Complexity of Counting the Optimal Solutions

Miki Hermann[1] and Reinhard Pichler[2]

[1] LIX (CNRS, UMR 7161), École Polytechnique, 91128 Palaiseau, France.
hermann@lix.polytechnique.fr
[2] Institut für Informationssysteme, Technische Universität Wien, A-1040 Wien, Austria.
pichler@dbai.tuwien.ac.at

Abstract. Following the approach of Hemaspaandra and Vollmer, we can define counting complexity classes $\#\cdot\mathcal{C}$ for any complexity class $\mathcal{C}$ of decision problems. In particular, the classes $\#\cdot\Pi_k\mathrm{P}$ with $k \geq 1$ corresponding to all levels of the polynomial hierarchy have thus been studied. However, for a large variety of counting problems arising from optimization problems, a precise complexity classification turns out to be impossible with these classes. In order to remedy this unsatisfactory situation, we introduce a hierarchy of new counting complexity classes $\#\cdot\mathrm{Opt}_k\mathrm{P}$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ with $k \geq 1$. We prove several important properties of these new classes, like closure properties and the relationship with the $\#\cdot\Pi_k\mathrm{P}$-classes. Moreover, we establish the completeness of several natural counting complexity problems for these new classes.

1 Introduction

Many natural decision problems are known to be complete for the class $\Theta_k\mathrm{P} = \Delta_k\mathrm{P}[\log n]$, defined by Wagner in [19], or for $\Delta_k\mathrm{P}$. In particular, they often occur in variants of $\Sigma_{k-1}\mathrm{P}$-complete problems when cardinality-minimality or weight-minimality (or, likewise, cardinality-maximality or weight-maximality) is imposed as an additional constraint. Two prototypical representatives of such problems are as follows (The completeness of these problems in $\Theta_2\mathrm{P}$ and $\Delta_2\mathrm{P}$ is implicit in [6]).

Problem: MIN-CARD-SAT (MIN-WEIGHT-SAT)
Input: A propositional formula φ in conjunctive normal form over variables X (together with a weight function $w\colon X \to \mathbb{N}$) and a subset of variables $X' \subseteq X$.
Question: Are X' set to true in some cardinality-minimal (weight-minimal) model of φ?

A straightforward $\Theta_2\mathrm{P}$-algorithm for MIN-CARD-SAT computes the minimum cardinality of the models of φ by means of logarithmically many calls to an NP-oracle, asking questions of the type "Does φ have a model of size $\leq k$?". As soon as the minimum cardinality k_0 is known, we can proceed by a simple NP-algorithm, checking if the subset X' is true in some model of size k_0. Analogously, a $\Delta_2\mathrm{P}$-algorithm for MIN-WEIGHT-SAT first computes the minimum weight of all models of φ. In any reasonable representation, the weights are exponential with respect to their representation (e.g., they are represented in binary notation). Hence, the straightforward algorithm for computing the minimum weight needs logarithmically many calls to an NP-oracle with respect to the total weight of all variables. This comes down to polynomially many calls with respect to the representation of the weights.

Note that the membership in $\Theta_2\mathrm{P}$ and $\Delta_2\mathrm{P}$ recalled above is in great contrast to *subset-minimality*, i.e., minimality with respect to set inclusion (or, likewise, *subset-maximality*), which often raises the complexity one level higher in the polynomial hierarchy. E.g., the following problem is well-known to be $\Sigma_2\mathrm{P}$-complete (cf. [13]).

Problem: MIN-SAT
Input: A propositional formula φ in conjunctive normal form over variables X and a subset $X' \subseteq X$.
Question: Are X' set to true in some subset-minimal model of φ?

As far as the complexity of the corresponding *counting problems* is concerned, only the counting problem corresponding to MIN-SAT has been satisfactorily classified so far. The following problem was shown to be $\#\cdot\mathrm{coNP}$-complete in [3]: Given a propositional formula φ in conjunctive normal form, how many subset-minimal models does φ have? On the other hand, the counting complexity of the remaining aforementioned problems has remained obscure. The main goal of this paper is to introduce new counting complexity classes $\#\cdot\mathrm{OptP}$ and $\#\cdot\mathrm{OptP}[\log n]$, needed to pinpoint the precise complexity of these and many similar optimality counting problems. We will also show the relationship of these new classes with respect to the known classes in the counting hierarchy. Moreover, we will show that these new classes are not identical to already known ones, unless the polynomial hierarchy collapses. Finally, we will present several natural optimization counting problems, which turn out to be complete for one or the other introduced counting class. The definition of new natural counting complexity classes is by no means limited to the first level of the polynomial hierarchy. Indeed, we will show how the counting complexity classes $\#\cdot\mathrm{OptP}$ and $\#\cdot\mathrm{OptP}[\log n]$ can be generalized to $\#\cdot\mathrm{Opt}_k\mathrm{P}$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ for arbitrary $k \geq 1$ with $\#\cdot\mathrm{OptP} = \#\cdot\mathrm{Opt}_1\mathrm{P}$ and $\#\cdot\mathrm{OptP}[\log n] = \#\cdot\mathrm{Opt}_1\mathrm{P}[\log n]$.

