SIGACT News Complexity Theory Column 38

Lane A. Hemaspaandra
Dept. of Computer Science, University of Rochester
Rochester, NY 14627, USA `lane@cs.rochester.edu`

Introduction to Complexity Theory Column 38

Lane's text.

Guest Column: Completeness in the Polynomial-Time Hierarchy: Part II[1]

Marcus Schaefer[2] *Christopher Umans*[3]

Abstract

In this survey, we highlight selected complete problems from Part I of this column, and include reductions for some of them. We also discuss methods for proving hardness of approximation in the second and higher levels of the Polynomial-Time Hierarchy, and illustrate with two full proofs of non-approximability utilizing error-correcting codes.

1 Introduction

Our goal in this column is to complement the rather terse compendium we compiled for Part I with some discussion that we hope may serve as a jumping-off point for researchers hoping to prove completeness and/or hardness of approximation for a problem that appears to be above the first level of the Polynomial-Time Hierarchy. In addition to providing some completeness proofs, and we also take the opportunity to emphasize that many hardness of approximation results for optimization problems residing in these levels rely on less complicated tools than Probabilistically Checkable Proofs (PCPs), while still employing sophisticated general techniques, mainly involving *list-decoding* of error-correcting codes. We will see some of these proofs.

The paper is organized as follows. In Section 2 we highlight some natural problems complete for the second and third levels of the Polynomial-Time Hierarchy, and provide reductions for three of them. In Section 3 we discuss methods for proving non-approximability for optimization problems in the second and higher levels of the Polynomial-Time Hierarchy. We conclude the section with two full proofs of non-approximability that illustrate these methods. In Section 4 we mention some open problems.

To entice the reader to continue on, we would like to point out that in the proofs we repeat in this paper, we have taken some care to distill only the essential ideas from the original papers. The resulting arguments are often quite simple and concise, with the only price being that in Section 3.2, the two proofs do not quite obtain the best-possible non-approximability factors.

Two problems, QSAT_2 and QSAT_3 are the starting points for most of the reductions in this paper; before we move on we briefly review the general definition here:

---

[2]School of CTI, DePaul Univ., 243 S. Wabash Ave., Chicago, IL 60604, USA, `schaefer@cs.depaul.edu`.

[3]Computer Science Dept., Caltech, 1200 East California Blvd., Pasadena, CA 91125, USA, `umans@cs.caltech.edu`.

QSAT$_i$

Given: A Boolean formula $\varphi(x^{(1)}, x^{(2)}, \ldots x^{(i)})$ on i sets of variables, where ϕ is a 3-CNF formula if i is odd and a 3-DNF formula if i is even.

Question: Is it true that $\exists x^{(1)}\ \forall x^{(2)}\ \exists x^{(3)}\ \cdots\ \mathrm{Q}x^{(i)}\ \ \varphi(x^{(1)}, x^{(2)}, \ldots x^{(i)}) = 1$? Here Q stands for "$\exists$" if i is odd and "$\forall$" if i is even.

2 Some classes of complete problems

As can be seen from Part I of this column, the collection of problems known to be complete for the second and higher levels of the PH is almost as diverse (although not nearly as numerous) as the known NP-complete problems. Of course, just as at the NP level, the reductions involved are often somewhat ad-hoc and tailored to the specific problem at hand. However, there are some easily identifiable characteristics that tend to make problems complete for the second and higher levels of the PH; we attempt to elucidate some of them here.

2.1 Succinct representations

In these problems, the input object (e.g., a graph or set system) is represented *succinctly*, often by a Boolean circuit. In the case of a graph $G = (V, E)$, the circuit C typically outputs 1 on input u, v iff $(u, v) \in E$. For set systems $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ with elements in a universe U, the circuit C typically outputs 1 on input u, i iff $u \in S_i$. Examples include SUCCINCT TOURNAMENT REACHABILITY [17] and SUCCINCT SET COVER [19] in the second level and VC DIMENSION [13] in the third level. The reductions involving these problems typically use the circuit representing the input in a *black-box* fashion, and therefore continue to hold with respect to any input representation that is polynomial-time equivalent to Boolean circuits. Wagner investigated a number of means of succinctly representing an input [23], and indeed several problems in the Compendium come from his paper. A more recent example is FULLY COMPRESSED TWO-DIMENSIONAL PATTERN MATCHING [2], in which the input is given by a straight-line program describing the construction of an array of symbols.

It is well known that the succinct versions of certain problems complete for P and NP, are complete for EXP and NEXP, respectively (a phenomenon first noted in [4]). The above examples show that this is not always the case, which raises the question: Is there any formal characterization of problems whose succinct versions are complete for the second and higher levels of the PH? It turns out that although no one has produced a *characterization* of these problems, there is a reliable *source* for problems of this sort: the complexity classes LOGNP and LOGSNP defined by Papadimitriou and Yannakakis [12]. Informally, these complexity classes attempt to capture the complexity of NP problems for which the structure sought has *logarithmic* size in the input length. The class LOGNP contains those languages polynomially reducible to languages L expressible as:

$$L = \{I | \exists S \in [n]^{\log n}\ \forall x \in [n]^p\ \exists y \in [n]^q\ \exists j \in [\log n]\ \varphi(I, S_j, x, y, j)\} \tag{1}$$

while the class LOGSNP contains those languages polynomially reducible to languages L expressible as:

$$L = \{I | \exists S \in [n]^{\log n}\ \forall x \in [n]^p\ \exists j \in [\log n]\ \varphi(I, S_j, x, j)\} \tag{2}$$

where in both definitions φ is a quantifier-free first-order expression involving the input relation symbol I. If the input were to be expressed in an exponentially more succinct form, then it would still be possible to evaluate $\exists j\ \varphi(I, S_j, x, j)$ in polynomial time, and the first three quantifier alternations of LOGNP would become the three quantifier alternations over exponentially large domains that characterize $\mathbf{\Sigma_3^p}$, while the first two alternations of LOGSNP would become the two alternations characterizing $\mathbf{\Sigma_2^p}$. Thus succinct versions of problems complete for LOGNP and LOGSNP are a good source of $\mathbf{\Sigma_3^p}$- and $\mathbf{\Sigma_2^p}$-complete problems, respectively.

