6. *For every $g \in \mathrm{wit}\cdot \mathrm{P}$ there is some two-objective* NP *optimization problem $\mathcal{O}$ such that $g \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{A-}\mathcal{O})$.*

Proof. Let $g = \mathrm{wit}_p\cdot R$ for some polynomial p and $R \in \mathrm{P}$. Define $\mathcal{O} = (S, f, \geq)$ such that $S^x = g(x)$ and $f^x(s) = (s, 2^{p(|x|)} - s)$ for all $s \in S^x$ and observe that $g(x) \equiv^{\mathrm{p}}_{\mathrm{T}} \mathrm{Val}(\text{A-}\mathcal{O})$. $\square$

Theorem 4.7. *Let $t, m\colon \mathbb{N} \to \mathbb{N}$ such that $t(i) = 2^{2^{2^{2^i}}}$ and $m(i) = 2^i$.*
Let $f \in \mathrm{wit}\cdot \mathrm{P}$ such that $\mathrm{supp}(f) \subseteq \{t(i) + k \mid i \in \mathbb{N}, 0 \leq k < m(i)\}$ and $f \equiv^{\mathrm{p}}_{\mathrm{T}} A$ for some $A \subseteq \mathbb{N}$.

1. $\mathrm{supp}(f) \in \mathrm{FewP}$*.*

2. If $\mathrm{supp}(f) = \{t(i) + k \mid i \in \mathbb{N}, 0 \leq k < m(i)\}$ *then* $A \in \mathrm{UP} \cap \mathrm{coUP}$*.*

Proof. We begin with the first statement. Since $f \leq^{\mathrm{p}}_{\mathrm{T}} A$, there is some partial function $g\colon \mathbb{N} \to \mathbb{N}$ that is a refinement of f such that $g \leq^{\mathrm{p}}_{\mathrm{T}} A$. Furthermore, since $A \leq^{\mathrm{p}}_{\mathrm{T}} f$, we have $g \leq^{\mathrm{p}}_{\mathrm{T}} f$ via some polynomial-time oracle Turing machine M. In order to simplify notation, we define for any $r \geq 0$ and any $q = \langle q_1, \ldots, q_r\rangle$ and $a = \langle a_1, \ldots, a_r\rangle$ the multivalued function $O^r_{q,a}$ such that $O^r_{q,a}(x) = \{a_i \mid x = q_i \text{ for some } 1 \leq i \leq r\}$. Let now

$$\begin{aligned} W := \{\langle t(i)+k, \langle r, q, a\rangle\rangle \mid 0 \leq k < m(i), q = \langle q_1, \ldots, q_r\rangle \text{ such that}& \\ q_1 < q_2 < \cdots < q_r \text{ and } \{q_1, \ldots, q_r\} \subseteq \{t(j) + k' \mid j \leq i, k' < m(j)\},& \\ \forall 1 \leq s \leq r\colon M^{O^r_{q,a}}(q_s) \in O^r_{q,a}(q_s) \subseteq f(q_s),& \\ t(i) + k \in \{q_1, \ldots, q_r\}\}& \end{aligned}$$

We show that $W \in \mathrm{P}$ and $\mathrm{supp}(f) \in \exists\cdot W$.

$W \in \mathrm{P}$: The only nontrivial parts are checking that $O^r_{q,a}(q_s) \subseteq f(q_s)$ and $M^{O^r_{q,a}}(q_s) \in O^r_{q,a}(q_s)$ for all $1 \leq s \leq r$. The former can be done in polynomial time since $f \in \mathrm{wit}\cdot \mathrm{P}$ and the latter by a simulation of the polynomial-time oracle Turing machine M.

$\mathrm{supp}(f) \in \exists\cdot W$: We first show that there is a polynomial p such that for $\langle t(i) + k, \langle r, q, a\rangle\rangle \in W$ it holds that $\langle r, q, a\rangle < 2^{p(|t(i)+k|)}$, or $|\langle r, q, a\rangle| \leq p(|t(i) + k|)$. For some $c \in \mathbb{N}$, we have an obvious bound of

$$|\langle r,q,a\rangle| \le c\sum_{j=0}^{i} m(j)|t(j)+m(j)|^c \le c\sum_{j=0}^{i}|2\,t(j)|^{c+1} \le c\sum_{j=0}^{i}(2+2^{2^{2^j}})^{c+1}$$
$$\le c\sum_{j=0}^{i} 2^{(c+2)2^{2^j}} \le c\sum_{j=0}^{(c+2)2^{2^i}} 2^j \le c\cdot 2^{1+(c+2)2^{2^i}},$$

which is polynomial in $2^{2^{2^i}}$ and thus in $|t(i)+k|$.

For $\mathrm{supp}(f) \subseteq \exists_p\cdot W$, let $x = t(i)+k \in \mathrm{supp}(f)$. Let $q_1 < \cdots < q_r$ such that $\{q_1,\ldots,q_r\} = \mathrm{supp}(f)\cap\{0,1,\ldots,t(i)+m(i)-1\}$ and define $q = \langle q_1,\ldots,q_r\rangle$ and $a = \langle g(q_1),\ldots,g(q_r)\rangle$. Remember that $g \le_{\mathrm{T}}^{\mathrm{p}} f$ via M. On input $t(i)+m(i)-1$ (or smaller), there is some $c\in\mathbb{N}$ such that the largest number M can query is at most

$$2^{|t(i)+m(i)-1|^c} \le 2^{|2\,t(i)|^c} \le 2^{|t(i)|^{2c}} \le 2^{\left(2^{2^i}\right)^{2c}} \le 2^{2^{2c\,2^i}}.$$

For large enough i it holds that

$$2^{2^{2c\,2^i}} < 2^{2^{\left(2^{2^i}\right)^2}} = 2^{2^{2^{2^{i+1}}}} = t(i+1).$$

By encoding oracle answers into the program, we can assume that M only queries the oracle for inputs with i large enough for the above inequality to hold and thus $M^{O^r_{q,a}}(x) = M^f(x) = g(x) \in O^r_{q,a}(x)$ for all $x\in\{q_1,\ldots,q_r\}$. This shows that $\langle t(i)+k,\langle r,q,a\rangle\rangle \in W$.