Due to lack of space, proofs had to be omitted in this paper. A full version with detailed proofs of all results presented here is provided as a technical report.

2 Preliminaries

We recall the necessary concepts and definitions, but we assume that the reader is familiar with the basic notions in computational counting complexity. For more information, the interested reader is referred to Chapter 18 in the book [13] or the survey [4].

The study of *counting problems* was initiated by Valiant in [17, 18]. While decision problems ask if at least one solution of a given problem instance exists, counting problems ask for the number of different solutions. The most intensively studied counting complexity class is $\#\mathrm{P}$, which denotes the functions that count the number of accepting paths of a non-deterministic polynomial-time Turing machine. In other words, $\#\mathrm{P}$ captures the counting problems corresponding to decision problems contained in NP. By allowing the non-deterministic polynomial-time Turing machine access to an oracle in NP, $\Sigma_2\mathrm{P}, \ldots$, we can define an infinite hierarchy of counting complexity classes.

Alternatively, a *counting problem* is presented using a suitable *witness* function which for every input x, returns a set of *witnesses* for x. Formally, a *witness* function is a function $A\colon \Sigma^* \to \mathcal{P}^{<\omega}(\Gamma^*)$, where Σ and Γ are two alphabets, and $\mathcal{P}^{<\omega}(\Gamma^*)$ is the collection of all finite subsets of Γ^*. Every such witness function gives rise to the following *counting problem*: given a string $x \in \Sigma^*$, find the cardinality $|A(x)|$ of the *witness* set $A(x)$. According to [7], if $\mathcal{C}$ is a complexity class of decision problems, we define $\#\cdot\mathcal{C}$ to be the class of all counting problems $\#\cdot A$ whose witness function A satisfies the following conditions.

1. There is a polynomial $p(n)$ such that for every x and every $y \in A(x)$, we have that $|y| \leq p(|x|)$, where $|x|$ is the length of x and $|y|$ is the length of y;
2. The decision problem "given x and y, is $y \in A(x)$?" is in $\mathcal{C}$.

It is easy to verify that $\#\mathrm{P} = \#\cdot\mathrm{P}$. The counting hierarchy is ordered by linear inclusion [7]. In particular, we have that $\#\mathrm{P} \subseteq \#\cdot\mathrm{coNP} \subseteq \#\cdot\Pi_2\mathrm{P} \subseteq \#\cdot\Pi_3\mathrm{P}$, etc. Analogously, one can define the classes $\#\cdot\mathrm{NP}$, $\#\cdot\Sigma_2\mathrm{P}$, $\#\cdot\Sigma_3\mathrm{P}$, etc. Toda and Ogiwara [15] determined the precise relationship between these classes as follows: $\#\cdot\Sigma_k\mathrm{P} \subseteq \#\cdot\mathrm{P}^{\Sigma_k\mathrm{P}} = \#\cdot\Pi_k\mathrm{P}$. Since the identity $\#\cdot\mathrm{P}^{\Sigma_k\mathrm{P}} = \#\cdot\Delta_{k+1}\mathrm{P}$ trivially holds, Toda and Ogiwara showed that there are no new Δ-classes in the counting hierarchy.

The prototypical $\#\cdot\Pi_k\mathrm{P}$-complete problem for $k \in \mathbb{N}$ is $\#\Pi_k\mathrm{SAT}$ [3], defined as follows. Given a formula $\psi(X) = \forall Y_1 \exists Y_2 \cdots QY_k\ \varphi(X, Y_1, \ldots, Y_k)$, where φ is a Boolean formula and X, Y_1, …, Y_k are sets of propositional variables, count the number of truth assignments to the variables in X that satisfy ψ.