As an example of this phenomenon, we consider one of the several natural complete problems for LOGNP and LOGSNP, a variant of VC DIMENSION in which the set system is given explicitly, rather than implicitly (and succinctly) as in the definition in the Compendium.

EXPLICIT VC DIMENSION

Given: A collection $\mathcal{C} = \{S_1, S_2, \ldots, S_n\}$ of subsets of a finite set U, and an integer k.

Question: Is $VC(\mathcal{C}) \geq k$; i.e., is there a set $X \subseteq U$ of size at least k such that for every $S \subseteq X$, there is an i such that $S = S_i \cap X$ (X is *shattered*)?

This variant is LOGNP-complete [12] (and indeed it is the problem that motivated the definition of LOGNP). Schaefer [13] argued that in many settings the succinct version of this problem defined below is more natural.

VC DIMENSION

Given: A collection $\mathcal{C}$ of subsets of a finite set U, represented succinctly by a Boolean circuit C such that $C(i, x) = 1$ if and only if element x is in the i-th set S_i, and an integer k.

Question: Is $VC(\mathcal{C}) \geq k$?

In the following theorem we give a completeness proof that is somewhat simplified from the original. Here (and throughout this paper) $[n]$ denotes the set of integers $\{1, 2, 3, \ldots, n\}$.

Theorem 2.1 (Schaefer [13]) VC DIMENSION *is* $\mathbf{\Sigma_3^p}$*-complete.*

Proof. Let $\varphi(a, b, c)$ be an instance of QSAT$_3$, with $|a| = |b| = |c| = n$. Our universe $U = \{0,1\}^n \times [n]$ has size $n2^n$; for convenience we define $U_a = \{a\} \times [n]$, and notice that the subsets U_a partition U. The sets in our collection $\mathcal{C}$ are labelled with triples (a, b, c). We identify n-bit strings with subsets of $[n]$ and define the set

$$S_{a,b,c} = \begin{cases} \{a\} \times b & \text{if } \varphi(a,b,c) = 1 \\ \emptyset & \text{otherwise.} \end{cases} \tag{3}$$

Clearly there is a small circuit C succinctly representing $\mathcal{C}$ – given an element and a set label, determining whether the element is in the specified set amounts to determining whether the set is empty, which requires only a single evaluation of φ.

Now, if the QSAT$_3$ instance is a positive instance, i.e., $\exists a\ \forall b\ \exists c\ \varphi(a, b, c)$, then the set U_a is shattered, because $\mathcal{C}$ contains sets of the form $\{a\} \times b$ for all b. Thus $VC(\mathcal{C})$ is at least n.

Conversely, if $VC(\mathcal{C})$ is at least n, then there is a set X of size n that is shattered, and we observe that X cannot contain elements from two *different* subsets U_a and $U_{a'}$ (because in particular the set containing these two elements cannot be expressed as this intersection of X with a subset in our collection). Thus $X \subseteq U_a$ for some a, and the fact that it is *shattered* implies that the sets $\{a\} \times b$ for *all* b are present in $\mathcal{C}$, which in turn implies $\forall b\ \exists c\ \varphi(a, b, c)$ and so we have a positive QSAT$_3$ instance. $\diamondsuit$

We will encounter VC DIMENSION again when discussing hardness of approximation in Section 3.2.

2.2 Min-max optimization problems

Succinct representations also play a role in the *min-max optimization problems* investigated by Ko and Lin [8, 9]. At the NP level, a familiar form of optimization problem asks for the size of the *largest* of some polynomial-time-recognizable structure within the input object (usually a graph). A prime example is MAXCLIQUE, in which we want to find the size of the largest clique in the input graph G.

Suppose instead that we are given a *collection* of graphs $G_1, G_2, \ldots, G_n$, and we want to find the largest k such that *for all* input graphs G_t, *there exists* a clique of size at least k in G_t. In other words, we want to find $\min_t \max_C \{|C| : C \text{ is a clique in } G_t\}$. We can already see the two alternations of quantifiers that characterize $\mathbf{\Pi_2^p}$; however for the problem to have a chance at being $\mathbf{\Pi_2^p}$-complete, the quantifiers must be over an exponentially-large domain. This requirement is met if the input is provided implicitly and succinctly by a short description of a family of exponentially many input graphs, as in the following problem, shown to be $\mathbf{\Pi_2^p}$-complete by Ko and Lin [8].

MINMAX CLIQUE

Given: Graph $G = (V, E)$, a partition $(V_{i,j})_{i \in I, j \in J}$ of V, integer k. For a function $t : I \to J$ let f_t be the size of the largest clique in G restricted to $\bigcup_{i \in I} V_{i,t(i)}$.

Question: Is $\min_{t \in J^I} f_t(G) \geq k$?

A word of explanation is in order regarding the family of graphs the input describes. If we imagine the $|I||J|$ subsets of the vertex set V arranged in a square array (with $V_{i,j}$ in position (i, j)), then we are considering all graphs G_t obtained by picking exactly one subset from each row, and taking G_t to be the subgraph of G induced by these vertices. The same input representation is used to define the $\mathbf{\Pi_2^p}$-complete graph problems MINMAX CIRCUIT and MAXMIN VERTEX COVER (notice that since vertex cover is a minimization problem, the min/max order is reversed), and a few other problems that are similar in spirit are considered in the same paper [8].

A second style of succinctly representing a family of graphs is employed by Ko and Lin in defining the min-max optimization problem LONGEST DIRECTED CIRCUIT, which is again $\mathbf{\Pi_2^p}$-complete [9]. Here, the input is a *directed* graph $G = (V, E)$ with a distinguished subset $E' \subseteq E$ of *alterable* edges. This describes the family of all graphs obtained from G by reversing some subset of the alterable edges; the optimization problem asks for the largest k for which all of these graphs contain a directed circuit of length at least k.

