The More the Merrier! On Total Coding and Lattice Problems and the Complexity of Finding Multicollisions*

Huck Bennett
University of Colorado Boulder, CO, USA

Surendra Ghentiyala
Cornell University, Ithaca, NY, USA

Noah Stephens-Davidowitz
Cornell University, Ithaca, NY, USA

Abstract

We show a number of connections between two types of search problems: (1) the problem of finding an L-wise *multicollision* in the output of a function; and (2) the problem of finding two codewords in a code (or two vectors in a lattice) that are within distance d of each other. Specifically, we study these problems in the *total* regime, in which L and d are chosen so that such a solution is guaranteed to exist, though it might be hard to find.

In more detail, we study the total search problem in which the input is a function $\mathcal{C} : [A] \to [B]$ (represented as a circuit) and the goal is to find $L \leq \lceil A/B \rceil$ *distinct* elements $x_1, \ldots, x_L \in A$ such that $\mathcal{C}(x_1) = \cdots = \mathcal{C}(x_L)$. The associated complexity classes Polynomial Multi-Pigeonhole Principle ((A, B)-PMPP^L) consist of all problems that reduce to this problem.

We show close connections between (A, B)-PMPP^L and many celebrated upper bounds on the minimum distance of a code or lattice (and on the list-decoding radius). In particular, we show that the associated computational problems (i.e., the problem of finding two distinct codewords or lattice points that are close to each other) are in (A, B)-PMPP^L, with a more-or-less smooth tradeoff between the distance d and the parameters A, B, and L. These connections are particularly rich in the case of codes, in which case we show that multiple incomparable bounds on the minimum distance lie in seemingly incomparable complexity classes.

Surprisingly, we also show that the computational problems associated with some bounds on the minimum distance of codes are actually *hard* for these classes (for codes represented by arbitrary circuits). In fact, we show that finding two vectors within a certain distance d is actually hard for the important (and well-studied) class $\mathsf{PWPP} = (B^2, B)$-PMPP^2 in essentially all parameter regimes for which an efficient algorithm is not known, so that our hardness results are essentially tight. In fact, for some d (depending on the block length, message length, and alphabet size), we obtain both hardness and containment. We therefore completely settle the complexity of this problem for such parameters and add coding problems to the short list of problems known to be complete for PWPP.

We also study (A, B)-PMPP^L as an interesting family of complexity classes in its own right, and we uncover a rich structure. Specifically, we use recent techniques from the cryptographic literature on multicollision-resistant hash functions to (1) show inclusions of the form (A, B)-$\mathsf{PMPP}^L \subseteq (A', B')$-$\mathsf{PMPP}^{L'}$ for certain non-trivial parameters; (2) black-box separations between such classes in different parameter regimes; and (3) a non-black-box proof that (A, B)-$\mathsf{PMPP}^L \in \mathsf{FP}$ if (A', B')-$\mathsf{PMPP}^{L'} \in \mathsf{FP}$ for yet another parameter regime. We also show that (A, B)-PMPP^L lies in the recently introduced complexity class Polynomial Long Choice for some parameters.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

* An earlier version of this work used the title "The more the merrier! On the complexity of finding multicollisions, with connections to codes and lattices."

16th Innovations in Theoretical Computer Science Conference (ITCS 2025).
Editor: Raghu Meka; Article No. 14; pp. 14:1–14:22

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Multicollisions, Error-correcting codes, Lattices

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.14

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2024/018/ [4]

Funding *Huck Bennett*: This work is supported in part by NSF Grant No. 2312297. Part of this work was done while the author was at Oregon State University.
Surendra Ghentiyala: This work is supported in part by the NSF under Grants Nos. CCF-2122230 and CCF-2312296, a Packard Foundation Fellowship, and a generous gift from Google.
Noah Stephens-Davidowitz: This work is supported in part by the NSF under Grants Nos. CCF-2122230 and CCF-2312296, a Packard Foundation Fellowship, and a generous gift from Google. Some of this work was completed while the author was visiting the National University of Singapore and the Centre for Quantum Technologies.

Acknowledgements The authors would like to thank Atri Rudra for very helpful discussions.

1 Introduction

We study *the complexity of total coding problems*, as well as total lattice problems. (We focus on coding problems in the introduction.)

Recall that a q-ary code with messages of length k, block length n, and distance d is a function $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ such that

$$d = \min_{\boldsymbol{x}_1 \neq \boldsymbol{x}_2} \Delta(\mathcal{C}(\boldsymbol{x}_1), \mathcal{C}(\boldsymbol{x}_2)) \ ,$$

where Δ is the Hamming distance, i.e., the number of entries in which two elements in $\mathbb{F}_q^n$ differ. For our purposes, we think of $\mathcal{C}$ as an arbitrary circuit with size $\mathrm{poly}(n, k, \log q)$. (Much of the literature is concerned with the special case when $\mathcal{C}$ is a non-singular *linear* function, in which case the code is called a *linear* code. We discuss our choice of definition more in Section 1.5.)

The most fundamental question in coding theory is to find bounds on d for fixed n, k, and q. In other words, one wishes to argue that any set of q^k points in $\mathbb{F}_q^n$ must contain two points that are within Hamming distance d for the smallest value of d possible. Many beautiful upper bounds on the distance d are known in terms of n, k, and q.

We therefore define the natural family of computational search problem associated with this question. Specifically, we define the $(n, k, d)_q$-Short Distance Problem ($(n, k, d)_q$-SDP) as the computational search problem in which the input is a circuit $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ and the goal is to find $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ such that $\Delta(\mathcal{C}(\boldsymbol{x}_1), \mathcal{C}(\boldsymbol{x}_2)) \leq d$.[1] Notice that $(n, k, d)_q$-SDP is total if and only if all q-ary codes with message length k and block length n have distance at most d.

So, upper bounds on the minimum distance of a code correspond to proofs of *totality* of SDP for some choice of $d = d(n, k, q)$. In other words, such bounds can be viewed as proofs that for such a choice of d, SDP is in the complexity class TFNP, which consists of *total* search problems with efficiently verifiable solutions. Starting with the seminal work of

[1] The problem of finding $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ that minimizes $\Delta(\mathcal{C}(\boldsymbol{x}_1), \mathcal{C}(\boldsymbol{x}_2))$ for the given input $\mathcal{C}$ is called the Minimum Distance Problem (MDP), and it is known to be NP-hard, even to approximate, and even for linear codes [14]. In contrast, SDP only asks for two codewords within some fixed distance, independent of the minimum distance of the code. Furthermore, one is often interested only in the important special case in which $\mathcal{C}$ computes a linear function over $\mathbb{F}_q$, but we stick to this more general setting because our upper bounds for this problem apply even to arbitrary codes (making them stronger), and unfortunately our lower bounds seem to require us to work with arbitrary codes specified by circuits.

Papadimitriou [34], the study of TFNP has focused largely on placing important total search problems in subclasses of TFNP, where it is often convenient to think of each subclass as capturing problems whose totality can be attributed to a particular fundamental principle. E.g., PPP (which we will discuss much more shortly) can be thought of as the subclass of TFNP corresponding to problems whose totality can be proven via a particular version of the pigeonhole principle.

It is therefore natural to ask what more we can say about the complexity of $(n, k, d)_q$-SDP when $d = d(n, k, q)$ is equal to one of the many celebrated upper bounds on the minimum distance of a q-ary code with block length n and message length k. Indeed, there are now many important upper bounds known on the minimum distance of codes (in terms of n, k, and q), such as the Singleton bound [39], the Hamming bound [19], the Plotkin bound [36], the Elias-Bassalygo bound [3], and the MRRW bound(s) [29]. These bounds are ubiquitous in the computer science literature and beyond. (See, e.g., [18].)

However, in spite of the importance of these bounds, there has been surprisingly little work addressing the complexity of the associated (total) search problems of actually finding a pair of distinct codewords within one of these bounds. Indeed, to our knowledge, the only prior work directly addressing such a question is the beautiful (and recent) work of Debris-Alazard, Ducas, and van Woerden [11], which showed (among other things) that the problem of solving SDP on linear codes within the Griesmer bound [16] can be solved efficiently.