Completeness of counting problems is usually proved by means of polynomial-time Turing reductions, also called Cook reductions. However, these reductions do not preserve the counting classes $\#\cdot\Pi_k\mathrm{P}$ [16]. Hence, *parsimonious reductions* are usually considered instead. Consider two counting problems $\#\cdot A\colon \Sigma^* \to \mathbb{N}$ and $\#\cdot B\colon \Sigma^* \to \mathbb{N}$. We say that $\#\cdot A$ reduces to $\#\cdot B$ via a parsimonious reduction if there exists a polynomial-time function $f \in \mathrm{FP}$, such that for each $x \in \Sigma^*$ we have $\#\cdot A(x) = \#\cdot B(f(x))$. Parsimonious reductions are a special case of Karp reductions with a one-to-one relation between solutions for the corresponding instances of the problems $\#\cdot A$ and $\#\cdot B$. However, parsimonious reductions are not always strong enough to prove completeness of well-known problems in counting complexity classes. E.g., the problem #POSITIVE 2SAT [2,18] of counting satisfying assignments to a propositional formula with positive literals only and with two literals per clause cannot be $\#\mathrm{P}$-complete under parsimonious reductions, unless $\mathrm{P} = \mathrm{NP}$. Therefore Durand, Hermann, and Kolaitis [3] generalized parsimonious reductions to *subtractive reductions* and showed that all the classes $\#\cdot\Pi_k\mathrm{P}$ are closed under them. Subtractive reductions are defined as follows. The counting problem $\#\cdot A$ reduces to $\#\cdot B$ via a *strong subtractive reduction* if there exist two polynomial-time computable functions f and g such that for each $x \in \Sigma^*$ we have $B(f(x)) \subseteq B(g(x))$ and $|A(x)| = |B(g(x))| - |B(f(x))|$.

A *subtractive reduction* is a composition (transitive closure) of a finite sequence of strong subtractive reductions. Thus, a *parsimonious* reduction corresponds to the special case of a strong subtractive reduction with $B(f(x)) = \emptyset$. In [3], subtractive reductions have been shown to be strong enough to prove completeness of many interesting problems in $\#\mathrm{P}$ and other counting classes, but their power remains tame enough to preserve several interesting counting classes between $\#\mathrm{P}$ and $\#\mathrm{PSPACE}$.

3 Optimization Counting Complexity Classes

Recall that, according to [7], a counting complexity class $\#\cdot\mathcal{C}$ can in principle be defined for any decision complexity class $\mathcal{C}$. However, as far as the polynomial hierarchy is concerned, this definition does not yield the desired diversity of counting complexity classes. In fact, if we simply consider $\#\cdot\mathcal{C}$ for either $\mathcal{C} = \Delta_k\mathrm{P}$ or $\mathcal{C} = \Theta_k\mathrm{P}$, then we do not get any new complexity classes, since the relationship $\#\cdot\Theta_k\mathrm{P} = \#\cdot\Delta_k\mathrm{P} = \#\cdot\Pi_{k-1}\mathrm{P}$ is an immediate consequence of the aforementioned result by Toda and Ogiwara [15].

Hence a different approach is necessary if we want to obtain a more fine grained stratification of the counting hierarchy. For this reason we introduce in the sequel the counting classes $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}$ for each $k \in \mathbb{N}$, which will be appropriate for optimization counting problems. Of special interest will be the classes $\#\cdot\mathrm{OptP}[\log n] = \#\cdot\mathrm{Opt}_1\mathrm{P}[\log n]$ and $\#\cdot\mathrm{OptP} = \#\cdot\mathrm{Opt}_1\mathrm{P}$. We will define the new counting complexity classes via the nondeterministic transducer model (see [14]), as well as by an equivalent predicate based definition following the approach from [7]. The following definition generalizes the definition of nondeterministic transducers [14] to oracle machines.

Definition 1. *A* nondeterministic transducer M *is a nondeterministic polynomial-time bounded Turing machine, such that every accepting path writes a binary number. If* M *is equipped with an oracle from the complexity class* $\mathcal{C}$*, then it is called a* nondeterministic transducer with $\mathcal{C}$-oracle. *A* $\Sigma_k\mathrm{P}$-transducer M *is a nondeterministic transducer with a* $\Sigma_{k-1}\mathrm{P}$ *oracle. We identify nondeterministic transducers without oracle and* $\Sigma_1\mathrm{P}$-*transducers.*

For $x \in \Sigma^*$*, we write* $\mathrm{opt}_M(x)$ *to denote the* optimal *value, which can be either the* maximum *or the* minimum*, on any accepting path of the computation of* M *on* x*. If no accepting path exists then* $\mathrm{opt}_M(x)$ *is undefined.*

The above definition of a nondeterministic transducer is similar to a metric Turing machine defined in [10] and its generalization in [11]. However, our definition deviates from the machine models in [10, 11] in the following aspects:

1. We take the optimum value only over the *accepting* paths, while in [10] every path is accepting. Our ultimate goal is to count the number of optimal solutions. Hence, above all, the objects that we want to count have to be *solutions*, i.e., correspond to an accepting computation, and only in the second place we are interested in the optimum.
2. In [10], only the maximum value is considered and it is mentioned that the minimum value is treated analogously. We prefer to make the applicability both to $\max$ and $\min$ explicit. The definition of the counting complexity classes below is not affected by this distinction.
3. In [11], NP-metric Turing machines were generalized to higher levels of the polynomial hierarchy by allowing alternations of minimum and maximum computations. However, for our purposes, in particular for the predicate-based characterization of the counting complexity classes below, the generalization via oracles is more convenient. Proving the equivalence of the two kinds of generalizations is straightforward.

It will be clear in the sequel that the generalization of nondeterministic transducers [14] to oracle machines is exactly the model we need. A similar idea but with a deterministic Turing transducer was used by Jenner and Torán in [8] to characterize the functional complexity classes $\mathrm{FP}^{\mathrm{NP}}_{\|}$, $\mathrm{FP}^{\mathrm{NP}}_{\log}$, and $\mathrm{FL}^{\mathrm{NP}}_{\log}$.

Definition 2. *We say that a counting problem $\#{\cdot}A\colon \Sigma^* \to \mathbb{N}$ is in the class $\#{\cdot}\mathrm{Opt}_k\mathrm{P}$ for some $k \in \mathbb{N}$, if there is a $\Sigma_k\mathrm{P}$-transducer M, such that $\#{\cdot}A(x)$ is the number of accepting paths of the computation of M on x yielding the optimum value $\mathrm{opt}_M(x)$. If no accepting path exists then $\#{\cdot}A(x) = 0$. If the length of the binary number written by M is bounded by $O(z(|x|))$ for some function $z(n)$, then $\#{\cdot}A$ is in the class $\#{\cdot}\mathrm{Opt}_k\mathrm{P}[z(n)]$.*

In this paper, we are only interested in $\#{\cdot}\mathrm{Opt}_k\mathrm{P}[z(n)]$ for two types of functions $z(n)$, namely the polynomial function $z(n) = n^{O(1)}$ and the logarithmic function $z(n) = \log n$. Clearly, $\#{\cdot}\mathrm{Opt}_k\mathrm{P}$ is the same as $\#{\cdot}\mathrm{Opt}_k\mathrm{P}[n^{O(1)}]$.

Distinguishing between $\max$ and $\min$ gives no additional computational power, as it is formalized by the following result.

Proposition 3. *Suppose that some counting problem $\#{\cdot}A\colon \Sigma^* \to \mathbb{N}$ is defined in terms of a $\Sigma_k\mathrm{P}$-transducer M with the optimum being the maximum (minimum). Then there exists a parsimonious reduction to a counting problem $\#{\cdot}A'$ defined via a $\Sigma_k\mathrm{P}$-transducer M' with the optimum value corresponding to the minimum (maximum).*

Krentel defined in [10] the class $\mathrm{OptP}[z(n)]$ of optimization problems for a given function $z(n)$. He showed that $\mathrm{OptP}[z(n)]$ essentially corresponds to $\mathrm{FP}^{\mathrm{NP}[z(n)]}$ (see also [9]). More precisely, for every "smooth" function[3] $z(n)$ (see [10]) we have $\mathrm{OptP}[z(n)] \subseteq \mathrm{FP}^{\mathrm{NP}[z(n)]}$ and every function $f \in \mathrm{FP}^{\mathrm{NP}[z(n)]}$ can be represented as an $\mathrm{OptP}[z(n)]$-problem followed by a polynomial-time function h. This correspondence between $\mathrm{OptP}[z(n)]$ and $\mathrm{FP}^{\mathrm{NP}[z(n)]}$ can be generalized as follows: replacing the $\Sigma_{k-1}\mathrm{P}$ oracle in a $\Sigma_k\mathrm{P}$-transducer by a $\Delta_k\mathrm{P}$ oracle does not increase the expressive power.

We show next that the definition of $\#{\cdot}\mathrm{Opt}_k\mathrm{P}[z(n)]$ via Turing machines (see Definition 1) has an equivalent definition via predicates. The basic idea is to decompose the computation of a $\Sigma_k\mathrm{P}$-transducer M into a predicate B, which associates inputs x with computations y, and a function f which computes the number written by the transducer M following the computation path y.