Several of the min-max optimization problems in the Compendium (but not all) are known to be hard to approximate to within some constant $c > 0$ [8, 9]; however it is unlikely that these constants are optimal.

2.3 Circuit minimization

For a fixed concrete model of computation $\mathcal{M}$ and a size measure $s : \mathcal{M} \to \mathbb{N}$, we can ask the question: Given $C \in \mathcal{M}$, what is the smallest (according to s) $C' \in \mathcal{M}$ that computes the same function? Typical choices for $\mathcal{M}$ are Boolean circuits, formulas, or DNF formulas. (Other, weaker, models have also been considered, e.g., OBDDs, for which the associated problem is in NP). The associated *circuit minimization* problems have long been cited as prime candidates for natural $\mathbf{\Sigma_2^p}$-complete problems – they certainly have the requisite quantifier alternation: "Does there *exist* C' such that on *all* inputs in the domain, C' and C compute the same value?" Sadly, with the exception of the DNF version, $\mathbf{\Sigma_2^p}$-completeness proofs are not known.

The proof that the DNF version (MIN DNF) is $\mathbf{\Sigma_2^p}$-complete [18] is somewhat involved; however in the remainder of this section we will provide a relatively simple completeness proof for a slightly restricted version, defined below:

IRREDUNDANT
Given: A DNF formula φ and an integer k.
Question: Is there a subset of at most k terms from φ whose disjunction is equivalent to φ?

This problem differs from MIN DNF "only" in the restriction that the terms in the equivalent DNF must come from the original DNF. In the proof that IRREDUNDANT is $\mathbf{\Sigma_2^p}$-complete, we will encounter one other $\mathbf{\Sigma_2^p}$-complete problem: SUCCINCT SET COVER, defined as follows:

SUCCINCT SET COVER
Given: A collection $S = \{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ of 3-DNF formulas on n variables, and an integer k.
Question: Is there a subset $S' \subseteq S$ of size at most k for which $\vee_{\varphi \in S'} \varphi \equiv 1$ (S' is a *cover*)?

We suspect that SUCCINCT SET COVER may play the same role for $\mathbf{\Sigma_2^p}$ that (ordinary) SET COVER plays for NP – the role of a fundamental and natural problem that is a useful starting point for reductions. We first prove that SUCCINCT SET COVER is $\mathbf{\Sigma_2^p}$-complete, and then reduce it to IRREDUNDANT.

Theorem 2.2 (Umans [19]) SUCCINCT SET COVER *is* $\mathbf{\Sigma_2^p}$*-complete.*

Proof. Let $\varphi(a, b)$ be an instance of QSAT$_2$ with $|a| = |b| = n$. We introduce two new sets of variables s, t with $|s| = |t| = n$. Define weight(s, t) to be the number of ones in s and t. We say that the pair (s, t) *encodes* a if for all i, $s_i = a_i$ and $t_i = \neg a_i$. We define the following Boolean function:

$$f(s,t,b) = \begin{cases} 0 & \text{if weight}(s,t) < n \\ 0 & \text{if weight}(s,t) = n \text{ and } (s,t) \text{ does not encode any } a \\ \sigma & \text{if weight}(s,t) = n \text{ and } (s,t) \text{ encodes } a \text{ for which } \varphi(a,b) = \sigma \\ 1 & \text{if weight}(s,t) > n \end{cases} \quad (4)$$

Note that f can be computed by a small circuit C, and that f is *monotone* in s, t – that is, flipping any s_i or t_i from a zero to a one can only increase the value of the function.

We can now define the instance of SUCCINCT SET COVER. Set $m = 2n+1$ and define $\varphi_i = \neg s_i$ and $\varphi_{n+i} = \neg t_i$. Finally, using Cook's Theorem, we produce from C a 3-DNF formula $\varphi_m(s, t, b, w)$ for which $f(s, t, b) = 1 \Leftrightarrow \forall w \; \varphi_m(s, t, b, w) = 1$, where the w variables are new variables.

If $\exists a \, \forall b \; \varphi(a, b)$ then we claim that there is a set cover S' of size $n + 1$; namely, take

$$S' = \{\varphi_i | a_i = 1\} \cup \{\varphi_{n+i} | a_i = 0\} \cup \{\varphi_m\}.$$

Consider any point (s, t, b, w): if (s, t) can be obtained from the (s', t') that encodes a by flipping 0's to 1's, then $\varphi_m(s, t, b, w) = 1$. Otherwise we must have that: for some i for which $a_i = 1$ we have $s_i = 0$ *or* for some i for which $a_i = 0$ we have $t_i = 0$. However in the first case $\varphi_i(s, t, b, w) = 1$ and in the second case $\varphi_{n+i}(s, t, b, w) = 1$. Thus S' covers every point in the domain as required.

Conversely, suppose there is a cover S' of size $n + 1$. We observe that S' must contain φ_m, as this is the only formula that covers the all-ones point. Now, consider the partial assignment obtained by setting $s_i = 1$ if $\varphi_i \in S'$ and zero otherwise, and $t_i = 1$ if $\varphi_{n+i} \in S'$ and zero otherwise. Notice that the only DNF in S' that can possibly cover points (s, t, b, w) extending this partial assignment is φ_m. Therefore we must have $\forall b \, \forall w \; \varphi_m(s, t, b, w) = 1$, and since weight$(s, t) = n$, this implies that (s, t) encodes a for which $\forall b \; \varphi(a, b)$. Thus the QSAT$_2$ instance is a positive instance as required. $\diamond$