1.1 The problem of finding multicollisions

In this work, we will show connections between the complexity of such total coding (and lattice) problems and a natural family of total search problems.

Specifically, for integers $A > B$ and $L \geq 2$, we consider the computational problem in which the input is a circuit $\mathcal{C} : [A] \to [B]$,[2] and the goal is to find L distinct input values $x_1, x_2, \ldots, x_L$ such that $\mathcal{C}(x_1) = \mathcal{C}(x_2) = \cdots = \mathcal{C}(x_L)$. Notice that, by the (generalized) pigeonhole principle, this problem is total if (and only if) the size A of the domain of $\mathcal{C}$ and the size B of its range satisfy $L \leq \lceil A/B \rceil$. We will focus on the regime in which the problem is total.

In the special case when $L = 2$ and $A = B + 1$, this is the canonical complete problem for the Polynomial Pigeonhole Principle complexity class (PPP).[3] This class was introduced by Papadimitriou in his work studying problems in TFNP (i.e., total search problems in FNP) [34]. Since then, the class PPP has been of great interest because it is known to contain many important computational problems, such as the problem of breaking a cryptographic collision-resistant hash function, the problem of breaking a one-way permutation, factoring (under randomized reductions) [22], and many more problems of interest [41, 8]. Most relevantly to our work, the problem of finding a non-zero vector in a lattice within Minkowski's bound (in the ℓ_∞ norm, though see below) was shown to be in PPP in [2].

Because of its relationship with the pigeonhole principle, we call the computational problem of finding such a collision Pigeon. (Papadimitriou originally used this name for the special case when $L = 2$ [34].) In fact, we get a family of problems (A, B)-Pigeon^L,

[2] See the full version [4] for discussion of what we mean by a circuit with input size A and output size B when A and B are not necessarily powers of 2.

[3] When $L = 2$ and $A/B \geq 1 + 1/\operatorname{poly}(\log B)$, one obtains the canonical complete problem for Polynomial Weak Pigeonhole Principle complexity class (PWPP). We will say much more about this distinction later, but for now we largely ignore this.

parameterized by the circuit input size A, the circuit output size B, and the number of colliding inputs $L \leq \lceil A/B \rceil$ that we must find. Accordingly, we define the complexity class (A, B)-PMPP^L as the set of all search problems that have a polynomial-time (Karp) reduction to (A, B)-Pigeon^L.[4]

In the complexity-theoretic literature, there is very little work on Pigeon for $L > 2$ (although we note Sotiraki's thesis [40, Section 4.5] as foundational work in this direction; and see also Section 1.3 for discussion of recent independent work by [21]). On the other hand, there is an exciting line of work in the cryptographic literature studying *multicollision-resistant hash functions* [23, 33, 43, 5, 28, 6, 25, 40, 13, 37, 20]. From our perspective, one can think of such cryptographic works as studying the *average-case* complexity of the Pigeon problem (under efficiently sampleable distributions of circuits $\mathcal{C}$). However, even these works have not fully explored this space, primarily focusing on the case where L is constant and $\log A \gtrsim 2 \log B$.

1.2 Our results

In this work, we show many connections between coding and lattice problems and PMPP. We also further study the classes (A, B)-PMPP^L.

1.2.1 Connections between coding problems and PMPP

Our first set of results shows a number of connections between PMPP and computational search problems related to error-correcting codes. (See Figure 1 for a summary of some of our results for binary codes.)

1.2.1.1 Upper bounds on the Shortest Distance Problem

We show that many of the celebrated bounds on the minimum distance of a code yield versions of SDP that are in different versions of PMPP, including the Singleton bound [39], the Hamming bound [19], the Plotkin bound [36], and the Elias-Bassalygo bound [3]. The following theorem shows some of these results.

▶ **Theorem 1** (Informal, see full version [4]). *The following hold for any n, k, and prime power q and any constant $\varepsilon > 0$.*

1. (n, k, d_S)-SDP$_q$ *is in* PWPP *for* $d_S(n, k, q) := n - k + 1$ *equal to the Singleton bound.*
2. (n, k, d_H)-SDP$_q$ *is in* PPP *and* $(n, k, (1 + \varepsilon)d_H)$-SDP$_q$ *is in* PWPP *for* $d_H = d_H(n, k, q)$ *equal to the Hamming bound.*
3. (n, k, d_P)-SDP$_q$ *is in* PMPP *for* $d_P(n, k, q) = (1 - 1/q + \varepsilon)(n - k)$ *slightly larger than the Plotkin bound.*
4. (n, k, d_E)-SDP$_q$ *is in* PMPP *for* $d_E(n, k, q)$ *equal to the Elias-Bassalygo bound.*

In fact, these results are special cases of more general results that we prove. Specifically, we show that one can obtain smaller values of d by increasing L *or* decreasing A (while holding B fixed and maintaining totality).

Our smooth tradeoff in particular is perhaps a bit surprising. Indeed, it is a-priori not clear why it would be useful to take $L > 2$ in such a reduction, since, after all, the goal in SDP is to find *two* codewords. (Of course, this is because our reductions mimic the proofs of

[4] To define PMPP formally, A and B should be functions of some asymptotic parameter n, and L might also be a function of n. But, we mostly ignore this issue in the introduction for simplicity.

the Plotkin and Elias-Bassalygo bounds. Both proofs work by first finding $L > 2$ codewords that lie in a relatively small Hamming ball and then showing that two of these codewords must be quite close to each other.)

1.2.1.2 Tight hardness for SDP and completeness when $L = 2$

We next show tight hardness results for SDP. In particular, we consider both the case when a code is represented by an arbitrary circuit $\mathcal{C}$ *and* the case when a code is represented in systematic form (i.e., the case when the first k symbols in a codeword consist of the message; see Section 1.5). We call this restricted problem sysSDP.

▶ **Theorem 2** (Informal, see full version [4]). *For any n, k, and prime power q and any constant $\varepsilon > 0$, (n, k, d)-SDP$_q$ is* PWPP*-hard for $d(n, k, q) = (1 - 1/q - \varepsilon)n$. Furthermore, (n, k, d_P)-sysSDP$_q$ is* PWPP*-hard for distance $d_P(n, k, q) = (1-1/q-\varepsilon)(n-k)$ slightly below the Plotkin bound.*

Theorem 2 is essentially tight. In particular, for $d \geq (1 - 1/q)n$, it is easy to see that SDP can be solved efficiently. And, we show that there is also a simple efficient algorithm for sysSDP for $d \gtrsim (1-1/q)(n-k)$ (see full version [4]). So, Theorem 2 essentially characterizes when there is a polynomial-time algorithm for SDP and sysSDP (assuming that FP $\neq$ PWPP).

Furthermore, there is a large overlap between the parameter regime for which we show containment in PWPP and the regime for which we show PWPP-hardness (both for arbitrary codes and codes in systematic form), so that for a large range of parameters we show that SDP is actually *complete* for PWPP. This essentially settles the complexity of SDP in these regimes and adds SDP to the relatively short list of problems known to be complete for this class. For example, we show that in many settings, the Hamming bound is PWPP-complete. (Since the Hamming bound can be viewed as a coding-theoretic analogue of Minkowski's bound for lattices, one might view this result as resolving a coding-theoretic analogue of the conjecture that Minkowski's bound is PPP-complete [2]. However, perhaps a better coding-theoretic analogue would be PPP-hardness of this problem over *linear* codes, which we do not prove. See Section 1.6.)

We summarize these results for binary codes in Figure 1. The picture for q-ary codes is quite similar (with the various bounds replaced by their q-ary versions). See full version [4] for the details.

1.2.1.3 Finding *many* close codewords and PMPP

We also show analogous results for the problem of finding *many* codewords that all lie in a small ball. This problem corresponds to bounds on the *list decoding* radius of a code (just as the problem of finding close pairs of codewords corresponds to bounds on the *unique decoding* radius).