In order to show $\exists_p\cdot W \subseteq \mathrm{supp}(f)$ let $\langle t(i)+k,\langle r,q,a\rangle\rangle\in W$ and $\langle q_1,\ldots,q_r\rangle = q$. Since $t(i)+k\in\{q_1,\ldots,q_r\}$ and $O^r_{q,a}(q_s)\subseteq f(q_s)$ for all $s\in\{1,\ldots,r\}$, it especially holds that $f(t(i)+k)\neq\emptyset$ and thus $t(i)+k\in\mathrm{supp}(f)$.

Let us now count the number of witnesses for each $t(i)+k\in\exists_p\cdot W$. Note that for a fixed set $\{q_1,\ldots,q_r\}$, the values in a are uniquely determined by the simulations $M^{O^r_{q,a}}$. Thus the number of witnesses $w(i)$ is at most the number of subsets of $\mathrm{supp}(f)\cap\{0,1,\ldots,t(i)+m(i)-1\}$, i.e.,

$$w(i) \le 2^{\sum_{j=0}^{i} m(j)} = 2^{2^{i+1}-1} \le \left(2^{2^i}\right)^2 \le |t(i)|^2,$$

which is polynomial in $|t(i)+k|$ and thus $\mathrm{supp}(f) = \exists_p\cdot W \in \mathrm{FewP}$.

For the second statement, note that it now holds that $\mathrm{supp}(f) = \{t(i)+k \mid i\in\mathbb{N}, 0\le k< m(i)\}$. We describe a $(\mathrm{UP}\cap\mathrm{coUP})$-Machine M' that accepts A. On input x, let $t(i)+k$ be the largest number in $\mathrm{supp}(f)$ that can possibly be queried in the reduction $A\le_{\mathrm{T}}^{\mathrm{p}} f$ on input length $|x|$. Observe that for $r=\sum_{j=0}^{i} m(j)$ there is exactly one pair (q,a) such that $\langle t(i)+k,\langle r,q,a\rangle\rangle\in W$. This means that if M' searches nondeterministically for a pair (q',a') such that $\langle t(i)+k,\langle r,q',a'\rangle\rangle\in W$, there is exactly one path that finds such a pair and it holds that $(q',a')=(q,a)$. Following that, M' can simulate the reduction $A\le_{\mathrm{T}}^{\mathrm{p}} f$, since $O^r_{q,a}$ is a “refinement” of f restricted to the part of $\mathrm{supp}(f)$ that can possibly be queried in the reduction. After this simulation, there is a single path of M' that has the information of whether or not $x\in A$ and thus $A\in\mathrm{UP}\cap\mathrm{coUP}$. □

The following corollary shows that under reasonable assumptions there are multivalued functions that are inequivalent to any set. Note that a multivalued function f is equivalent to a set if and only if the set of partial functions that are refinements of f has a minimal element with respect to the partial order $\leq_{\mathrm{T}}^{\mathrm{p}}$. In other words, a multivalued function f is not equivalent to any set if and only if no partial function that is a refinement of f is reducible (and thus equivalent) to f.

Corollary 4.8.

1. *If* $\mathrm{FewEEE} \neq \mathrm{NEEE}$*, then there exists an* $f \in \mathrm{wit}\cdot\mathrm{P}$ *such that* $f \not\equiv_{\mathrm{T}}^{\mathrm{p}} A$ *for all* $A \subseteq \mathbb{N}$*.*
2. *If* $\mathrm{UEEE} \cap \mathrm{coUEEE} \neq \mathrm{NEEE} \cap \mathrm{coNEEE}$*, then there exists an* $f \in \mathrm{wit}\cdot\mathrm{P}$ *such that* $f \not\equiv_{\mathrm{T}}^{\mathrm{p}} A$ *for all* $A \subseteq \mathbb{N}$*.*

Proof. 1. Proposition 2.5.4 provides a $B \in \mathrm{NP} - \mathrm{FewP}$ such that $B \subseteq \{t(c\cdot i)+k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$ for some $c \geq 1$ and $t(n) = 2^{2^{2^{2^n}}}$. Choose a polynomial p and $R \in \mathrm{P}$ such that $B = \exists_p\cdot R$. Let $f = \mathrm{wit}_p\cdot R$ and note that $\mathrm{supp}(f) = B \subseteq \{t(c\cdot i)+k \mid i \in \mathbb{N}, 0 \leq k < 2^i\} \subseteq \{t(i)+k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$. By Theorem 4.7.1, if $f \equiv_{\mathrm{T}}^{\mathrm{p}} A$ for some $A \subseteq \mathbb{N}$, then $B = \mathrm{supp}(f) \in \mathrm{FewP}$, which is a contradiction.