Theorem 2.3 (Umans [19]) IRREDUNDANT *is* $\mathbf{\Sigma_2^P}$*-complete.*

Proof. Let $S = \{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ be an instance of SUCCINCT SET COVER. The previous proof shows that we may assume that φ_i consists of a single literal for $1 \leq i < m$, that every cover $S' \subseteq S$ contains φ_m, and that S itself is a cover. Let $t_1, t_2, \ldots, t_n$ be the terms in φ_m. We introduce new variables $z_1, z_2, \ldots, z_n$ and define the DNF for the instance of IRREDUNDANT as follows:

$$\varphi = \bigvee_{i=1}^{n} (z_1 z_2 \ldots z_{i-1} z_{i+1} \ldots z_n t_i) \vee \bigvee_{j=1}^{m-1} (z_1 z_2 \ldots z_n \varphi_j). \tag{5}$$

If $S' \subseteq S$ is a cover of size k, then we claim that the following $k+n-1$ term DNF is equivalent to φ:

$$\varphi' = \bigvee_{i=1}^{n} (z_1 z_2 \ldots z_{i-1} z_{i+1} \ldots z_n t_i) \vee \bigvee_{\varphi_j \in S' \setminus \{\varphi_m\}} (z_1 z_2 \ldots z_n \varphi_j)$$

To see this, note that whenever at least two z variable are 0, φ and φ' are both identically 0; whenever z_i is 0 and the other z variables are 1, φ and φ' are both equivalent to t_i; and when all z variables are 1, both φ and φ' are equivalent to $\vee_{\varphi_i \in S} \varphi_i$ and $\vee_{\varphi_i \in S'} \varphi_i$, respectively. Since S' is a cover, both of these expressions are identically 1, which completes the claim.

Conversely, if there exists a DNF φ' equivalent to φ and consisting of a subset of $k+n-1$ terms of φ, then we claim that there exists a cover of size k. We observe that each of the n original terms involving a t_i must appear in φ', for if φ' omitted the term involving t_i, then φ and φ' would differ on some point with z_i set to 0 and the other z variables set to 1. The remaining $k-1$ terms of φ' each involve some φ_j. Letting S' consist of these φ_j's together with φ_m, we observe that $\vee_{\varphi_j \in S'} \varphi_j$ is equivalent to φ' with all z variables set to 1, which by assumption is equivalent to φ with all z variables set to 1. However, this restriction of φ is just $\vee_{\varphi_j \in S} \varphi_j$, which is identically 1 since S is a cover. Hence S' is a cover of size k, completing the claim. $\diamondsuit$

3 Hardness of Approximation

In this section we turn our attention to results showing that certain optimization problems in the Polynomial-Time Hierarchy are hard to approximate. As usual, we show that an optimization problem is hard to approximate to within a ratio c by reducing a complete problem to the *gap* version of the optimization problem. For example, if the optimization problem in question is a $\mathbf{\Sigma_2^P}$ minimization problem, we show that the problem is hard to approximate to within a factor c by describing a reduction that maps an instance of QSAT$_2$ to an instance whose optimum is at most k if we started with a YES instance, and an instance whose optimum is greater than ck if we started with a NO instance.

It is well known that for many (but not all) problems at the NP level, such a *gap-producing* reduction is equivalent to the existence of *probabilistically checkable proofs* (PCPs), in which a random "local test" can verify the correctness of the proof with high probability. The "local test" property is very constraining and constructing such PCP systems is technically involved. In contrast, for many optimization problems in the second and higher levels of the PH, gap-producing reductions do not appear to require anything resembling PCPs, and they seem to be less complicated. Error-correcting codes and list-decoding algorithms, two objects at the heart of PCP constructions, have been particularly successful tools in these reductions, and in Section 3.2 we will give two examples of gap-producing proofs that use these tools.

3.1 Four ways to turn QSAT into an optimization problem

At the NP level MAXSAT is regarded as the natural optimization version of SAT; in the second and higher levels of the Polynomial-Time Hierarchy, there are a number of natural ways of defining an "optimization version" of the canonical complete problem QSAT_i. We outline four such optimization problems here, together with the associated hardness of approximation results.

The first problem, an optimization version of $\forall\exists$3SAT called MINMAX SAT, appeared in one of the original PH papers [10]. It is $\mathbf{\Pi_2^p}$-complete.

MINMAX SAT
Given: 3-CNF formula $\varphi(x, y)$ and integer k.
Question: For every truth-assignment to x, is there a truth-assignment to y making at least k clauses in $\varphi(x, y)$ true?

Shortly following the PCP characterization of NP, several researchers defined generalizations of PCPs tailored to problems in the Polynomial-Time Hierarchy [3, 7]. These proof systems lead to inapproximability results for MINMAX SAT:

Theorem 3.1 (Ko and Lin [8]) MINMAX SAT *is* $\mathbf{\Pi_2^p}$*-hard to approximate to within some constant* $c > 0$.

One way to define QSAT_2 is to say that $\varphi(a, b)$ is a positive instance if and only if there exists an *implicant* of φ consisting only of the a variables. An implicant of a function f is a set of literals whose conjunction implies f. This suggests a natural way to produce an optimization problem from QSAT_2. The following problem is $\mathbf{\Sigma_2^p}$-complete [22]:

SHORTEST IMPLICANT
Given: A Boolean circuit φ, and an integer k.
Question: Is there an implicant C of φ of size at most k?

A simple argument shows that this problem is in fact $\mathbf{\Sigma_2^p}$-hard to approximate to within very large factors:

Theorem 3.2 (Umans [22]) SHORTEST IMPLICANT *is* $\mathbf{\Sigma_2^p}$*-hard to approximate to within* $n^{1-\varepsilon}$ *for any constant* $\varepsilon > 0$*, where* n *is the size of the circuit.*

A more involved argument shows that the same result holds if φ is a Boolean *formula*. Interestingly, the further restricted variant in which φ is a DNF formula becomes much "easier"– it is complete for a class between **coNP** and $\mathbf{\Sigma_2^p}$ utilizing $\log^2 n$-limited nondeterminism. Both results are in [20].