Theorem 4. *For any function $z(n)$, a counting problem $\#{\cdot}A\colon \Sigma^* \to \mathbb{N}$ is in the class $\#{\cdot}\mathrm{Opt}_k\mathrm{P}[z(n)]$ if and only if there exist an alphabet Γ, a predicate B on $\Sigma^* \times \Gamma^*$, and a polynomial-time computable function $f\colon \Gamma^* \to \mathbb{N}$ satisfying the following conditions.*

- *(i) There is a polynomial $p(n)$ such that every pair of strings $(x, y) \in B$ satisfies the relation $|y| \leq p(|x|)$;*
- *(ii) The predicate B is decidable by a $\Delta_k\mathrm{P}$ algorithm;*
- *(iii) The length of $f(y)$ is bounded by $O(z(|x|))$ for every (x, y);*

[3] A function $f\colon \mathbb{N} \to \mathbb{N}$ is *smooth* if it is nondecreasing and its unary representation is computable in polynomial time.

(iv) $\mathrm{opt}_f^B(x) = \mathrm{opt}(\{f(y) \mid (x,y) \in B\})$ *with* $\mathrm{opt} \in \{\max, \min\}$*;*
(v) $A(x) = \{y \mid (x,y) \in B \wedge f(y) = \mathrm{opt}_f^B(x)\}$.

As far as complete problems for these new complexity classes are concerned, we propose the following natural generalizations of minimum cardinality and minimum weight counting satisfiability problems to quantified Boolean formulas.

Problem: #MIN-CARD-Π_kSAT (#MIN-WEIGHT-Π_kSAT)
Input: A Π_kSAT formula $\psi(X) = \forall Y_1 \exists Y_2 \cdots QY_k\ \varphi(X, Y_1, \ldots, Y_k)$ with $k \in \mathbb{N}$, where φ is a quantifier-free formula and $X, Y_1, \ldots, Y_k$ are sets of propositional variables, (and a weight function $w\colon X \to \mathbb{N}$ assigning positive values to each variable $x \in X$) such that Q is either $\exists$ for k even or $\forall$ for k odd.
Output: Number of cardinality-minimal (weight-minimal) models of $\psi(X)$ or 0 if $\psi(X)$ is unsatisfiable.

We define the classes #MIN-CARD-SAT and #MIN-WEIGHT-SAT to be the #MIN-CARD-Π_0SAT and #MIN-WEIGHT-Π_0SAT, respectively. Moreover, we can assume, following the ideas of Wrathal [20], that the formula φ is in CNF for k even and in DNF for k odd. Notice that for k *even* (*odd*), the formula φ has an *odd* (*even*) number of variable vectors, since the first variable block X remains always unquantified.

Theorem 5. *For every $k \in \mathbb{N}$, the following problems are complete via parsimonious reductions.* #MIN-WEIGHT-Π_kSAT *is* $\#\cdot\mathrm{Opt}_{k+1}\mathrm{P}$*-complete and* #MIN-CARD-Π_kSAT *is* $\#\cdot\mathrm{Opt}_{k+1}\mathrm{P}[\log n]$*-complete.*

As usual, also the versions of #MIN-WEIGHT-Π_kSAT and #MIN-CARD-Π_kSAT restricted to 3 literals per clause are $\#\cdot\mathrm{Opt}_{k+1}\mathrm{P}$-complete and $\#\cdot\mathrm{Opt}_{k+1}\mathrm{P}[\log n]$-complete, respectively, since there exists a parsimonious reduction to them.

Apart from containing natural complete problems, a complexity class should also be closed with respect to an appropriate type of reductions. We consider the closure of the considered counting classes under subtractive reductions. Note that we cannot expect the class $\#\cdot\mathrm{Opt}_k\mathrm{P}[z(n)]$ to be closed under subtractive reductions for any function $z(n)$ since we can always get an arbitrary polynomial speed-up simply by padding the input. We show in the sequel that the two most interesting cases, namely $\#\cdot\mathrm{Opt}_k\mathrm{P}$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ for each $k \in \mathbb{N}$, are indeed closed under subtractive reductions.

Theorem 6. *The complexity classes $\#\cdot\mathrm{Opt}_k\mathrm{P}$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ are closed under subtractive reductions for all $k \in \mathbb{N}$.*

Our new considered classes $\#\cdot\mathrm{Opt}_k\mathrm{P}$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ need to be confronted with the already known counting hierarchy. We will present certain inclusions of the new classes with respect to already known counting complexity classes and show that the inclusions are proper, unless the polynomial hierarchy collapses.