2. Proposition 2.5.5 provides a $B \in (\mathrm{NP} \cap \mathrm{coNP}) - (\mathrm{UP} - \mathrm{coUP})$ such that $B \subseteq \{t(c \cdot i) + k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$ for some $c \geq 1$ and $t(n) = 2^{2^{2^{2^n}}}$. Choose a polynomial p and $R, R' \in \mathrm{P}$ such that $B = \exists_p\cdot R$ and $\overline{B} = \exists_p\cdot R'$. Let $S = \{\langle t(i) + k, y\rangle \in R \cup R' \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$ and note that $S \in \mathrm{P}$. Let $f = \mathrm{wit}_p\cdot S$. Observe that $\mathrm{supp}(f) = \exists_p\cdot S = \{t(i) + k \mid i \in \mathbb{N}, 0 \leq k < 2^i\}$. By Theorem 4.7.2, if $f \equiv_{\mathrm{T}}^{\mathrm{p}} A$ for some $A \subseteq \mathbb{N}$, then $A \in \mathrm{UP} \cap \mathrm{coUP}$ and hence $B \in \mathrm{UP} \cap \mathrm{coUP}$ (since $B \leq_{\mathrm{T}}^{\mathrm{p}} f \leq_{\mathrm{T}}^{\mathrm{p}} A$). The latter is a contradiction. □

In Corollary 4.3 we characterized the compositions of sets $A, L, D, W \in \mathrm{NP}$ for which there exist problems $\mathcal{O}$ with search notions equivalent to A, L, D, W. Besides the trivial requirements $A \leq_{\mathrm{T}}^{\mathrm{p}} L \leq_{\mathrm{T}}^{\mathrm{p}} D$ and $L \leq_{\mathrm{T}}^{\mathrm{p}} W$ (they hold for all problems by Theorem 3.2) there is one additional:

$$W \equiv_{\mathrm{T}}^{\mathrm{p}} g \text{ for some } g \in \max\cdot D \tag{4}$$

Observe that every set $X \in \mathrm{NP}$ is equivalent to some function $g \in \max\cdot Y$ for some $Y \equiv_{\mathrm{T}}^{\mathrm{p}} \mathrm{SAT}$ (define $Y = \{\langle x, 3 + c_X(x)\rangle \mid x \in \mathbb{N}\} \cup \{\langle x, 1 + c_{\mathrm{SAT}}(x)\rangle \mid x \in \mathbb{N}\}$). So for a problem $\mathcal{O}$ where Val(D-$\mathcal{O}$) is NP-hard, the complexity of Val(W-$\mathcal{O}$) can be arbitrary. The easier Val(D-$\mathcal{O}$) gets, the more restrictions are imposed on the complexity for Val(W-$\mathcal{O}$). However, this does not mean that Val(W-$\mathcal{O}$) needs to have lower complexity, since Val(W-$\mathcal{O}$) can be NP-hard while Val(D-$\mathcal{O}$) is polynomial-time solvable (take, for example, D as a witness set for SAT).

We now further investigate the particular situation where Val(D-$\mathcal{O}$) is polynomial-time solvable. Here, Val(W-$\mathcal{O}$) must be equivalent to some function in $\max\cdot\mathrm{P}$. Does this really restrict the complexity of Val(W-$\mathcal{O}$)? Using a technique by Beigel at al. [BBFG91] we give evidence for the existence of sets in NP that are not equivalent to functions from $\mathrm{wit}\cdot\mathrm{P}$ (resp., $\max\cdot\mathrm{P}$). More precisely, under the assumption $\mathrm{EE} \neq \mathrm{NEE}$ there exist very sparse sets in $X \in \mathrm{NP} - \mathrm{P}$ and we show that such sets cannot be equivalent to functions in $\mathrm{wit}\cdot\mathrm{P}$. It follows that there is no multiobjective NP optimization problem $\mathcal{O}$ such that Val(W-$\mathcal{O}$) $\equiv_{\mathrm{T}}^{\mathrm{p}} X$, while Val(A-$\mathcal{O}$), Val(L-$\mathcal{O}$), and Val(D-$\mathcal{O}$) are polynomial-time solvable. This is an evidence that the requirement (4) is indeed a restriction.

Lemma 4.9. *If $A \notin \mathrm{P}$ and $A \subseteq \{2^{2^{x^c}} \mid x \in \mathbb{N}\}$ where $c \geq 1$, then $A \not\equiv^{\mathrm{p}}_{\mathrm{T}} f$ for all $f \in \mathrm{wit}\cdot\mathrm{P}$.*

Proof. Assume there exists an $f \in \mathrm{wit}\cdot\mathrm{P}$ such that $A \equiv^{\mathrm{p}}_{\mathrm{T}} f$. So $f \leq^{\mathrm{p}}_{\mathrm{T}} A$ via an oracle Turing machine M whose running time is bounded by some polynomial q. On inputs of length n, the machine M cannot ask queries longer than $q(n)$. In particular, it cannot query $y = 2^{2^{x^c}}$ where $x = \lceil \log q(n) \rceil$, since $|y| > 2^{x^c} \geq 2^x \geq q(n)$.

Therefore, for inputs of length n, we can replace M's oracle A by the characteristic sequence

$$a_n \;=\; \chi_A(2^{2^{0^c}})\,\chi_A(2^{2^{1^c}})\,\chi_A(2^{2^{2^c}})\;\cdots\;\chi_A(2^{2^{\lfloor \log q(n)\rfloor^c}}).$$

Since $|a_n| = 1 + \lfloor \log q(n) \rfloor \leq 1 + \log q(n)$, there are at most $2^{1+\log q(n)} = 2q(n)$ sequences of length $|a_n|$. So on input y where $n = |y|$ we can simulate in polynomial time the computation of M on y for all characteristic sequences of length $|a_n|$. If $f(y) \neq \emptyset$, then at least one simulation returns a value from $f(y)$. Moreover, we can verify the correctness of these values in polynomial time, since $\mathrm{graph}(f) \in \mathrm{P}$. This shows that f has a refinement in PF and hence $A \in \mathrm{P}$, which is a contradiction. □

Theorem 4.10.

1. *If $\mathrm{EE} \neq \mathrm{NEE}$, then there exists a $B \in \mathrm{NP}$ such that $B \not\equiv^{\mathrm{p}}_{\mathrm{T}} f$ for all $f \in \mathrm{wit}\cdot\mathrm{P}$.*
2. *If NP has P-bi-immune sets, then there exists a $B \in \mathrm{NP}$ such that $B \not\equiv^{\mathrm{p}}_{\mathrm{T}} f$ for all $f \in \mathrm{wit}\cdot\mathrm{P}$.*

Proof. 1. Proposition 2.5 provides a $B \in \mathrm{NP} - \mathrm{P}$ such that $B \subseteq \{2^{2^{x^c}} \mid x \in \mathbb{N}\}$ where $c \geq 1$. Now apply Lemma 4.9. 2. Choose a P-bi-immune $L \in \mathrm{NP}$ and let $B = L \cap \{2^{2^x} \mid x \in \mathbb{N}\}$. From the P-bi-immunity of L it follows that $B \notin \mathrm{P}$. Now apply Lemma 4.9. □

Corollary 4.11.