We first show that list-decoding generalizations of the Singleton bound and the Hamming bound lie in PMPP for appropriate parameters. (See full version [4]) We then show that the problem of finding many codewords that all lie in a small ball is also *hard* for PMPP for some choices of parameters.

Our results for this problem are significantly more subtle than the $L = 2$ case, since for $L > 2$, the complexity of (A, B)-Pigeon^L seems to vary quite a bit with the parameters, whereas for $L = 2$, the parameters A and B matter much less. In particular, for $L > 2$ we do not find a complete problem for PMPP^L. See full version [4] for the details.

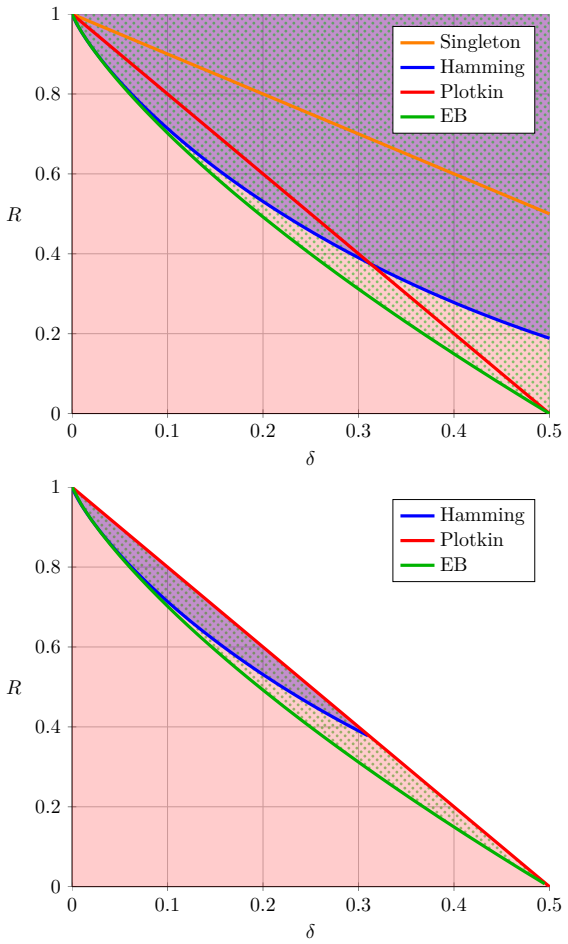

Figure 1 Two plots showing the complexity of the problem of finding two codewords within relative distance $\delta := d/n$ in a code with rate $R := k/n$ for binary codes (represented by circuits). The shaded red region represents PWPP-hardness. (The entirety of both figures are shaded red.) Regions not shown ($\delta > 1/2$ for the left figure and $\delta > 1 - 2R$ for the right figure) are known to be in FP. The shaded blue region represents containment in PWPP. The region of overlap between red and blue therefore represents regimes where the problem is PWPP-complete. The area covered with green dots represents problems in PMPP^L for some $L \geq 2$. The left figure is for arbitrary (binary) codes represented by arbitrary circuits, while the right figure is for (binary) codes in systematic form. (See Section 1.5 for discussion of systematic form and how we define codes in this context.)

1.2.2 Lattice problems in PMPP

We next show similar results for computational lattice problems. Recall that a lattice is the set of all integer linear combinations of n linearly independent basis vectors $\mathbf{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{Z}^n$, i.e.,

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{z_1\boldsymbol{b}_1 + \cdots + z_n\boldsymbol{b}_n \ : \ z_i \in \mathbb{Z}\} \ .$$

A fundamental question about lattices asks when there must exist a non-zero lattice point $\boldsymbol{y} \in \mathcal{L}_{\neq \mathbf{0}}$ such that $\|\boldsymbol{y}\|_K \leq r$, where $\|\cdot\|_K$ is some norm of interest and r is some bound.

Minkowski's celebrated theorem [32] tells us that such a $\boldsymbol{y}$ is guaranteed to exist for some

$$r \leq 2\det(\mathcal{L})^{1/n}/\operatorname{vol}(K)^{1/n} ,$$

where $\det(\mathcal{L}) := |\det(\mathbf{B})|$ is the lattice determinant and K is the unit ball of the norm $\|\cdot\|_K$. The corresponding computational problem Minkowski_K is the search problem in which the input is a basis $\mathbf{B}$ for a lattice, and the goal is to output a non-zero vector $\boldsymbol{y} \in \mathcal{L}_{\neq \mathbf{0}}$ such that

$$\|\boldsymbol{y}\|_K \leq 2\det(\mathcal{L})^{1/n}/\operatorname{vol}(K)^{1/n} .$$

This problem is quite important in cryptography, particularly in the ℓ_2 norm. In the special case of the ℓ_∞ norm, it is known that $\text{Minkowski}_\infty \in \mathsf{PPP}$, and Ban, Jain, Papadimitriou, Psomas, and Rubinstein [2] conjectured that Minkowski_∞ is actually PPP-complete. (This conjecture remains open.)

We study the more general problem of finding $\boldsymbol{y} \in \mathcal{L}_{\neq \mathbf{0}}$ with

$$\|\boldsymbol{y}\|_p \leq \gamma \det(\mathcal{L})^{1/n} ,$$

where

$$\|\boldsymbol{y}\|_p := (|y_1|^p + \cdots + |y_n|^p)^{1/p}$$

is the ℓ_p norm. This problem is known as the γ-Hermite Shortest Vector Problem (γ-HSVP_p). Since this problem is relevant to cryptography, it is very well studied for a wide range of parameters γ [27, 38, 15, 31, 1], particularly for $p = 2$.

The case $\gamma = 2/\operatorname{vol}(\mathcal{B}_p^n)^{1/n}$ corresponds to Minkowski_p, where $\mathcal{B}_p^n$ is the unit ℓ_p ball. However, Minkowski's bound is not tight (except in the case when K tiles space, such as when K is the hypercube). For example, Blichfeldt improved on Minkowski's celebrated theorem in the ℓ_2 norm, proving that γ-HSVP_2 is still total when $\gamma \approx \sqrt{2}/\operatorname{vol}(\mathcal{B}_2^n)^{1/n}$ [7].[5]

Perhaps surprisingly, we show that γ-$\text{HSVP}_2 \in (A, B)$-PMPP^L for $\gamma \approx \sqrt{2}/\operatorname{vol}(\mathcal{B}_2^n)^{1/n}$ and appropriate choices of A, B, and L. In fact, we show a smooth tradeoff between the Hermite factor γ and the parameters A, B, and L, showing that one can obtain shorter vectors by either increasing L or decreasing the ratio between A and B (while maintaining totality). A similar story holds for ℓ_p norms more generally.

▶ **Theorem 3** (Informal; see full version [4]). *For any constant integer $p \geq 1$ and any $A, B, L \in \mathbb{Z}^+$ satisfying $2^{s(n)} = B < A \leq 2^{\text{poly}(n)}$ and $L \leq \lceil A/B \rceil$ for a sufficiently large polynomial $s(n)$, it holds that γ-$\text{HSVP}_p \in (A, B)$-PMPP^L for*

$$\gamma \approx d_p(L, n) \cdot (A/B)^{1/n} \cdot \operatorname{vol}(\mathcal{B}_p^n)^{-1/n} ,$$

where n is the rank of the lattice in the γ-HSVP instance and $1 \leq d_p(L, n) \leq 2$ is a particular function that is decreasing in the parameter L.

[5] We note that Blichfeldt proved two distinct relevant theorems in this context, which might easily confuse the reader. So, we endeavor to clarify here. The theorem commonly referred to as "Blichfeldt's theorem" says that any measurable set $S \subset \mathbb{R}^n$ with $\operatorname{vol}(S) > \det(\mathcal{L})$ is guaranteed to contain two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in S$ with $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ such that $\boldsymbol{x}_1 - \boldsymbol{x}_2 \in \mathcal{L}$. This theorem is often discussed in the context of total lattice problems because it can be used to prove Minkowski's theorem. In fact, Sotiraki, Zampetakis, and Zirdelis introduced a related computational problem that they called BLICHFELDT (in which the set S is represented implicitly by a circuit), and they showed that BLICHFELDT is actually PPP-complete. It is not clear how BLICHFELDT is related to γ-HSVP for $\gamma \approx \sqrt{2}/\operatorname{vol}(\mathcal{B}_2^n)^{1/n}$ (except that they are two computational lattice problems whose totality was proven by Blichfeldt).