Theorem 7. *We have $\#\cdot\Pi_k\mathrm{P} \subseteq \#\cdot\mathrm{Opt}_{k+1}\mathrm{P}[\log n] \subseteq \#\cdot\mathrm{Opt}_{k+1}\mathrm{P} \subseteq \#\cdot\Pi_{k+1}\mathrm{P}$ for each $k \in \mathbb{N}$.*

Finally, the following result shows that the new classes are robust.

Theorem 8. *If $\#\cdot\mathrm{Opt}_{k+1}\mathrm{P}[\log n]$ or $\#\cdot\mathrm{Opt}_{k+1}\mathrm{P}$ coincides with either $\#\cdot\Pi_k\mathrm{P}$ or $\#\cdot\Pi_{k+1}\mathrm{P}$ for some $k \in \mathbb{N}$, then the polynomial hierarchy collapses to the k-th or $(k+1)$-st level, respectively.*

4 Further Optimization Counting Problems

The most interesting optimization counting problems are of course those belonging to the classes on the first level of the optimization counting hierarchy, namely $\#\cdot\mathrm{OptP}$ and $\#\cdot\mathrm{OptP}[\log n]$. In this section we will focus on such problems of particular interest.

Gasarch *et al.* presented in [6] a plethora of optimization problems complete for OptP and $\mathrm{OptP}[\log n]$. Either their lower bound is already proved by a parsimonious reduction or the presented reduction can be transformed into a parsimonious one similarly to Galil's construction in [5]. The counting version of virtually all these problems can therefore be proved to be complete for $\#\cdot\mathrm{OptP}$ or $\#\cdot\mathrm{OptP}[\log n]$. Likewise, Krentel presented in [11] several problems belonging to higher levels of the optimization hierarchy. They give rise to counting problems complete for $\#\cdot\mathrm{Opt}_k\mathrm{P}$ or $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ with $k > 1$.

Problem: #MIN-CARD-SAT
Input: A propositional formula φ in conjunctive normal form over the variables X.
Output: Number of models of φ with minimal Hamming weight.

The dual problem #MAX-CARD-SAT asks for the number of models with *maximal* Hamming weight. The problems #MIN-WEIGHT-SAT and #MAX-WEIGHT-SAT are the corresponding weighted versions of the aforementioned problems.

Following Theorem 5, both counting problems #MIN-CARD-SAT and #MAX-CARD-SAT are $\#\cdot\mathrm{OptP}[\log n]$-complete, whereas #MIN-WEIGHT-SAT and #MAX-WEIGHT-SAT are $\#\cdot\mathrm{OptP}$-complete. We consider only the cardinality-minimal problems in the sequel.

It is also interesting to investigate special cases of the optimization counting problems involving the following restrictions on the formula φ. As usual, a literal is a propositional variable (positive literal) or its negation (negative literal), whereas a clause is a disjunction of literals, and a formula in conjunctive normal form is a conjunction of clauses. We say that a clause c is **Horn** if it contains at most one positive literal, **dual Horn** if it contains at most one negative literal, **Krom** if it contains at most two literals. A formula $\varphi = c_1 \wedge \cdots \wedge c_n$ in conjunctive normal form is Horn, dual Horn, or Krom if all clauses c_i for $i = 1, \ldots, n$ satisfy the respective condition. Formulas restricted to conjunctions of Horn, dual Horn, or Krom clauses are often investigated in computational problems related to artificial intelligence, in particular to closed world reasoning [1]. We denote by the specification in brackets the restriction of the counting problem #MIN-CARD-SAT to the respective class of formulas.

The models of Horn formulas are closed under conjunction, i.e., for two models m and m' of a Horn formula φ, also the Boolean vector $m \wedge m' = (m[1] \wedge m'[1], \ldots, m[k] \wedge m'[k])$ is a model of φ. Hence there exists a unique model with minimal Hamming weight if and only if φ is satisfiable. Therefore a Horn formula φ has either one cardinality-minimal model or none, depending on the satisfiability of φ. A similar situation arises for #MIN-CARD-DNF, the problem of counting the number of assignments with minimal Hamming weight to a propositional formula in disjunctive normal form. These considerations imply the following results.

Proposition 9. #MIN-CARD-SAT[HORN] *and* #MIN-CARD-DNF *are in* FP.

Vertex covers, cliques, and independent sets have a particular relationship. The set X is a smallest vertex cover in $G = (V, E)$ if and only if $V \smallsetminus X$ is a largest independent set in G if and only if $V \smallsetminus X$ is a largest clique in the complement graph $\bar{G} = (V, V \times V \smallsetminus E)$. The size of the largest clique has been investigated by Krentel [10] and proved to be $\mathrm{OptP}[\log n]$-complete (the same proof is also given in [13]). Using this knowledge, we can determine the complexity of the following problems.