1. *If $\mathrm{EE} \neq \mathrm{NEE}$, then there exists an $B \in \mathrm{NP}$ such that $B \not\equiv^{\mathrm{p}}_{\mathrm{T}} g$ for all $g \in \max\cdot\mathrm{P}$.*
2. *If NP has P-bi-immune sets, then there exists an $B \in \mathrm{NP}$ such that $B \not\equiv^{\mathrm{p}}_{\mathrm{T}} g$ for all $g \in \max\cdot\mathrm{P}$.*

Proof. Let B be the set provided by Theorem 4.10. It suffices to show that for every $g \in \max\cdot\mathrm{P}$ there exists some $f \in \mathrm{wit}\cdot\mathrm{P}$ such that $g \equiv^{\mathrm{p}}_{\mathrm{T}} f$. Let $g \in \max\cdot\mathrm{P}$ and choose $R' \in \mathrm{P}$ and a polynomial p such that $g = \max_p \cdot R'$. The set $R = \{\langle\langle x, z\rangle, y\rangle \mid 1 \leq z \leq y < 2^{p(|x|)} \text{ and } \langle x, y\rangle \in R'\}$ is in P. Let $f = \mathrm{wit}_p\cdot R$ and observe $f \equiv^{\mathrm{p}}_{\mathrm{T}} g$. □

5 Complexity of Search Notions

As opposed to the value notions from the previous section, the complexities of search notions A-$\mathcal{O}$, L-$\mathcal{O}$, D-$\mathcal{O}$, and W-$\mathcal{O}$ do not cover all problems in NP, unless NEE = coNEE. However, the complexities of L-$\mathcal{O}$, D-$\mathcal{O}$, and W-$\mathcal{O}$ exactly coincide with the complexities of wit$\cdot$P-functions. This does not hold for the complexities of A-$\mathcal{O}$, unless $\mathrm{EE} = \mathrm{NEE} \cap \mathrm{coNEE}$. They cover at least all problems in $\mathrm{NP} \cap \mathrm{coNP}$, but it remains a task for further research to exactly determine these complexities.

Theorem 5.1. *Let $k \geq 1$ and h be a multivalued function. The following statements are equivalent:*

1. *There is some $g \in \mathrm{wit}\cdot \mathrm{P}$ such that $h \equiv^{\mathrm{p}}_{\mathrm{T}} g$.*
2. *There is some k-objective* NP *optimization problem $\mathcal{O} = (S, f, \geq)$ such that $h \equiv^{\mathrm{p}}_{\mathrm{T}}$ L-$\mathcal{O}$.*
3. *There is some k-objective* NP *optimization problem $\mathcal{O} = (S, f, \geq)$ such that $h \equiv^{\mathrm{p}}_{\mathrm{T}}$ D-$\mathcal{O}$.*
4. *There is some k-objective* NP *optimization problem $\mathcal{O} = (S, f, \geq)$ such that $h \equiv^{\mathrm{p}}_{\mathrm{T}}$ W-$\mathcal{O}$.*

Proof. "1 ⇒ 2, 3, 4": Define the k-objective problem $\mathcal{O} = (S, f, \geq)$ with $S^x = g(x)$ and $f^x(s) = (0, 0, \ldots, 0)$ for all $s \in S^x$. It holds that D-$\mathcal{O} \equiv^{\mathrm{p}}_{\mathrm{T}}$ W-$\mathcal{O} \equiv^{\mathrm{p}}_{\mathrm{T}}$ L-$\mathcal{O} = g \equiv^{\mathrm{p}}_{\mathrm{T}} h$.

"2 ⇒ 1": Let $\mathcal{O} = (S, f, \geq)$ be a k-objective problem such that $h \equiv^{\mathrm{p}}_{\mathrm{T}}$ L-$\mathcal{O}$. We assume that the order of objectives for L-$\mathcal{O}$ is $1, 2, \ldots, k$. Let $X = \{\langle\langle x, c_1, \ldots, c_k\rangle, s\rangle \mid s \in S^x, c_1, \ldots, c_k \in \mathbb{N}$, there is some $1 \leq i_0 \leq k+1$ such that $f^x_i(s) = c_i$ for all $i < i_0$ and (if $i_0 \leq k$) $f^x_{i_0}(s) > c_{i_0}\} \in \mathrm{P}$ and $g = \mathrm{wit}_p\cdot X$ for a large enough polynomial p.
$g \leq^{\mathrm{p}}_{\mathrm{T}}$ L-$\mathcal{O}$: On input $\langle x, c_1, \ldots, c_k\rangle$ we query L-$\mathcal{O}(x)$. If the answer is $\perp$, we return $\perp$, since in this case $S^x = \emptyset$. Otherwise, let the answer be $s \in S^x$. If there is some $1 \leq i_0 \leq k+1$ such that $f^x_i(s) = c_i$ for all $i < i_0$ and $f^x_{i_0}(s) > c_{i_0}$, return s, otherwise return $\perp$. We have to show that the reduction is correct if this i_0 does not exist. In this case, there is some $1 \leq j_0 \leq k$ such that $f^x_i(s) = c_i$ for all $i < j_0$ and $f^x_{j_0}(s) < c_{j_0}$. Assume our answer is incorrect. Then there is some $s' \in S^x$ such that $f^x_i(s') = f^x_i(s) = c_i$ for all $i < j_0$ and $f^x_{j_0}(s') \geq c_{j_0} > f^x_{j_0}(s)$. This contradicts the optimality of s with respect to the j_0-th objective.
L-$\mathcal{O} \leq^{\mathrm{p}}_{\mathrm{T}} g$: Start with the constraint vector $(c_1, c_2, \ldots, c_k) = (0, 0, \ldots, 0)$ and successively determine the highest value for each constraint using binary search (leaving the constraints with lower index at their highest value and setting the constraints with higher index to zero). The obtained solution is lexicographically optimal.