1.2.3 Containments between different classes

The above results bring new attention to the classes PMPP. So, we next study containments between these PMPP classes with different parameters A, B, and L, and other classes of interest in TFNP.

Recall that the celebrated Merkle-Damgård construction [30, 10] shows that the ratio of the input size A to the output size B of a circuit essentially does not matter in the special case when $L = 2$, since one can efficiently reduce from the problem of finding a single collision (i.e., $L = 2$) in a barely compressing circuit $\mathcal{C} : A \to B$ with $A = B + B/\operatorname{poly}(\log B)$ to the (seemingly much easier) problem of finding a single collision in a much more compressing circuit $\mathcal{C}' : A' \to B$ with $\log A' = \log(B)^C$ for any constant $C > 1$. In our terminology,

$$(B + B/\operatorname{poly}(\log B), B)\text{-PMPP}^2 = (2^{\log^C B}, B)\text{-PMPP}^2 .$$

This surprising collapse of complexity classes is known as *domain extension*, and it has innumerable applications in cryptography and complexity theory.[6]

Already in 2004, Joux noticed that Merkle and Damgård's elegant domain extension technique does not seem to work for $L > 2$ [23]. So, it appears that for $L > 2$, the relationship between A and B (i.e., how "compressing" the circuit $\mathcal{C}$ is) might matter quite a bit. This suggests a surprising fundamental difference between the case $L = 2$ and the case when $L > 2$.

However, we show two (rather weak) notions of domain extension that still work in the setting of multicollisions. The first such result follows by analyzing the Merkle-Damgård construction in this setting and showing that it does achieve *something*, albeit with a large loss in parameters. The second result shows a more sophisticated reduction (using Merkle trees together with list-recoverable codes) that is better than the Merkle-Damgård-based reduction in the regime where A is very large compared to B. The latter result can be thought of as a translation into our setting of a beautiful result due to Bitanski, Kalai, and Paneth [6] in the setting of multicollision-resistant hash functions. (In fact, the proof is substantially simpler in our setting than in that of [6], since we do not have to worry about the many additional issues that arise in the average-case setting.)

Together, these results show that it is possible to reduce the problem of finding multicollisions in a less compressing function to the problem of finding multicollisions in a more compressing function, but at the expense of a large loss in the number of collisions found.

▶ **Theorem 4** (Informal; see full version [4]). *For $m > a > b$,*

$$(2^a, 2^b)\text{-PMPP}^{L'} \subseteq (2^m, 2^b)\text{-PMPP}^L$$

for $L' \approx L^{(a-b)/(m-b)}$.

For any $r \geq 2$, any $k \geq 1$, and any L,

$$(2^{vr}, 2^v)\text{-PMPP}^{M'} \subseteq (2^{vrk}, 2^v)\text{-PMPP}^M$$

under randomized reductions, for

$$M' \approx M^{\log r/(2\log r + \log k + \log(M)/2)} .$$

[6] Admittedly, we are deliberately conflating the distinction between the problem of breaking a cryptographic hash function (which is what Merkle and Damgård actually studied) and the problem of finding a collision in an arbitrary worst-case circuit. All of the above statements hold in both cases.

We also show that the class PMPP is contained in the recently introduced ([35]) complexity class Polynomial Long Choice (PLC), for appropriate choices of the parameters A, B, and L. In fact, we show a reduction to the Unary Long Choice problem. (This result was recently independently discovered in [21]. See Section 1.3.) This strengthens a result of Pasarkar, Papadimitriou, and Yannakakis [35], who showed that PWPP $\subseteq$ PLC.

▶ **Theorem 5** (Informal; see full version [4]). *For any $L < n$,*

$(2^n, 2^{n-L})$-PMPP$^L \subseteq$ PLC

1.2.4 Black-box separations (and a non-black-box non-separation)

Our final set of results concerns black-box separations between (A, B)-PMPP^L for different values of A, B, and L, which suggest that it might be hard to prove stronger containments than what we show above. However, we note that recent independent work of Jain, Li, Robere, and Xun [21] also showed exciting black-box separations of this form. While our results are formally incomparable to theirs, we believe that the results of [21] are more interesting than our own. (See Section 1.3.)

The starting point for our results is a beautiful idea due to Komargodski, Naor, and Yogev [25] for separating $(2^n, 2^{n/2})$-PMPP^L from $(2^n, 2^{n/2})$-PMPP$^{L'}$ for any constants $L \neq L'$ (particularly in the average-case setting relevant to cryptography). Unfortunately, however, the proof in [25] had a subtle bug that does not seem easy to fix [26].

We use similar ideas to show two different black-box separations, which can be seen as partial evidence that domain extension and range compression are not possible when $L > 2$. However, we note that our black-box separations are rather weak, since they only rule out rather fine-grained black-box reductions between the classes.

Finally, we show that a very clever non-black-box proof due to Rothblum and Vasudevan [37] extends to our setting. In particular, Rothblum and Vasudevan show, using non-black-box techniques, that the existence of a certain sufficiently strong multicollision-resistant hash function implies the existence of a collision-resistant hash function. We prove an analogue of their result in our (worst-case) setting, showing that suitable hardness of PMPP for large L implies hardness of PMPP for smaller L. (After a preliminary version of this work appeared, Buzek and Tessaro [9] improved upon the main result in [37]. We do not know whether the stronger result in [9] extends to our setting.)

As these results are rather technical, we refer the reader to the full version [4] for the details.

1.3 Comparison with Jain, Li, Robere, and Xun

Recent exciting work by Jain, Li, Robere, and Xun [21] also defines and studies the computational problem (A, B)-Pigeon^L and the associated complexity class (A, B)-PMPP^L (with slightly different notation). (They also define additional classes that correspond to the union of (A, B)-PMPP^L over different parameters A, B, and L.) [21] is concurrent with and independent of this work (and appeared as a preprint shortly before we released a preprint of the present work). Here, we provide a brief comparison of their work with ours.

Both the present work and [21] define multi-collision classes and study relationships between them. At a high level, [21] has morally stronger (although formally incomparable) results on the structural complexity of these classes, whereas the present work focuses on showing containment and hardness of coding and lattice problems with respect to these classes (which [21] does not study at all).

In terms of structural complexity, [21] contains exciting black-box separation results, which, although formally incomparable to our black-box separations, we think of as stronger. In particular, [21] are essentially able to black-box separate (A, B)-PMPP^L from (A', B')-PMPP$^{L'}$ for *any* constants $L \neq L'$ and *any* (reasonable) A, B, A', and B'. They also show black-box separations between PMPP and other interesting complexity classes in TFNP, and in particular show a black-box separation between the Ramsey problem and PMPP. We refer the reader to [21] for the technical details.

[21] also studies the relationship between PLC and PMPP. Indeed, they prove a result that is essentially identical to our Theorem 5, which shows that PMPP $\subseteq$ PLC for certain parameters. (Formally, our technical result in the full version [4] is more general than the analogous result in [21], but it is clear that the proof in [21] yields the more general result as well.) In addition, they show a containment in the other direction, that PLC $\subseteq$ PMPP, albeit for different parameters. Finally, [21] shows that an interesting problem in TFBQP is in PMPP. The latter two results have no analogue in the present work.

In this work, we focus on the relationship of PMPP with coding and lattice problems (see the full version [4]). We also show Merkle-Damgård-style reductions between PMPP with different parameters (see full version [4]) and a non-block-box non-separation in the style of [37] (see full version [4]). These topics are not studied in [21].

1.4 Other related work

Our work lies at the intersection of a number of different areas, and there is therefore much related work to discuss in addition to [21]. Here, we focus on how this prior work relates to our work.