As we will see in Section 3.2.1 *monotonicity* of the solution set is a key feature of SUCCINCT SET COVER, and indeed many of the other optimization problems in the "logic" section of the Compendium. This inspires the definition of the following two problems, which give minimization and maximization versions of QSAT_2, and are both $\mathbf{\Sigma_2^p}$-complete [19, 21]. These problems seem to be more fruitful starting points for reductions than SHORTEST IMPLICANT.

MONOTONE MINIMUM WEIGHT WORD
Given: A Π_1 nondeterministic circuit C that accepts a nonempty monotone set (although C may contain NOT gates) and an integer k. A Π_1 *nondeterministic circuit* is an ordinary Boolean circuit with two sets of inputs x and y. We say that C *accepts an input* x iff $\forall y\, C(x, y) = 1$. A *monotone set* is a subset S for which $x \in S$ implies $x' \in S$ for all $x' \succeq x$, where $\succeq$ is the bitwise partial order on bitstrings.
Question: Does C accept an input x with at most k ones?

MONOTONE MAXIMUM ZEROS

Given: A Π_1 nondeterministic circuit C that accepts a nonempty monotone set (although C may contain NOT gates) and an integer k.

Question: Does C accept an input x with at least k zeros?

It should be noted that the NP version of MONOTONE MINIMUM WEIGHT WORD has been studied elsewhere [1]; there it is called MONOTONE MINIMUM SATISFYING ASSIGNMENT, and its definition forbids NOT gates in the circuit C (in contrast to the present definitions).

Umans showed that both of these problems are hard to approximate to within very large factors. The proofs use *asymmetric error-correcting codes* in the same manner as we will see in Section 3.2.1.

Theorem 3.3 (Umans; Ta-Shma, Umans and Zuckerman [19, 16]) MONOTONE MINIMUM WEIGHT WORD *is $\mathbf{\Sigma_2^p}$-hard to approximate to within $n^{1-\varepsilon}$ for any constant $\varepsilon > 0$, where n is the size of the circuit.*

Theorem 3.4 (Umans [21]) MONOTONE MAXIMUM ZEROS *is $\mathbf{\Sigma_2^p}$-hard to approximate to within $n^{1/8-\varepsilon}$ for any constant $\varepsilon > 0$, where n is the size of the circuit.*

One can also define generalized versions of these problems in which the circuit C has m sets of inputs, x and $y^{(1)}, y^{(2)}, \ldots, y^{(m-1)}$, and C accepts an input x iff $\forall y^{(1)}\ \exists y^{(2)}\ \forall y^{(3)}\ \ldots\ C(x, y^{(1)}, y^{(2)}, \ldots y^{(m-1)}) = 1$. These are $\mathbf{\Sigma_m^p}$-complete and $\mathbf{\Sigma_m^p}$-hard to approximate to within the same factors as the $m = 2$ variants above [21, 16].

3.2 Two proofs based on list-decoding

In this section we will give complete proofs that SUCCINCT SET COVER and VC DIMENSION are hard to approximate. As mentioned above, both proofs rely on error-correcting codes, and techniques for *list-decoding* them. List-decoding is a relaxation of the usual unique-decoding setting; given a corrupted codeword, the list-decoding procedure produces a short list of candidate codewords, one of which is correct. This relaxation allows one to tolerate a very high error rate, and it has been useful in a wide variety of settings – for a survey, see Chapter 12 of Guruswami's thesis [5].

3.2.1 Asymmetric codes and SUCCINCT SET COVER

We begin with SUCCINCT SET COVER, defined in Section 2.3. One of the salient features of SUCCINCT SET COVER is that its solution set is monotone: that is, any superset of a set cover is also a set cover. The (somewhat unusual) *asymmetric* error-correcting codes that we use in the reduction showing that SUCCINCT SET COVER is hard to approximate are tailored for use in this monotone setting: they tolerate errors that flip zeros to ones but not errors that flip ones to zeros.

In the following $\preceq$ is the bitwise partial order on binary strings; i.e., $x \preceq y$ if y can be obtained from x by flipping zeros to ones. The Hamming weight of a binary string y, denoted weight(y), is the number of ones in y.

Definition 3.5 *An (ℓ, ε)* list-decodable asymmetric code *is a function $C : \{0,1\}^{\bar{k}} \to \{0,1\}^{\bar{n}}$ such that for all $y \in \{0,1\}^{\bar{n}}$ with weight$(y) \le (1-\varepsilon)\bar{n}$, the set L_y defined as:*

$$L_y = \{x | C(x) \preceq y\} \tag{6}$$

has size at most ℓ*. The code is* efficiently encodable *if* C *can be computed in polynomial time, and* efficiently list-decodable *if* L_y *can be computed from* y *in time polynomial in* $|y|$ *and* ℓ*. The* weight *of the code is* $\max\{weight(C(x)) | x \in \{0,1\}^{\bar{k}}\}$.

In Theorem 3.8 we outline a construction of list-decodable asymmetric codes described in [19]. We first need one further definition:

Definition 3.6 *A function* $D : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ *is a* (k, ε) disperser *if for every set* $X \subseteq \{0,1\}^n$ *of size at least* 2^k*, we have*

$$|\{z | \exists x \in X, y \in \{0,1\}^t \;\; D(x,y) = z\}| > (1-\varepsilon)2^m.$$

We call t *the* seed length *and we say that* D *is* explicit *if it can be computed in polynomial time.*

In words, the set X *disperses* if the image of D with its first argument restricted to X hits more than an $(1-\varepsilon)$ fraction of $\{0,1\}^m$. Many constructions of explicit dispersers are known; see [14] for a survey. In this presentation we only need a fairly simple construction of explicit dispersers due to Srinivasan and Zuckerman [15]:

Theorem 3.7 (Srinivasan and Zuckerman [15]) *For every* n*,* $k < n$ *and constant* $\varepsilon > 0$*, there exists a construction of explicit* (k, ε) *dispersers with seed length* $t = 4k + O(\log n)$*, and output length* $m = k + t - O(1)$*.*