"3 ⇒ 1": Let $\mathcal{O} = (S, f, \geq)$ be a k-objective problem such that $h \equiv^{\mathrm{p}}_{\mathrm{T}}$ D-$\mathcal{O}$. Define $X = \{\langle\langle x, \langle c\rangle\rangle, y\rangle \mid y \in S^x, c \in \mathbb{N}^k, f^x(y) \geq c\} \in \mathrm{P}$ and note that D-$\mathcal{O} \in \mathrm{wit}\cdot X \subseteq \mathrm{wit}\cdot \mathrm{P}$.

"4 ⇒ 3": Note that by Proposition 2.7, W-$\mathcal{O}$ = A-$\mathcal{O}'$ for some single-objective problem $\mathcal{O}'$ and A-$\mathcal{O}' \equiv^{\mathrm{p}}_{\mathrm{T}}$ D-$\mathcal{O}'$ since $\mathcal{O}'$ is a single-objective problem. □

The search notion A-$\mathcal{O}$ is missing in Theorem 5.1. Here we show that each function in $\mathrm{wit}\cdot\mathrm{P}$ is equivalent to (even equals) some A-$\mathcal{O}$ and we provide evidence against the converse (Corollary 5.5).

Proposition 5.2. *For every $k \geq 1$ and every function $g \in \mathrm{wit}\cdot \mathrm{P}$ there is some k-objective* NP *optimization problem $\mathcal{O}$ such that g = A-$\mathcal{O}$.*

Proof. Define the k-objective problem $\mathcal{O} = (S, f, \geq)$ with $S^x = g(x)$ and $f^x(s) = (0, 0, \ldots, 0)$ for all $s \in S^x$ and observe that $g(x)$ = A-$\mathcal{O}$. □

The proposition raises the question of whether every A-$\mathcal{O}$ is equivalent to some function in $\mathrm{wit}\cdot\mathrm{P}$. We show that the answer is *no*, unless EE = NEE ∩ coNEE. For this purpose, we first prove that the complexities of the A-$\mathcal{O}$ cover at least all problems in NP ∩ coNP.

Theorem 5.3. *For every $L \in \mathrm{NP} \cap \mathrm{coNP}$ there is a two-objective* NP *optimization problem $\mathcal{O}$ such that* A-$\mathcal{O} \equiv_{\mathrm{T}}^{\mathrm{p}} L$.

Proof. Let $L \in \mathrm{NP} \cap \mathrm{coNP}$. Hence there are witness sets $L_1, L_2 \in \mathrm{P}$ and a polynomial p such that $L = \exists_p \cdot L_1$ and $\overline{L} = \exists_p \cdot L_2$, which means that

$$\begin{aligned} x \in L &\iff \exists y \text{ with } y < 2^{p(|x|)} \text{ and } \langle x, y\rangle \in L_1 \\ x \notin L &\iff \exists y \text{ with } y < 2^{p(|x|)} \text{ and } \langle x, y\rangle \in L_2 \end{aligned}$$

for all $x \in \mathbb{N}$. Note that L_1 and L_2 are disjoint. Let $\mathcal{O} = (S, f, \leq)$, where $S^x = \mathrm{wit}_p \cdot L_1(x) \cup \mathrm{wit}_p \cdot L_2(x) \cup \{2^{p(|x|)}, 2^{p(|x|)} + 1\}$ and

$$f^x(y) = \begin{cases} (1,0) & \text{if } y < 2^{p(|x|)} \text{ and } \langle x, y\rangle \in L_1 \\ (2,0) & \text{if } y = 2^{p(|x|)} \\ (0,1) & \text{if } y < 2^{p(|x|)} \text{ and } \langle x, y\rangle \in L_2 \\ (0,2) & \text{if } y = 2^{p(|x|)} + 1 \end{cases}$$

for all $x \in \mathbb{N}$ and $y \in S^x$. Observe that $\mathcal{O}$ is a 2-objective NP optimization problem. We have the following reductions.

1. $L \leq_{\mathrm{T}}^{\mathrm{p}}$ A-$\mathcal{O}$: For all $x \in \mathbb{N}$ we have

$$\begin{aligned} x \in L &\iff \exists y \text{ with } y < 2^{p(|x|)} \text{ and } \langle x, y\rangle \in L_1 \text{ and} \\ &\qquad \forall y' \text{ with } y' < 2^{p(|x|)} \text{ we have } \langle x, y'\rangle \notin L_2 \\ &\iff \text{A-}\mathcal{O}(x) = \mathrm{wit}_p \cdot L_2(x) \cup \{2^{p(|x|)} + 1\} \end{aligned}$$

   and $x \notin L \iff$ A-$\mathcal{O}(x) = \mathrm{wit}_p \cdot L_2(x) \cup \{2^{p(|x|)}\}$ analogously. If we get an arbitrary element from A-$\mathcal{O}(x)$ we can distinguish the two cases in polynomial time and thus $L \leq_{\mathrm{T}}^{\mathrm{p}}$ A-$\mathcal{O}$.