The complexity of total lattice problems

The complexity of Minkowski's bound and HSVP more generally is quite well studied, particularly in the ℓ_∞ norm and the ℓ_2 norm. In particular, algorithms for HSVP_2 play a very important role in lattice-based cryptography, and algorithms for HSVP_2 with different approximation factors are a very active area of research [27, 38, 15, 31, 1]. Some of these algorithms can be viewed as constructive proofs of classical results about the minimum distance of a lattice relative to the determinant. (For example, the celebrated LLL algorithm gives a constructive proof of Hermite's bound [27], and the slide reduction algorithm gives a constructive proof of Mordell's inequality [15].)

Finding a vector within Minkowski's bound in the ℓ_∞ norm is considered one of the most important problems in the complexity class PPP. In particular, Ban, Jain, Papadimitriou, Psomas, and Rubinstein showed that other important problems in PPP can be reduced to this problem, and they conjectured that this problem is actually PPP-complete [2]. Sotiraki, Zampetakis, and Zirdelis further investigated the relationship between lattice problems, PPP, and PWPP, showing two problems related to lattices that are PPP-complete and PWPP-complete respectively [41].

The complexity of total coding problems

To our knowledge, much less is known about the complexity of total problems that arise in coding theory. Instead, much work has focused on the *γ-approximate Minimum Distance Problem* (γ-MDP), in which the input is a linear code and the goal is to output a pair of distinct codewords $\boldsymbol{c}_1, \boldsymbol{c}_2$ such that the distance between them is within a factor γ of the minimum distance of the input code (or, since the code is linear, one can equivalently output a non-zero codeword with nearly minimal Hamming weight). This problem is known to be

NP-hard [42], even to approximate [14]. In contrast, we are interested in the problem of finding distinct codewords $\boldsymbol{c}_1, \boldsymbol{c}_2$ that are within distance d, where d depends only on the message length k and block size n of the code (and *not* on the minimum distance). We are particularly interested in the total regime, where such problems are very unlikely to be NP-hard.

To our knowledge, the only direct work in this area is a beautiful paper by Debris-Alazard, Ducas, and van Woerden [11], which showed an LLL-like algorithm for linear codes that efficiently finds codewords within the Griesmer bound [16]. (Note that this algorithm only works for linear codes, and indeed the Griesmer bound itself only applies to linear codes.)

Literature on multicollisions

There is quite a bit of prior work in the cryptography literature on *multicollision-resistant hash functions*. In our terminology, a multicollision-resistant hash function is some efficiently sampleable distribution of instances of (A, B)-Pigeon^L (i.e., circuits $\mathcal{C} : [A] \to [B]$) that are actually hard (i.e., that cannot be solved by polynomial-time algorithms with non-negligible probability). A survey of this literature is beyond the scope of this work. Broadly speaking, work in this area has been concerned with (1) understanding the relationship between collision resistance and multicollision resistance (i.e., the relationship between the case when $L = 2$ and the case when $L > 2$); (2) finding applications of multicollision-resistant hash functions; and (3) building multicollision-resistant hash functions from various cryptographic hardness assumptions.

Many of the techniques that we use to understand the relationship between (A, B)-Pigeon^L in different parameter regimes are directly inspired by this cryptographic literature. In particular, our inclusion based on Merkle trees and list-recoverable codes is a direct adaption of Bitanski, Kalai, and Paneth's construction from the cryptographic setting to our setting [6]; our black-box separations are inspired by [25]; and our non-black-box non-separation is a direct adaptation of Rothblum and Vasudevan's proof from the cryptographic setting to our setting [37]. (See also the very recent work of [9].)

In contrast, there is very little prior work in the worst-case setting. To our knowledge, the only works that consider the worst-case complexity of finding multicollisions are Komargodski, Naor, and Yogev [25] and Sotiraki [40] (though see Section 1.3). In both of these works, the worst-case complexity of (A, B)-Pigeon^L is not the primary focus, but both do define the special cases of the complexity class (A, B)-PMPP^L, when $A = B^2$ (specifically, $A = 2^{2n}$ and $B = 2^n$) and L is a constant. This is a natural setting of parameters, but we show interesting results in other parameter regimes as well.

Sotiraki in particular shows a complete problem for PMPP that is similar to the PPP-complete and PWPP-complete problems from [41]. In particular, Sotiraki's PMPP-complete problem bears some resemblance to certain lattice and coding problems, though the relationship is unclear. We refer the reader to [41, 40] for more.

Komargodski, Naor, and Yogev claimed a black-box separation between $(2^{2n}, 2^n)$-PMPP^L and $(2^{2n}, 2^n)$-PMPP$^{L'}$ for any constants $L' < L$, but their elegant proof contained a subtle bug that has not been fixed [26]. Our black-box separations use similar ideas but are weaker than what they originally claimed.

Literature on PPP and PWPP

In contrast, there is much literature studying the complexity classes PPP and PWPP, which correspond to the special case of PMPP when $L = 2$ (in different parameter regimes). PPP was introduced by Papadimitriou in his seminal work [34]. (PWPP appeared in many works

but seems to have been first named PWPP in [22].) Since then, many problems of interest have been shown to be contained in either PWPP or PPP [22, 2, 41, 8]. Until recently, only a small handful of problems were known to be complete for PPP or PWPP [34, 41, 2]. However, Bourneuf, Folwarczný, Hubáček, Rosen, and Schwartzbach recently showed a number of problems arising from extremal combinatorics that are complete for either PPP or PWPP [8].

There has also been some literature concerned with generalizing PPP and PWPP to classes other than PMPP. In particular, Pasarkar, Papadimitriou, and Yannakakis [35] recently introduced the class Polynomial Long Choice (PLC), which can be thought of as corresponding to the generalization of the pigeonhole principle obtained by iterating the pigeonhole principle many times.

1.5 A note on "codes" represented by circuits, injectivity, and systematic form

The coding theory results in this paper are concerned with the problem of finding close codewords (or many codewords that lie in a relatively small ball) in a "code" *represented by an arbitrary circuit* $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with size $\text{poly}(n, k, \log q)$. It is far more common in the literature to consider *linear* codes represented by a generator matrix (or, equivalently, an invertible circuit with linear gates). (Sometimes when arbitrary codes are considered in the literature, the code is simply represented by listing all q^k points, while we represent our codes succinctly.)

Whether one should really think of a generic circuit $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ as representing a "code" is not so clear. In particular, such a circuit might not be *injective*, i.e., there might be distinct "messages" $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{F}_q^k$ that map to the same "codeword" $\mathcal{C}(\boldsymbol{x}_1) = \mathcal{C}(\boldsymbol{x}_2)$. (And, there is likely no way to efficiently determine whether such a circuit is injective. Indeed, determining this is coNP-complete.) But, we find the coding-theoretic perspective to be quite useful. In particular, the notion of the distance of a code still makes sense with this slightly more general definition, and the bounds on the distance that we discuss in this paper still apply. Indeed, much of coding theory still makes sense if we treat such degenerate, non-injective codes simply as "codes with distance 0," and standard bounds in coding theory, such as the Singleton and Hamming bounds, still apply.

Of course, it is not an issue, and actually a strength that our upper bounds apply to such general "codes." Indeed, any upper bound that applies in the more general case when "codes" are represented by arbitrary circuits certainly applies to the special case of injective circuits or the even more special (and quite important) case of linear codes.

For our lower bounds, however, this can be viewed as a major flaw in our model. To partially mitigate this, we prove our hardness results in two different settings.

In the first "generic" setting, we show hardness for "codes" represented by arbitrary circuits $\mathcal{C} : [q]^k \to [q]^n$, with no presumed structure. In particular, the circuit is not necessarily injective. Indeed, one may argue that our reductions are rather artificial, in that the reductions often produce "codes" $\mathcal{C}$ such that $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$ is either zero or strictly larger than d, where d is the bound on the distance needed to solve the associated coding problem. So, these reductions rely quite heavily on this rather strange definition of a code in which multiple messages can map to the same codeword.