Theorem 3.8 (Umans [19]) *One can construct an* $(\ell = 2^k, \varepsilon)$ *list-decodable asymmetric code* $C : \{0,1\}^{\bar{k}} \to \{0,1\}^{\bar{n}=2^m}$ *with weight* $(\bar{k}+1)2^t$ *from any* (k,ε) *disperser* $D : \{0,1\}^{n=\bar{k}+1} \times \{0,1\}^{t=O(\log n)} \to \{0,1\}^m$*. If* D *is explicit, then* C *is efficiently encodable and efficiently list-decodable.*

Proof. Let D be the (k, ε) disperser in the statement of the theorem, and let T be the complete depth-$\bar{k}$ binary tree. We associate the set $\{0,1\}^{\bar{k}}$ with the $2^{\bar{k}}$ leaves of T and the set $\{0,1\}^n \setminus \{0^n\}$ with the $2^n - 1$ vertices of T. Given a leaf w, we let $P(w)$ denote the set of $\bar{k}+1$ vertices on the unique path from the root to w in T.

For a message $w \in \{0,1\}^{\bar{k}}$, we define the associated codeword $C(w)$ as follows:

$$C(w) = \bigvee_{v \in P(w), y \in \{0,1\}^t} e_{D(v,y)}, \tag{7}$$

where e_i is the string in $\{0,1\}^{\bar{n}}$ with one in the i-th position, and zero elsewhere, and we identify $\{0,1\}^m$ with $[\bar{n}]$. Clearly C is computable in polynomial time if D is explicit.

We now argue that C is a $(2^k, \varepsilon)$ list-decodable asymmetric code. For a vertex v in the tree T, define $e_v = \vee_{y \in \{0,1\}^t} e_{D(v,y)}$, and let y be any string in $\{0,1\}^{\bar{n}}$ with weight$(y) \leq (1-\varepsilon)\bar{n}$. We say that v is *active with respect to* y if $e_v \preceq y$. Let A_y be the set of vertices active with respect to y, and observe the following key fact: the associated subset of $\subseteq \{0,1\}^n$ fails to disperse. We immediately conclude that $|A_y| < \ell = 2^k$, and since $|L_y| \leq |A_y|$, C is a $(2^k, \varepsilon)$ list-decodable asymmetric code.

To efficiently compute L_y from y, we perform a depth-first search starting from the root of T, *restricted to the vertices in* A_y, and output the set of leaves reachable from the root. It is clear that for all $w \in L_y$, $P_w \subseteq A_y$, and so w appears in the output. Since D is explicit we can test whether a given vertex is active with respect to y in time polynomial in n, and we examine at most $O(|A_y|) = O(\ell)$ vertices, so the overall running time is polynomial in $n \leq \bar{n}+1$ and ℓ as required. $\diamondsuit$

Corollary 3.9 *For every $\bar{k}$, constant $\varepsilon > 0$, and constant $\delta > 0$, there exists an (ℓ, ε) asymmetric code $C : \{0,1\}^{\bar{k}} \to \{0,1\}^{\bar{n}}$ with $\ell = \bar{k}^{O(1)}$, $\bar{n} = \bar{k}^{O(1)}$, and weight at most $\bar{n}^{4/5+\delta}$. Moreover, C is efficiently decodable and efficiently list-decodable.*

Proof. We use the disperser of Proposition 3.7 with $k = c \log \bar{k}$ (for some large constant c) in Theorem 3.8. As required, $\ell = 2^k = \bar{k}^c$, and $\bar{n} = 2^m \leq \bar{k}^{5c+O(1)}$, and observe that the weight is $(\bar{k}+1)2^t \leq \bar{k}^{4c+O(1)}$, and so by picking c sufficiently large the weight can be made to be at most $\bar{n}^{4/5+\delta}$. ♢

We can now prove that SUCCINCT SET COVER is hard to approximate.

Theorem 3.10 (Umans [19]) SUCCINCT SET COVER *is $\mathbf{\Sigma_2^p}$-hard to approximate to within a factor n^{ε} for some $\varepsilon > 0$.*

Proof. The proof mirrors the completeness proof (Theorem 2.2), using the asymmetric codes just developed in place of the simple encoding used in that proof.

Let $\varphi(a, b)$ be an instance of QSAT$_2$ with $|a| = |b| = n$, and let $\bar{C} : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ be an $(\ell = n^{O(1)}, 1/2)$ list-decodable asymmetric code with weight at most $n^{9/10}$ from Corollary 3.9. We introduce a new set of variables s with $|s| = \bar{n}$, and define the following Boolean function:

$$f(s, b) = \begin{cases} 0 & \text{if weight}(s) \leq \bar{n}/2 \text{ and } \forall a \in L_s \;\; \varphi(a, b) = 0 \\ 1 & \text{if weight}(s) \leq \bar{n}/2 \text{ and } \exists a \in L_s \;\; \varphi(a, b) = 1 \\ 1 & \text{if weight}(s) > \bar{n}/2 \end{cases} \tag{8}$$

Note that f can be computed by a small circuit, and that f is *monotone* in s – that is, flipping any s_i from a zero to a one can only increase the value of the function.

We now describe the instance of SUCCINCT SET COVER. We define $\varphi_i = \neg s_i$, and, using Cook's Theorem, we produce from the small circuit for f a 3-DNF formula $\varphi_{\bar{n}+1}(s, b, w)$ for which $f(s, b) = 1 \Leftrightarrow \forall w \; \varphi_{\bar{n}+1}(s, b, w) = 1$.