2. A-$\mathcal{O} \leq_{\mathrm{T}}^{\mathrm{p}} L$: For $x \in \mathbb{N}$, observe that $\{2^{p(|x|)}, 2^{p(|x|)} + 1\} \subseteq S^x$. We will argue that one of those solutions is optimal and, furthermore, this solution can be determined by a single query to L. For that purpose, observe that if $x \in L$, then for all $y < 2^{p(|x|)}$ we have $\langle x, y\rangle \notin L_2$, hence there is no y whose value dominates $(0,2)$, and we can return $y = 2^{p(|x|)} + 1$ as solution for A-$\mathcal{O}(x)$. On the other hand, if $x \notin L$, then for all $y < 2^{p(|x|)}$ we have $\langle x, y\rangle \notin L_1$, hence there is no y whose value dominates $(2,0)$, and we can return $y = 2^{p(|x|)}$ as solution for A-$\mathcal{O}(x)$. In all cases we compute a refinement of A-$\mathcal{O}$ and thus have A-$\mathcal{O} \leq_{\mathrm{T}}^{\mathrm{p}} L$ as claimed. □

Theorem 5.4.

1. *If* $\mathrm{EE} \neq \mathrm{NEE} \cap \mathrm{coNEE}$, *then there exists a* $B \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$ *such that* $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} f$ *for all* $f \in \mathrm{wit} \cdot \mathrm{P}$.

2. *If* $\mathrm{NP} \cap \mathrm{coNP}$ *has* P*-bi-immune sets, then there exists a* $B \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$ *such that* $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} f$ *for all* $f \in \mathrm{wit} \cdot \mathrm{P}$.

Proof. 1. Proposition 2.5 provides a $B \in (\mathrm{NP} \cap \mathrm{coNP}) - \mathrm{P}$ such that $B \subseteq \{2^{2^{x^c}} \mid x \in \mathbb{N}\}$ for some $c \geq 1$. Now apply Lemma 4.9. 2. Choose a P-bi-immune $L \in \mathrm{NP} \cap \mathrm{coNP}$ and let $B = L \cap \{2^{2^x} \mid x \in \mathbb{N}\}$. From the P-bi-immunity of L it follows that $B \notin \mathrm{P}$. Now apply Lemma 4.9. □

Corollary 5.5.

1. *If* EE $\neq$ NEE $\cap$ coNEE*, then there exists a two-objective* NP *optimization problem* $\mathcal{O}$ *such that* A-$\mathcal{O} \not\equiv^{\mathrm{p}}_{\mathrm{T}} f$ *for all* $f \in \mathrm{wit}\cdot\mathrm{P}$.

2. *If* NP $\cap$ coNP *has* P*-bi-immune sets, then there exists a two-objective* NP *optimization problem* $\mathcal{O}$ *such that* A-$\mathcal{O} \not\equiv^{\mathrm{p}}_{\mathrm{T}} f$ *for all* $f \in \mathrm{wit}\cdot\mathrm{P}$.

Proof. Let B be the set provided by Theorem 5.4. By $B \in \mathrm{NP} \cap \mathrm{coNP}$ and Theorem 5.3, there exists a 2-objective NP optimization problem $\mathcal{O}$ such that A-$\mathcal{O} \equiv^{\mathrm{p}}_{\mathrm{T}} B$. □

The Theorems 5.1 and 5.3 raise the following questions: Is every set in NP equivalent to some A-$\mathcal{O}$ (resp., L-$\mathcal{O}$, D-$\mathcal{O}$, W-$\mathcal{O}$)? With Theorem 5.7 we show that the answer is *no*, unless NEE = coNEE. There we use the following idea by Beigel et al. [BBFG91]: If NEE $\neq$ coNEE, then NP $-$ coNP contains very sparse sets. Such sets cannot be equivalent to some A-$\mathcal{O}$ and hence (by Lemma 4.9) they cannot be equivalent to functions in wit$\cdot$P.

Lemma 5.6. *If* $B \notin \mathrm{coNP}$ *and* $B \subseteq \{2^{2^{z^c}} \mid z \in \mathbb{N}\}$ *where* $c \geq 1$*, then* $B \not\equiv^{\mathrm{p}}_{\mathrm{T}}$ A-$\mathcal{O}$ *for all multiobjective* NP *optimization problems* $\mathcal{O}$.

Proof. Assume there exists a k-objective NP optimization problem $\mathcal{O} = (S, f, \leftarrow)$ such that $B \equiv^{\mathrm{p}}_{\mathrm{T}}$ A-$\mathcal{O}$. Without loss of generality we may assume that A-$\mathcal{O}(x) \neq \emptyset$ for all x. Choose a polynomial p and oracle Turing machines M_1 and M_2 such that $B \leq^{\mathrm{p}}_{\mathrm{T}}$ A-$\mathcal{O}$ via M_1, A-$\mathcal{O} \leq^{\mathrm{p}}_{\mathrm{T}} B$ via M_2, and the running times of M_1 and M_2 are bounded by p.

Let M be the following polynomial-time oracle Turing machine: M on input x simulates the computation of M_1 on x such that each inquiry q to the oracle is replaced by the computation M_2 on q (where the queries caused by M_2 on q are passed to M's oracle). Note that $B \leq^{\mathrm{p}}_{\mathrm{T}} B$ via M and hence $L(M^B) = B$.

Consider M on input of some x of length n. The queries q generated by the simulation of M_1 on x cannot be longer than $p(n)$. Each such q causes a computation of M_2 on q whose running time is bounded by $p(|q|) \leq p(p(n))$. Therefore, all oracle queries asked by M on x are of length at most $p(p(n))$. In particular, M on x cannot query $q = 2^{2^{z^c}}$ where $z = \lceil \log p(p(n)) \rceil$, since $|q| > 2^{z^c} \geq 2^z \geq p(p(n))$. So for inputs of length at most n, we can replace M's oracle B by the characteristic sequence

$$a_n \;=\; \chi_B(2^{2^{0^c}})\,\chi_B(2^{2^{1^c}})\,\chi_B(2^{2^{2^c}})\;\cdots\;\chi_B(2^{2^{\lfloor \log p(p(n)) \rfloor^c}}).$$

For $w \in \{0,1\}^*$, let $M^w(x)$ denote the computation of M on x, where M's oracle is replaced by w (i.e., M interprets w as the characteristic sequence a_n and answers oracle queries accordingly). Since $|a_n| = 1 + \lfloor \log p(p(n)) \rfloor \leq 1 + \log p(p(n))$, there are at most $2^{1+\log p(p(n))} = 2p(p(n))$ sequences of length $|a_n|$. Therefore, on input x where $n = |x|$ we can simulate in polynomial time the computations $M^w(x)$ for all $w \in \{0,1\}^*$ of length $|a_n|$.