Problem: #MAX-CARD-INDEPENDENT SET
Input: Graph $G = (V, E)$.
Output: Number of independent sets in G with maximum cardinality, i.e., number of subsets $V' \subseteq V$ where $|V'|$ is maximal and for all $u, v \in V'$ we have $(u, v) \notin E$.

Problem: #MAX-CARD-CLIQUE
Input: Graph $G = (V, E)$.
Output: Number of cliques in G with maximum cardinality, i.e., number of subsets $V' \subseteq V$ where $|V'|$ is maximal and $(u, v) \in E$ holds for all $u, v \in V'$ such that $u \neq v$.

Problem: #MIN-CARD-VERTEX COVER
Input: Graph $G = (V, E)$.
Output: Number of vertex covers of G with minimal cardinality, i.e., number of subsets $V' \subseteq V$ where $|V'|$ is minimal and $(u, v) \in E$ implies $u \in V'$ or $v \in V'$.

Theorem 10. *The problems #MAX-CARD-INDEPENDENT SET, #MAX-CARD-CLIQUE, and #MIN-CARD-VERTEX COVER are $\#{\cdot}\mathrm{OptP}[\log n]$-complete. Their weighted versions are $\#{\cdot}\mathrm{OptP}$-complete.*

We can easily transform the counting problem #MIN-CARD-VERTEX COVER to both #MIN-CARD-SAT[DUAL HORN] and #MIN-CARD-SAT[KROM]. Indeed, we can represent an edge $(u, v) \in E$ of a graph $G = (V, E)$ by a clause $(u \vee v)$ which is both Krom and dual Horn. Hence a cardinality-minimal vertex cover of a graph $G = (V, E)$ corresponds to a cardinality-minimal model of the formula $\varphi_G = \bigwedge_{(u,v) \in E} (u \vee v)$.

Corollary 11. *The counting problems #MIN-CARD-SAT[DUAL HORN] and #MIN-CARD-SAT[KROM] are $\#{\cdot}\mathrm{OptP}[\log n]$-complete via parsimonious reductions.*

The following problem is a classic in optimization theory. It is usually formulated as the maximal number of clauses that can be satisfied. We can also ask for the number of truth assignments that satisfy the maximal number of clauses.

Problem: #MAX2SAT
Input: A propositional formula φ in conjunctive normal form over the variables X with at most two variables per clause.
Output: Number of assignments to φ that satisfy the maximal number of clauses.

The optimization variant of the following counting problem is presented in [6] under the name CHEATING SAT. We can interpret it as a satisfiability problem in a 3-valued logic, where the middle value τ is a "don't-know". In this setting it is interesting to investigate the minimal size of uncertainty we need to satisfy a formula for the optimization variant, as well as the number of satisfying assignments with the minimal size of uncertainty.

Problem: #MIN-SIZE UNCERTAINTY SAT
Input: A propositional formula φ in conjunctive normal form over the variables X.
Output: Number of satisfying assignments $m\colon X \to \{0, \tau, 1\}$ of the formula φ, where $m(x) = \tau$ satisfies both literals x and $\neg x$, with minimal cardinality of the set $\{x \in X \mid m(x) = \tau\}$.

Theorem 12. *#MAX2SAT and #MIN-SIZE UNCERTAINTY SAT are* $\#\cdot\mathrm{OptP}[\log n]$*-complete.*

Even though the complete problems for the classes $\#\cdot\mathrm{OptP}$ and $\#\cdot\mathrm{OptP}[\log n]$ are the most interesting ones, there also exist some interesting complete problems in the classes $\#\cdot\mathrm{Opt}_k\mathrm{P}$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$ for $k > 1$. The following problem is an example of such a case.

Problem: #MAXIMUM k-QUANTIFIED CIRCUIT
Input: A Boolean circuit $C(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ over variable vectors $\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k$.
Output: Number of maximum values $\boldsymbol{x} \in \{0,1\}^n$ in binary notation satisfying the quantified expression $\forall \boldsymbol{y}_1 \exists \boldsymbol{y}_2 \cdots Q \boldsymbol{y}_k \ (C(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k) = 1)$, where Q is either $\forall$ or $\exists$ depending on the parity of k.