If $\exists a \, \forall b \; \varphi(a, b)$ then we claim that there is a set cover S' of size $n^{9/10}$; namely, take

$$S' = \{\varphi_i | C(a)_i = 1\} \cup \{\varphi_{\bar{n}+1}\}.$$

Consider any point (s, b, w): if s can be obtained from $C(a)$ by flipping 0's to 1's, then $\varphi_{\bar{n}+1}(s, b, w) = 1$. Otherwise we must have for some i that $s_i = 0$ while $C(a)_i = 1$, and in this case, $\varphi_i(s, b, w) = 1$. Thus S' covers every point in the domain as required.

Now, suppose there is a cover S' of size $\bar{n}/2$. Consider the partial assignment obtained by setting $s_i = 1$ if $\varphi_i \in S'$ and zero otherwise. Notice that the only DNF in our collection that can possibly cover points (s, b, w) extending this partial assignment is $\varphi_{\bar{n}+1}$. Therefore S' must include $\varphi_{\bar{n}+1}$ and we must have $\forall b \, \forall w \; \varphi_{\bar{n}+1}(s, b, w) = 1$. Since weight$(s) \leq \bar{n}/2$, this implies that $\exists a \in L_s \, \forall b \; \varphi(a, b) = 1$. Thus the QSAT$_2$ instance is a positive instance.

In summary, we have shown that the size of the optimal cover is at most $\bar{n}^{9/10}$ if the QSAT$_2$ instance was a YES instance, and larger than $\bar{n}/2$ if the QSAT$_2$ instance was a NO instance. Since $\bar{n} = n^{O(1)}$, this is a gap of at least n^{ε} for some $\varepsilon > 0$, as required. ♢

With additional ideas from [19] ("self-improvement"), and asymmetric codes built from the dispersers constructed in [16], the hardness of approximation ratio may be improved from n^{ε} to $n^{1-\delta}$ for every constant $\delta > 0$, which is optimal up to lower order terms.

3.2.2 VC DIMENSION

We now show that VC DIMENSION is hard to approximate to within a factor $2-\varepsilon$. The main technical tool is once again an error-correcting code with an efficient list-decoding procedure; here we need only "ordinary" binary codes, not the asymmetric codes of the previous section. In the following definition, $\Delta(x,y)$ denotes the Hamming distance between binary strings x and y, i.e., the number of positions in which x and y differ.

Definition 3.11 *An (ℓ,ε)* list-decodable binary code *is a function $C:\{0,1\}^{\bar{k}} \to \{0,1\}^{\bar{n}}$ such that for all $y \in \{0,1\}^{\bar{n}}$ the set*

$$D_y = \{x | \Delta(C(x),y) \leq (1/2-\varepsilon)\bar{n}\} \tag{9}$$

has size at most ℓ. The code is efficiently encodable *if C can be computed in polynomial time, and* efficiently list-decodable *if D_y can be computed from y in time polynomial in $|y|$ and ℓ.*

Guruswami and Sudan have constructed such codes suitable for the present application:

Theorem 3.12 (Guruswami and Sudan [6]) *For every $\bar{k}$ and constant $\varepsilon > 0$ there exists an $(\ell = O(1), \varepsilon)$ list-decodable binary code with $\bar{n} = O(\bar{k}^2)$, that is efficiently encodable and efficiently list-decodable.*

We can now proceed with the proof that VC DIMENSION is hard to approximate.

Theorem 3.13 (Mossel and Umans [11]) VC DIMENSION *is $\mathbf{\Sigma_3^p}$-hard to approximate to within a factor $2-\varepsilon$, for any constant $\varepsilon > 0$.*

Proof. The proof is similar to the completeness proof (Theorem 2.1).

Let $\varphi(a,b,c)$ be an instance of QSAT$_3$, with $|a|=|b|=|c|=n$, and let $C:\{0,1\}^{\bar{k}=n} \to \{0,1\}^{\bar{n}}$ be an $(\ell = O(1), \delta)$ list-decodable binary code from Theorem 3.12 (for a constant $\delta > 0$ to be specified below).

Our universe $U = \{0,1\}^n \times [\bar{n}]$ has size $\bar{n}2^n$; for convenience we define $U_a = \{a\} \times [\bar{n}]$, and notice that the subsets U_a partition U. The sets in our collection $\mathcal{C}$ are labelled with tuples $(a, y, c^{(1)}, c^{(2)}, \ldots, c^{(\ell)})$. We identify $\bar{n}$-bit strings with subsets of $[\bar{n}]$ and define the set

$$S_{a,y,c^{(1)},c^{(2)},\ldots,c^{(\ell)}} = \begin{cases} \{a\} \times y & \text{if } \forall b \in D_y \; \exists j \in [\ell] \;\; \varphi(a,b,c^{(j)}) = 1 \\ \emptyset & \text{otherwise.} \end{cases} \tag{10}$$

As in the proof of Theorem 2.1 there is a small circuit succinctly representing $\mathcal{C}$, although it is slightly more complicated now. As before, given an element and a set label, determining whether the element is in the specified set amounts to determining whether the set is empty; this determination requires us to compute D_y (which can be done efficiently since C is efficiently list-decodable), and then compute ℓ^2 evaluations of φ.

Now, if the QSAT$_3$ instance is a positive instance, i.e., $\exists a \; \forall b \; \exists c \; \varphi(a,b,c)$, then the set U_a is shattered, because we claim $\mathcal{C}$ contains sets of the form $\{a\} \times y$ for all y. To see this, note that for every y, if $b^{(1)}, b^{(2)}, \ldots, b^{(m)}$ (for $m \leq \ell$) is an enumeration of D_y, then we can pick $c^{(1)}, c^{(2)}, \ldots, c^{(\ell)}$ so that $\varphi(a, b^{(j)}, c^{(j)}) = 1$ for all j, which implies that the set $S_{a,y,c^{(1)},c^{(2)},\ldots,c^{(\ell)}} = \{a\} \times y$. Thus $VC(\mathcal{C})$ is at least $\bar{n}$.