Recall that during the computation $M^w(x)$ (more precisely in the simulation of M_1 on x), each query q is replaced by the computation M_2 on q, which in turn computes an answer to the query q. We combine all these queries q and their answers a in the following set.

$$Q^w(x) = \{(q,a) \mid M^w(x) \text{ simulates } M_2 \text{ on } q \text{ and this simulation results in the answer } a\}$$

Let $W_n = \{w \in \{0,1\}^* \mid |w| = |a_n|\}$. We claim that for all x where $n = |x|$ it holds that

$$x \in B \iff \exists w \in W_n[M^w(x) = 1 \ \wedge\ \forall (q,a) \in Q^w(x)\,[a \in \text{A-}\mathcal{O}(q)]]. \tag{5}$$

Assume $x \in B$. Let $w = a_n$ and note that $M^w(x) = 1$, since $B \leq_{\mathrm{T}}^{\mathrm{p}} B$ via M and $M^B(x) = M^{a_n}(x)$. Let $(q,a) \in Q^w(x)$, i.e., $M_2^w(q)$ returns a. From $M_2^w(q) = M_2^{a_n}(q) = M_2^B(q)$ and $\text{A-}\mathcal{O} \leq_{\mathrm{T}}^{\mathrm{p}} B$ via M_2 it follows that $a \in \text{A-}\mathcal{O}$.

Assume that the right-hand side of (5) holds. In particular, $a \in \text{A-}\mathcal{O}(q)$ for all $(q,a) \in Q^w(x)$. Therefore, $M^w(x)$ correctly simulates $M_1^{\text{A-}\mathcal{O}}$ on x, since all queries q are answered appropriately, i.e., according to a partial function that is a refinement of A-$\mathcal{O}$. Hence $M_1^{\text{A-}\mathcal{O}}(x) = M^w(x) = 1$. From $B \leq_{\mathrm{T}}^{\mathrm{p}} \text{A-}\mathcal{O}$ via M_1 it follows that $x \in B$. This proves the equivalence (5).

If we negate both sides of (5), we obtain the following for all x where $n = |x|$.

$$x \in \overline{B} \iff \forall w \in W_n[M^w(x) \neq 1 \ \vee\ \exists (q,a) \in Q^w(x)\,[a \notin \text{A-}\mathcal{O}(q)] \tag{6}$$

Recall that $|W_n| \leq 2p(p(n))$. Moreover, for all $w \in W_n$ it holds that $|Q^w(x)| \leq p(n)$, since the running time of M_1 on x is bounded by $p(n)$. So the ranges of both quantifiers at the right-hand side of (6) have polynomial size. Hence, in order to verify $x \in \overline{B}$, we have to check only a polynomial number of conditions of the form $[a \notin \text{A-}\mathcal{O}(q)]$. The latter can be tested in nondeterministic polynomial time, since

$$a \notin \text{A-}\mathcal{O}(q) \iff a \notin S^q \vee \exists b \in S^q \text{ such that } b \text{ dominates } a.$$

This shows that the right-hand side of (6) can be tested in nondeterministic polynomial time. Therefore, $\overline{B} \in \mathrm{NP}$ and hence $B \in \mathrm{coNP}$. This contradicts the assumption. □

Theorem 5.7. *If* NEE $\neq$ coNEE*, then there exists a* $B \in \mathrm{NP} - \mathrm{coNP}$ *such that for every multiobjective* NP *optimization problem* $\mathcal{O} = (S, f, \geq)$ *it holds that* $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} \text{A-}\mathcal{O}$, $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} \text{L-}\mathcal{O}$, $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} \text{D-}\mathcal{O}$, *and* $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} \text{W-}\mathcal{O}$.

Proof. Proposition 2.5 provides a $B \in \mathrm{NP} - \mathrm{coNP}$ such that $B \subseteq \{2^{2^{x^c}} \mid x \in \mathbb{N}\}$ for some $c \geq 1$. By Lemma 5.6, $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} \text{A-}\mathcal{O}$ for all multiobjective NP optimization problems $\mathcal{O}$. Moreover, by Lemma 4.9, $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} f$ for all $f \in \mathrm{wit}\cdot\mathrm{P}$. This implies the theorem, since by Theorem 5.1, the search notions L-$\mathcal{O}$, D-$\mathcal{O}$, and W-$\mathcal{O}$ are equivalent to some function in wit$\cdot$P. □

The proof shows that if we drop the condition $B \not\equiv_{\mathrm{T}}^{\mathrm{p}} \text{A-}\mathcal{O}$, then the theorem can be shown under the weaker assumption EE $\neq$ NEE (Theorem 4.10).

We complete this section by showing that the complexities of the search notion A-$\mathcal{O}$ are covered by the complexities of the value notions Val(A-$\mathcal{O}'$).

Proposition 5.8. *For every multiobjective* NP *optimization problem* $\mathcal{O} = (S, f, \geq)$ *there is a multiobjective* NP *optimization problem* $\mathcal{O}' = (S, g, \geq)$ *such that* $\text{A-}\mathcal{O} = \text{A-}\mathcal{O}' \equiv^{\text{p}}_{\text{T}} \text{Val}(\text{A-}\mathcal{O}')$.