Theorem 13. *#MAXIMUM k-QUANTIFIED CIRCUIT is* $\#\cdot\mathrm{Opt}_k\mathrm{P}$*-complete.*

5 Concluding Remarks

In the scope of the result from [16] showing that all classes between $\#\mathrm{P}$ and $\#\mathrm{PH}$, the counting equivalent of the polynomial hierarchy, collapse to $\#\mathrm{P}$ under 1-Turing reductions, it is necessary (1) to find suitable reductions strong enough to prove completeness of well-known counting problems, but tame enough to preserve at least some counting classes, (2) to identify counting classes with interesting complete problems preserved under the aforementioned reduction. The first problem was mainly addressed in [3], whereas in this paper we focused on the second point. We introduced a new hierarchy of optimization counting complexity classes $\#\cdot\mathrm{Opt}_k\mathrm{P}$ and $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$. These classes allowed us to pinpoint the complexity of many natural optimization counting problems which had previously resisted a precise classification. Moreover, we have shown that these new complexity classes have several desirable properties and they interact well with the counting hierarchy defined by Hemaspaandra and Vollmer in [7]. Nevertheless, the Hemaspaandra-Vollmer counting hierarchy does not seem to be sufficiently detailed to capture all interesting counting problems. Therefore an even more fine-grained stratification of the counting complexity classes is necessary, which started with the contribution of Pagourtzis and Zachos [12] and has been pursued in this paper.

Finally, further decision problems in $\Delta_k\mathrm{P}$ (respectively $\Theta_k\mathrm{P}$) with $k \in \mathbb{N}$ and corresponding counting problems should be inspected. It should be investigated if the complexity of the latter can be precisely identified now that we have the new counting complexity classes $\#\cdot\mathrm{Opt}_k\mathrm{P}$ (respectively $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$) at hand. Moreover, we would also like to find out more about the nature of the problems that are complete for these new counting complexity classes. In particular, it would be very interesting to find out if there also exist "easy to decide, hard to count" problems, i.e., problems whose counting variant is complete for $\#\cdot\mathrm{Opt}_k\mathrm{P}$ (respectively $\#\cdot\mathrm{Opt}_k\mathrm{P}[\log n]$) while the corresponding decision problem is below $\Delta_k\mathrm{P}$ (respectively $\Theta_k\mathrm{P}$). Clearly, such a phenomenon can only exist if we consider completeness with respect to reductions stronger than the parsimonious ones. Hence, the closure of our new counting classes under *subtractive reductions* (rather than just under parsimonious reductions) in Theorem 6 is an indispensable prerequisite for further research in this direction.

References

1. M. Cadoli and M. Lenzerini. The complexity of propositional closed world reasoning and circumscription. *Journal of Computer and System Sciences*, 48(2):255–310, 1994.
2. N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125(1):1–12, 1996.
3. A. Durand, M. Hermann, and P. G. Kolaitis. Subtractive reductions and complete problems for counting complexity classes. *Theoretical Computer Science*, 340(3):496–513, 2005.
4. L. Fortnow. Counting complexity. In L. A. Hemaspaandra and A. L. Selman, editors, *Complexity theory retrospective II*, pages 81–107. Springer-Verlag, 1997.
5. Z. Galil. On some direct encodings of nondeterministic Turing machines operating in polynomial time into P-complete problems. *SIGACT News*, 6(1):19–24, January 1974.
6. W. I. Gasarch, M. W. Krentel, and K. J. Rappoport. OptP as the normal behavior of NP-complete problems. *Mathematical Systems Theory*, 28(6):487–514, 1995.
7. L. A. Hemaspaandra and H. Vollmer. The satanic notations: Counting classes beyond #P and other definitional adventures. *SIGACT News, Complexity Theory Column 8*, 26(1):2–13, March 1995.
8. B. Jenner and J. Torán. Computing functions with parallel queries to NP. *Theoretical Computer Science*, 141(1-2):175–193, 1995.
9. J. Köbler, U. Schöning, and J. Torán. On counting and approximation. *Acta Informatica*, 26(4):363–379, 1989.
10. M. W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.
11. M. W. Krentel. Generalizations of OptP to the polynomial hierarchy. *Theoretical Computer Science*, 97(2):183–198, 1992.
12. A. Pagourtzis and S. Zachos. The complexity of counting functions with easy decision version. In R. Královič and P. Urzyczyn, editors, *Proceedings 31st MFCS, Stará Lesná (Slovakia)*, LNCS 4162, pages 741–752. Springer, 2006.
13. C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
14. A. Selman, X. Mei-Rui, and R. Book. Positive relativizations of complexity classes. *SIAM Journal on Computing*, 12(3):565–579, 1983.
15. S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 21(2):316–328, 1992.
16. S. Toda and O. Watanabe. Polynomial-time 1-Turing reductions from #PH to #P. *Theoretical Computer Science*, 100(1):205–221, 1992.
17. L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
18. L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.
19. K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.
20. C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.