In the "systematic" setting, our codes $\mathcal{C} : [q]^k \to [q]^n$ are in *systematic* form, which means that the first k characters of $\mathcal{C}(\boldsymbol{x})$ are simply $\boldsymbol{x}$ itself. (Formally, we actually work with a smaller circuit $\mathcal{C}' : [q]^k \to [q]^{n-k}$, and we simply interpret $(\boldsymbol{x}, \mathcal{C}'(\boldsymbol{x}))$ as the codeword associated with $\boldsymbol{x}$.) Such circuits are clearly injective, and therefore this setting is much less artificial. (Codes in systematic form are also quite fundamental objects that arise naturally in coding theory.) In this setting, our formal results are a bit different, and we even show

that there are efficient algorithms for this setting in regimes where the problem is provably hard for codes represented by arbitrary circuits. However, the high-level picture is the same. In particular, we still show completeness in this setting. (See Figure 1.)

1.6 Open problems

We leave a number of interesting open problems. Here, we mention some of them.

Complexity of coding and lattice problems

We place the computational problems associated with a number of fundamental bounds on the minimum distance of a code in PMPP. However, we were unable to say anything non-trivial about the complexity of Delsarte's linear programming bound [12] and the closely related MRRW bound(s) [29], which are the best known bounds in an important range of parameters. The associated computational problems are, by definition, in TFNP, but we do not know any natural subclass of TFNP that contains these problems.

Similarly, the best known bound on the minimum distance in the ℓ_2 norm of a lattice relative to the determinant is the celebrated bound due to Kabatjanskiĭ and Levenšteĭn [24] (which is better than Blichfeldt's [7] by a factor of roughly $2^{1/10}$). The problem of finding a vector within the Kabatjanskiĭ and Levenšteĭn bound is again, by definition, in TFNP, but we do not know any natural subclass of TFNP that contains this problem.

In a different direction, we show hardness of various total coding problems (and even completeness in some regimes), but only for codes represented by circuits. It would be very exciting to show hardness results of total problems on *linear* codes (represented, e.g., by generator matrices), since these are the more standard problems. (The analogous question for lattices was already asked in [2] and also remains open.)

Better understanding of PMPP across different regimes

The first question that comes to mind about the complexity classes (A, B)-PMPP^L is, of course, "how do these classes relate to each other across different parameter regimes?" In the case when $L = 2$, it has long been known that the relationship between A and B does not matter much. For example,

$$((1 + 1/\operatorname{poly}(n)) \cdot 2^n, 2^n)\text{-PMPP}^2 = ((2^{n+\operatorname{poly}(n)}, 2^n)\text{-PMPP}^2 = \text{PWPP} .$$

For $L > 2$, it seems unlikely that a similar result holds. We instead describe a rich and rather complicated set of inclusions (and non-black-box relationships) among these classes, as well as black-box separations.

However, we have no evidence that these results are tight. One would ideally like to show black-box separations and inclusions that are tight with one another. (Or, alternatively, one might hope to prove non-trivial equalities (A, B)-PMPP$^L = (A', B')$-PMPP$^{L'}$ like what we know in the case of $L = L' = 2$.) The analogous average-case question has been the topic of much research in the cryptography literature [23, 33, 43, 5, 6, 25, 28, 13, 40, 37, 20], but to the authors' knowledge the worst-case setting was barely explored before this work and the recent exciting (concurrent and independent) work of Jain, Li, Robere, and Xun [21].

2 Upper bounds for coding problems

In this section, we prove that SDP and DenseBall are contained in PMPP in many parameter regimes of interest. In particular, for SDP, we show such results when the distance $d(n, k, q)$ corresponds to a number of celebrated bounds on the minimum distance of a code, including

the Singleton bound (Corollary 7), the Plotkin bound (Corollary 8), the Hamming bound (Corollary 13), and the Elias-Bassalygo bound (Corollary 15). For DenseBall, we show analogous results for the list Singleton bound (Corollary 11) and list Hamming bound (Corollary 13).

In fact, all of these results are corollaries of four technical results: Theorem 6, which reduces SDP to Pigeon via a simple truncation technique; Theorem 10, which does something similar for DenseBall; Theorem 12, which reduces DenseBall to Pigeon by "finding many overlapping Hamming balls centered on codewords"; and Theorem 14, which shows a generic reduction from SDP to DenseBall.

We also show in Section 2.1.1 an efficient algorithm for SDP within the Plotkin bound for codes in systematic form.

2.1 Upper bounds for the Singleton and Plotkin bounds

We start by giving a fairly general reduction from SDP (the problem of finding a pair of close codewords) to Pigeon^L. On input an instance $\mathcal{C}$ of SDP, the reduction works by defining a compressing circuit $\mathcal{C}'$ that truncates the output of $\mathcal{C}$. It then finds $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ such that $\mathcal{C}'(\boldsymbol{x}_1) = \cdots = \mathcal{C}'(\boldsymbol{x}_L)$ using its Pigeon^L oracle, and finally outputs a pair $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ that minimizes $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$. We then observe that special cases of this general result place (n, k, d)-SDP$_q$ in PWPP for d corresponding to the Singleton bound (Corollary 7) and PMPP for d corresponding to the Plotkin bound (Corollary 8).

▶ **Theorem 6.** *Let $k, m, n, q, L \in \mathbb{Z}^+$ be such that $m \leq k \leq n$ and $2 \leq L \leq q^{k-m}$ with $L \leq \mathrm{poly}(n, \log q)$. Then there is a Karp reduction from (n, k, d)-SDP$_q$ to (q^k, q^m)-Pigeon^L where $d := d_q^*(n - m, L)$.*

Proof. See full version [4]. ◀

We get useful corollaries from Theorem 6 that show how to compute vectors achieving the Singleton bound (the $L = 2$ case) and the Plotkin bound (for larger L) using a Pigeon^L oracle.

We now show that the problem of finding a pair of codewords whose distance is at most the Singleton bound is in PWPP.

▶ **Corollary 7** (Singleton bound in PWPP)**.** *Let $k, n, q \in \mathbb{Z}^+$ with $k \leq n$. Then there is a Karp reduction from $(n, k, n - k + 1)$-SDP$_q$ to (q^k, q^{k-1})-Pigeon^2. In particular, $(n, k, n - k + 1)$-SDP$_q$ is in PWPP.*

Proof. The claim follows from Theorem 6 by setting $m := k - 1$ and noting that, trivially, $d_q^*(n - m, L) = d_q^*(n - k + 1, L) \leq n - k + 1$ for any $L \geq 2$. ◀

We now show that the problem of computing a pair of codewords whose distance satisfies the high-rate Plotkin bound is in PMPP.

▶ **Corollary 8** (Plotkin bound in PMPP)**.** *Let $k, n, d, m, q, L \in \mathbb{Z}^+$ be such that $k \leq n$, $L \leq \mathrm{poly}(n)$, $L \leq q^{k-m}$, and*

$$d > \frac{L}{L-1}\left(1 - \frac{1}{q}\right)(n - m)\ , \tag{1}$$

then there is a Karp reduction from $(n, k, d-1)$-SDP$_q$ to (q^k, q^m)-Pigeon^L which runs in time $\mathrm{poly}(n, \log(q))$.

Proof. See full version [4]. ◀

2.1.1 Efficiently solving sysSDP up to the Plotkin bound

We now note that there is an efficient algorithm for finding codewords within the Plotkin bound *when the code is in systematic form.* In other words, we show that (n, k, d)-sysSDP$_q$ can be solved efficiently for d below the Plotkin bound, where we recall that sysSDP is the special case of SDP in which the code is in systematic form.

▶ **Theorem 9.** *Let $k, n, q \in \mathbb{Z}^+$ with $k < n$ and $q \geq 2$. There is a* poly(n, q)*-time algorithm for* (n, k, d)-sysSDP$_q$ *where*

$$d := (1 - 1/q)(n - k + \lceil \log_q(4qn) \rceil) \ .$$

Proof. See full version [4]. ◀

2.2 An upper bound for the list Singleton bound

Next, we give a reduction from DenseBall to Pigeon similar to the reduction from SDP to Pigeon in Theorem 6. We then show that this reduction places the *list* Singleton bound (i.e., a variant of the Singleton bound that corresponds to list decoding rather than unique decoding) in PMPP (up to a small error).