If X is shattered, then as in the proof of Theorem 2.1, we observe that $X \subseteq U_a$ for some a. Now suppose the QSAT$_3$ instance is a negative instance, so $\exists b \; \forall c \;\; \varphi(a,b,c) = 0$. Since X is shattered, our collection includes a set $\{a\} \times y^*$ for which $y^*_i = C(b)_i$ for all $i \in \{i | \{a\} \times \{i\} \in X\}$; note that $\Delta(y^*, C(b)) \leq \bar{n} - |X|$. We also observe that for all y in the set

$$Y = \{y \in \{0,1\}^{\bar{n}} | \Delta(y, C(b)) \leq (1/2 - \delta)\bar{n}\},$$

and all $c^{(1)}, c^{(2)}, \ldots, c^{(\ell)}$, we have $S_{a,y,c^{(1)},c^{(2)},\ldots,c^{(\ell)}} = \emptyset$. This implies that for all $y \in Y$, the set $\{a\} \times y$ is *not* present in our collection. Therefore, $\Delta(y^*, C(b)) > (1/2 - \delta)\bar{n}$, and we conclude that $|X| < (1/2 + \delta)\bar{n}$, and thus $VC(\mathcal{C}) < (1/2 + \delta)\bar{n}$.

Picking δ so that $1/(1/2 + \delta) \geq (2 - \varepsilon)$, we obtain the stated gap: $VC(\mathcal{C}) \geq \bar{n}$ if the QSAT_3 instance was a positive instance, and $VC(\mathcal{C}) < n/(2 - \varepsilon)$ if the QSAT_3 instance was a negative instance. $\diamondsuit$

Mossel and Umans go on to show that there is an Arthur-Merlin protocol that approximates VC DIMENSION to within a factor 2 [11], so the hardness of approximation ratio of $2 - \varepsilon$ is tight.

4 Open problems

There are many open problems in this portion of the PH. The known hardness of approximation ratios for many of the problems in the Compendium are not optimal, and the goal of obtaining optimal ratios for these problems has received far less attention than the analogous program at the NP level. And, while there are few (if any) central problems that one can point to at the NP level for which NP-completeness is not known, there remain important problems in the second level of the PH for which completeness proofs remain stubbornly elusive after nearly 30 years.

References

[1] M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi. Minimum propositional proof length is NP-hard to linearly approximate. In *Mathematical Foundations of Computer Science, 23rd International Symposium (MFCS 98)*, pages 176–184, Brno, Czech Republic, August 1998. Springer-Verlag. To appear in Journal of Symbolic Logic.

[2] P. Berman, M. Karpinski, L. L. Larmore, W. Plandowski, and W. Rytter. On the complexity of pattern matching for highly compressed two-dimensional texts. In A. Apostolico and J. Hein, editors, *Combinatorial Pattern Matching, 8th Annual Symposium*, volume 1264 of *Lecture Notes in Computer Science*, pages 40–51, Aarhus, Denmark, 30 June–2 July 1997. Springer.

[3] A. Condon, J. Feigenbaum, C. Lund, and P. Shor. Probabilistically checkable debate systems and approximation algorithms for PSPACE-hard functions. In *Proc. 25th Ann. ACM Symp. on Theory of Computing*, pages 305–314, 1993.

[4] H. Galperin and A. Wigderson. Succinct representations of graphs. *Information and Control*, 56:183–198, 1983.

[5] V. Guruswami. *List decoding of error-correcting codes*. PhD thesis, M.I.T., 2001.

[6] V. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC)*, 2000.

[7] M. A. Kiwi, C. Lund, A. Russell, D. A. Spielman, and R. Sundaram. Alternation in interaction. In *Structure in Complexity Theory Conference*, pages 294–303, 1994.

[8] K.-I. Ko and C.-L. Lin. On the complexity of min-max optimization problems and their approximation. In D.-Z. Du and P. M. Pardalos, editors, *Minimax and Applications*, pages 219–239. Kluwer Academic Publishers, Boston, 1995.

[9] K.-I. Ko and C.-L. Lin. On the longest circuit in an alterable digraph. *Journal of Global Optimization*, 7:279–295, 1995.

[10] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory*, pages 125–129, The University of Maryland, 25–27 Oct. 1972. IEEE.

[11] E. Mossel and C. Umans. On the complexity of approximating the VC dimension. In *Proceedings of the 16th Annual Conference on Computational Complexity*, pages 220–225. IEEE Computer Society Press, 2001.

[12] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V–C dimension. *Journal of Computer and System Sciences*, 53(2):161–70, 1996.

[13] M. Schäfer. Deciding the Vapnik-Cervonenkis dimension is Σ_3^P-complete. *Journal of Computer and System Sciences*, 58:177–182, 1999.

[14] R. Shaltiel. Recent developments in explicit constructions of extractors. *BEATCS Computational Complexity Column*, 77, June 2002.

[15] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999.

[16] A. Ta-Shma, C. Umans, and D. Zuckerman. Loss-less condensers, unbalanced expanders, and extractors. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 143–152. ACM, 2001.

[17] T. Tantau. A note on the complexity of the reachability problem for tournaments. Technical Report TR01-092, ECCC, Nov. 2001.

[18] C. Umans. The Minimum Equivalent DNF Problem and Shortest Implicants. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 556–563, 1998.

[19] C. Umans. Hardness of Approximating Σ_2^p Minimization Problems. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 465–474, 1999.

[20] C. Umans. On the complexity and inapproximability of shortest implicant problems. In *Automata, Languages and Programming, 26th International Colloquium (ICALP 99)*, pages 687–696, July 1999.

[21] C. Umans. *Approximability and Completeness in the Polynomial Hierarchy.* PhD thesis, U.C. Berkeley, 2000.

[22] C. Umans. The Minimum Equivalent DNF Problem and Shortest Implicants. *Journal of Computer and System Sciences*, 63(4):597–611, Dec. 2001.

[23] K. W. Wagner. The Complexity of Combinatorial Problems with Succinct Input Representation. *Acta Informatica*, 23(3):325–356, 1986.