Proof. Let $\mathcal{O} = (S, f, \geq)$ be a k-objective problem and assume $k \geq 2$ (use the same objective function twice for $k = 1$). Let p be a polynomial such that for all x and all $s \in S^x$ it holds that $s < 2^{p(|x|)}$ and $f^x_i(s) < 2^{p(|x|)}$ for all $1 \leq i \leq k$. Define the k-objective problem $\mathcal{O}' = (S, g, \geq)$ where

$$g^x_i(s) = f^x_i(s)\, k\, 2^{3\,p(|x|)} + \sum_{j=1}^{k} f^x_j(s)\, 2^{p(|x|)} + \begin{cases} 2^{p(|x|)} - 1 - s & \text{for } i = 1 \\ s & \text{for } i \geq 2. \end{cases}$$

Claim 5.8.1. *The following statements are equivalent for all* $x \in \mathbb{N}$ *and* $s_1, s_2 \in S^x$*:*

1. $f^x(s_1) \neq f^x(s_2)$ *and* $f^x(s_1) \leq f^x(s_2)$
2. $g^x(s_1) \neq g^x(s_2)$ *and* $g^x(s_1) \leq g^x(s_2)$

Proof. "1 $\Rightarrow$ 2": Assume $f^x(s_1) \neq f^x(s_2)$ and $f^x(s_1) \leq f^x(s_2)$ and let $1 \leq j \leq k$ such that $f^x_j(s_1) < f^x_j(s_2)$. Since f_j occurs in each g_i with a factor of at least $2^{p(|x|)}$ and $s_1, s_2, 2^{p(|x|)} - 1 - s_1, 2^{p(|x|)} - 1 - s_2 < 2^{p(|x|)}$, we have $g^x_i(s_1) < g^x_i(s_2)$ for each i.

"2 $\Rightarrow$ 1": Assume $g^x(s_1) \neq g^x(s_2)$ and $g^x(s_1) \leq g^x(s_2)$. It is not possible that $f^x(s_1) = f^x(s_2)$, since in this case, $0 \neq s_1 - s_2 = g^x_1(s_2) - g^x_1(s_1) = -(g^x_2(s_2) - g^x_2(s_1))$, which contradicts the fact that $g^x(s_1) \leq g^x(s_2)$. Hence we have $f^x(s_1) \neq f^x(s_2)$. Finally, assume that there is some $1 \leq j \leq k$ such that $f^x_j(s_1) > f^x_j(s_2)$. Then we would also have $g^x_j(s_1) > g^x_j(s_2)$ because of the large factor $k\, 2^{3\,p(|x|)}$. □

From the claim it follows that a solution is not optimal in $\mathcal{O}$ if and only if it is not optimal in $\mathcal{O}'$ and thus the set of optimal solutions coincide, i.e. $\text{A-}\mathcal{O} = \text{A-}\mathcal{O}'$. Furthermore, since the solution is encoded into the value for $\mathcal{O}'$, we obtain $\text{A-}\mathcal{O}' \equiv^{\text{p}}_{\text{T}} \text{Val}(\text{A-}\mathcal{O}')$. □

References

[Bal89] J. L. Balcázar. Self-reducibility structures and solutions of NP problems. *Revista Matematica de la Universidad Complutense de Madrid*, 2(2-3):175–184, 1989.

[BBFG91] R. Beigel, M. Bellare, J. Feigenbaum, and S. Goldwasser. Languages that are easier than their proofs. In *IEEE Symposium on Foundations of Computer Science*, pages 19–28, 1991.

[BD76] A. B. Borodin and A. J. Demers. Some comments on functional self-reducibility and the NP hierarchy. Technical Report TR76-284, Cornell University, Department of Computer Science, 1976.

[BLS84] R. V. Book, T. Long, and A. L. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13:461–487, 1984.

[BS85] J. L. Balcázar and U. Schöning. Bi-immune sets for complexity classes. *Mathematical Systems Theory*, 18(1):1–10, 1985.

[FGH$^+$99] S. Fenner, F. Green, S. Homer, A. L. Selman, T. Thierauf, and H. Vollmer. Complements of multivalued functions. *Chicago Journal of Theoretical Computer Science*, 1999. Article 3 of volume 1999.

[FHOS97] S. Fenner, S. Homer, M. Ogihara, and A. L. Selman. Oracles that compute values. *SIAM Journal on Computing*, 26:1043–1065, 1997.

[GRSW10] C. Glaßer, C. Reitwießner, H. Schmitz, and M. Witek. Approximability and hardness in multi-objective optimization. Number 6158 in Lecture Notes in Computer Science, pages 180–189. Springer-Verlag, 2010.

[HNOS96] L. Hemaspaandra, A. Naik, M. Ogihara, and A. L. Selman. Computing solutions uniquely collapses the polynomial hierarchy. *SIAM Journal on Computing*, 25:697–708, 1996.

[HW00] H. Hempel and G. Wechsung. The operators min and max on the polynomial hierarchy. *International Journal of Foundations of Computer Science*, 11(2):315–342, 2000.

[Kre88] M. W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.

[MA78] K. L. Manders and L. Adleman. NP-complete decision problems for binary quadratics. *Journal of Computer and System Sciences*, 16(2):168 – 184, 1978.

[PY82] C. H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *J. ACM*, 29(2):285–309, 1982.

[Sel92] A. L. Selman. A survey of one-way functions in complexity theory. *Mathematical Systems Theory*, 25:203–221, 1992.

[Sel94] A. L. Selman. A taxonomy on complexity classes of functions. *Journal of Computer and System Sciences*, 48:357–381, 1994.

[Sel96] A. L. Selman. Much ado about functions. In *Proceedings 11th Conference on Computational Complexity*, pages 198–212. IEEE Computer Society Press, 1996.

[Val76] L. G. Valiant. Relative complexity of checking and evaluating. *Information Processing Letters*, 5(1):20–23, 1976.