▶ **Theorem 10.** *Let $k, n, m, q, L \in \mathbb{Z}^+$ be such that $m < n$ and $2 \leq L \leq q^{k-m}$. Then there is a Karp reduction from* $(n, k, n - m - \lfloor (n - m)/L \rfloor)$-DenseBall$_q^L$ *to* (q^k, q^m)-Pigeon^L.

Proof. See full version [4]. ◀

Finally, we show that the problem of computing a set of codewords lying in a ball whose radius *almost* satisfies the list Singleton bound is in PMPP.

▶ **Corollary 11** (List Singleton bound in PMPP)**.** *Let $n, k, q, L \in \mathbb{Z}^+$ with $k > \lfloor \log_q(L) \rfloor$, let $m := k - \lfloor \log_q L \rfloor$, and let*

$$t := n - m + \lfloor (n - m)/L \rfloor = n - k + \lfloor \log_q L \rfloor - \lfloor (n - k + \lfloor \log_q L \rfloor)/L \rfloor \ . \tag{2}$$

Then (n, k, t)-DenseBall$_q^L$ *is in* (q^k, q^m)-PMPP^L.

Proof. See full version [4]. ◀

We remark that the radius t in Equation (2) is almost as small the smallest t satisfying the list Singleton bound. Indeed, one can check that any t satisfying the list Singleton bound must be such that

$$t > (1 - 1/L) \cdot (n - k + \log_q(L - 1)) \ ,$$

and that the radius t in Equation (2) satisfies

$$t \leq (1 - 1/L) \cdot (n - k + \lfloor \log_q(L) \rfloor) + 1 \ .$$

2.3 Upper bounds for the (list) Hamming and Elias-Bassalygo bounds

We now show containment of (n, k, d)-SDP$_q$ in PMPP, where d corresponds to the Hamming bound and Elias-Bassalygo bound. In particular, when d is equal to the value given by the Hamming bound, we show that SDP is contained in PPP (see Item 1 of Corollary 13) and for d slightly above the Hamming bound, SDP is contained in PWPP (see Item 2 of Corollary 13). In fact, we show a more general result that applies to the generalization (n, k, d)-DenseBall$_q^L$

of SDP and shows that for d within the *list* Hamming bound, this problem is in PMPP (see Item 3 of Corollary 13). We then show that (n, k, d)-SDP$_q$ is in PMPP whenever d above the Elias-Bassalygo, via a generic reduction from SDP to DenseBall (see Corollary 15). In fact, we obtain a smooth tradeoff between the distance d obtained by the reduction and the parameters A, B, and L in the reduction to (A, B)-Pigeon^L.

2.3.1 The (list) Hamming bound

We now give a reduction from DenseBall to Pigeon^L, the key to which is the circuit $\mathcal{C}_H^{n,V,q}$ giving an injection from $[V]$ into a Hamming ball for sufficiently large V defined in the full version [4]. (In fact, we will use such circuits $\mathcal{C}_H^{n,V,q}$ that are *bijective*.)

▶ **Theorem 12.** *Let $n, k, t, L \in \mathbb{Z}^+$, and let q be a prime power satisfying*

$$L \leq \left\lceil \frac{q^k \cdot V_q^n(t)}{q^n} \right\rceil .$$

Then there is a Karp reduction from (n, k, t)-DenseBall$_q^L$ to $(q^k \cdot V_q^n(t), q^n)$-Pigeon^L.

Proof. See full version [4]. ◀

We now use Theorem 12 to show that SDP and DenseBall with parameters corresponding to the (list) Hamming bound are in (one or more of) PPP, PWPP, and PMPP.

▶ **Corollary 13** ((List) Hamming bound in PPP, PWPP, and PMPP)**.** *Let $n, k, t, q, L \in \mathbb{Z}^+$ with $2 \leq q \leq \mathrm{poly}(n)$ and $L \leq \mathrm{poly}(n)$. Let*

$$\widetilde{L} := \frac{q^k \cdot V_q^n(t)}{q^n} .$$

Then:
1. *If $\widetilde{L} > 1$, then $(n, k, 2t)$-SDP$_q$ is in PPP.*
2. *If $\widetilde{L} \geq 1 + 1/\mathrm{poly}(n)$, then $(n, k, 2t)$-SDP$_q$ is in PWPP.*
3. *If $\widetilde{L} \geq L$, then (n, k, t)-DenseBall$_q^L$ is in $(q^k \cdot V_q^n(t), q^n)$-PMPP^L.*

Proof. See full version [4]. ◀

2.3.2 E pluribus duo – from many codewords to two

We now give a simple reduction from SDP to DenseBall^L for $L \geq 2$. The idea is that a ball of small radius t containing $L \geq 2$ codewords $\mathcal{C}(\boldsymbol{x}_1), \ldots, \mathcal{C}(\boldsymbol{x}_L)$ must contain a pair of codewords at small distance d. Clearly such a pair must exist at distance at most $2t$, but in fact when L is larger it is possible to get a better upper bound. So, the reduction simply computes the distance $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$ between each pair of codewords $\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j)$ for $i \neq j$, and outputs a pair $\boldsymbol{x}_i, \boldsymbol{x}_j$ that minimizes $\Delta(\mathcal{C}(\boldsymbol{x}_i), \mathcal{C}(\boldsymbol{x}_j))$.

▶ **Theorem 14.** *Let $n, k, t, q, L \in \mathbb{Z}^+$ be such that*

$$L \leq \left\lceil \frac{q^k \cdot V_q^n(t)}{q^n} \right\rceil .$$

Then there is a $\mathrm{poly}(n, L, \log q)$-time Karp reduction from (n, k, d)-SDP$_q$ to (n, k, t)-DenseBall$_q^L$, where $d := d_{[q]^n, \Delta}(t, L)$.

Proof. See full version [4]. ◀

2.3.3 The Elias-Bassalygo bound

We are now ready to show that the problem of computing a pair of codewords satisfying the Elias-Bassalygo bound is in PMPP.

▶ **Corollary 15** (Elias-Bassalygo bound in PMPP). *Let $n, k, d, t, q \in \mathbb{Z}^+$ with $t < J_q(d/n) \cdot n$, let $L := qnd + 1$, and suppose that*

$$L \leq \left\lceil \frac{q^k \cdot V_q^n(t)}{q^n} \right\rceil .$$

Then there is a poly(n, q)*-time Karp reduction from $(n, k, d-1)$-SDP$_q$ to $(q^k \cdot V_q^n(t), q^n)$-*Pigeon^L*. In particular, $(n, k, d-1)$-SDP$_q$ is in $(q^k \cdot V_q^n(t), q^n)$-*PMPP^L*.*

Proof. See full version [4]. ◀

3 Hardness of coding problems

We now turn to proving *hardness results* for SDP and DenseBall. In each of our hardness reductions we first assume that a gadget code, specified by a circuit $\mathcal{C}_A$, is given to the reduction as auxiliary input. We then instantiate the gadget code $\mathcal{C}_A$ with an explicit efficiently computable code $\mathcal{C}_A$ to obtain a true Karp reduction.

3.1 PWPP-hardness of SDP

We first show a generic reduction, with auxiliary from Pigeon^2 to SDP.

▶ **Theorem 16.** *Let $k, m, n, d, d' \in \mathbb{Z}^+$ be such that $m < k$ and $d < d'$. Let q be a prime power. Then there is a Karp reduction from (q^k, q^m)-*Pigeon^2 *to (n, k, d)-SDP$_q$ that takes a circuit $\mathcal{C}_A : \mathbb{F}_q^m \to \mathbb{F}_q^n$ defining an $(n, m, d')_q$ code as auxiliary input.*

Proof. See full version [4]. ◀

We will instantiate the reduction in Theorem 16 with codes $\mathcal{C}_A$ meeting the Zyablov bound, which can be computed in polynomial time (see [18, Theorem 10.2.1]). In fact, what we state is the special case of the (effective) Zyablov bound with sub-constant rate.[7]

▶ **Theorem 17** (Zyablov bound [44]). *For any (efficiently computable) $k = k(n) = o(n)$, constant prime power q, and constant $\varepsilon > 0$, there is a* poly(n)*-time algorithm that takes as input n and computes (a circuit representing) a code $\mathcal{C}_q^{\mathsf{Zyb}} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with relative distance $\delta \geq 1 - 1/q - \varepsilon$ for sufficiently large n.*

We now show that SDP is PWPP-hard on q-ary codes of relative distance $\delta := d/n$ less than $1 - 1/q$ and any constant rate. This hardness corresponds to the red shaded region in the top plot in Figure 1. We remark that $(n, k, \delta n)$-SDP$_q$ is easy for $\delta \geq 1 - 1/q$. In particular, for such any such δ, this problem can be solved by choosing any collection of polynomially many codewords and outputting the closest pair among them. So, Corollary 18 shows essentially tight hardness of SDP in terms of δ.

[7] We note that the premise of [18, Theorem 10.2.1] is stated with the requirement $\delta < 1/2$. While $\delta < 1/2$ is necessary for the $q = 2$ case, for general constant q, the result holds for $\delta < 1 - 1/q$, as in Theorem 17. It is also evident from the proof of the Zyablov bound that the result extends to superconstant q as well – i.e., that for any prime power q that is an unbounded function of n and any $k = o(n)$, there is an efficiently computable family of codes $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ with relative distance $1 - \varepsilon$.

▶ **Corollary 18** (PWPP-hardness of SDP). *Let q be a fixed prime power and let $\varepsilon, \varepsilon' > 0$ be constants. Then for all sufficiently large $n, k, d \in \mathbb{Z}^+$ with $k \le n \le \mathrm{poly}(k)$ and $d \le (1 - 1/q - \varepsilon')n$, there is a Karp reduction from $(q^k, q^{k^{1-\varepsilon}})$-Pigeon^2 to (n, k, d)-SDP$_q$.*

In particular, $(n, Rn, \delta n)$-SDP$_q$ is PWPP*-hard for any constant rate for any constant rate $R \in (0, 1]$ and constant relative distance $\delta \in (0, 1 - 1/q)$.*

Proof. See full version [4]. ◀

▶ Remark 19. We remark that the assumption $k \le n$ (equivalently, $R \le 1$) in Corollary 18 is not necessary, except to ensure that the instance $\mathcal{C}'$ of SDP output by the reduction meets the definition of a code used there. In fact, modifying Corollary 18 slightly shows PWPP-hardness of "(n, k, d)-SDP$_q$" for any $n \ge \Omega(k^\varepsilon)$ for constant $\varepsilon > 0$ (and any constant relative distance less than). This is because the hard instances $\mathcal{C}'$ of SDP constructed in Corollary 18 are such that either $\mathcal{C}'(\boldsymbol{x}) = \mathcal{C}'(\boldsymbol{y})$ or $\Delta(\mathcal{C}'(\boldsymbol{x}), \mathcal{C}'(\boldsymbol{y}))$ is large for $\boldsymbol{x} \neq \boldsymbol{y}$.

We also extend Corollary 18 to codes in systematic form (and in particular to codes with distance $d \ge 1$, which are represented by injective circuits; see Section 1.5). This hardness corresponds to the red shaded region in the bottom plot in Figure 1.

▶ **Corollary 20** (PWPP-hardness of sysSDP). *Let q be a fixed prime power and let $\varepsilon, \varepsilon' > 0$ be constants. Then for all sufficiently large $n, k, d \in \mathbb{Z}^+$ with $k \le n - k \le \mathrm{poly}(k)$ and $d \le (1 - 1/q - \varepsilon')(n - k)$, there is a Karp reduction from $(q^k, q^{(n-k)^{1-\varepsilon}})$-Pigeon^2 to (n, k, d)-sysSDP$_q$.*

In particular, $(n, Rn, \delta n)$-sysSDP$_q$ is PWPP*-hard for any constant rate $R \in (0, 1)$ and constant relative distance $\delta \in (0, (1 - 1/q) \cdot (1 - R))$.*

Proof. See full version [4]. ◀

3.2 PMPP-hardness of DenseBall

We next show a reduction from Pigeon^L for $L \ge 2$ to DenseBall analogous to Theorem 16.

▶ **Theorem 21.** *Let $k, m, n, t, L, L_A, L' \in \mathbb{Z}^+$ and let q be a prime power. Suppose that $L' \le \min\{\mathrm{poly}(n), q^{n-k}\}$ and $L \le \min\{\lceil L'/L_A \rceil, q^{k-m}\}$. Then there is a Karp reduction from (q^k, q^m)-Pigeon^L to (n, k, t)-DenseBall$_q^{L'}$ that takes a circuit $\mathcal{C}_A : \mathbb{F}_q^m \to \mathbb{F}_q^n$ defining a (t, L_A)-list decodable-code with minimum distance at least 1 (i.e., $\mathcal{C}_A$ is injective) as auxiliary input.*

Proof. See full version [4]. ◀

We now state a result on explicit codes that nearly achieve list-decoding capacity [17] (see also [18, Theorem 17.3.8]). We note in passing that these codes are folded Reed-Solomon codes, but we will only use them in a black-box way as our gadget codes $\mathcal{C}_A$ in Theorem 21.

▶ **Theorem 22** ([17], [18, Theorem 17.3.8]). *For any constant rate $R^* \in (0, 1)$, any sufficiently small constant $\varepsilon > 0$, and all sufficiently large $n \in \mathbb{Z}^+$, there exist linear q-ary $((1 - R^* - \varepsilon)n, L_A)$-list-decodable codes $\mathcal{C}_q^{\mathsf{FRS}}$ of dimension $R^* \cdot n$ and sufficiently large block length n, for some*

$$L_A \le \left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}, \quad q \le \left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon^2)}.$$

Furthermore, there is a $\mathrm{poly}(n)$-time algorithm for computing (a circuit representing) such codes $\mathcal{C}_q^{\mathsf{FRS}}$.

Finally, we use Theorem 22 to prove PMPP-hardness of DenseBall.

▶ **Corollary 23** (PMPP-hardness of DenseBall)**.** *For any constants $R, \rho \in (0,1)$ and positive integers $m := m(n) < o(n)$ and $L := L(n) \leq \text{poly}(n)$, there exists a prime power $q := q(n) \leq \text{poly}(n)$ and a list size $L' := L'(n) \leq \text{poly}(n)$ such that there is a Karp reduction from (q^k, q^m)-Pigeon^L to $(n, k, \rho n)$-DenseBall$_q^{L'}$, where $k := k(n) = \lfloor Rn \rfloor$.*

In particular, for these parameters, $(n, k, \rho n)$-DenseBall$_q^{L'}$ is (q^k, q^m)-PMPP^L-hard.

Proof. See full version [4]. ◀

As with SDP, we also show hardness of DenseBall for codes in systematic form. (See Section 1.5.)

▶ **Corollary 24** (PMPP-hardness of sysDenseBall)**.** *For any constants $R \in (0,1)$ and $0 < \rho < 1 - R$ and positive integers $m := m(n) < o(n)$ and $L := L(n) \leq \text{poly}(n)$, there exists a prime power $q = q(n) \leq \text{poly}(n)$ and a list size $L' = L'(n) \leq \text{poly}(n)$ such that there is a Karp reduction from (q^k, q^m)-Pigeon^L to $(n, k, \rho n)$-sysDenseBall$_q^{L'}$, where $k := k(n) = \lfloor Rn \rfloor$.*

In particular, for these parameters, $(n, k, \rho n)$-sysDenseBall$_q^{L'}$ is (q^k, q^m)-PMPP^L-hard

Proof. See full version [4]. ◀

4 Finding short lattice vectors is in PMPP

In this section, we show that the problem of finding suitably short non-zero lattice vectors (in ℓ_p norms) can be placed in (A, B)-PMPP^L, with a smooth tradeoff between the length of the vector obtained and L and B. In particular, when $L = 2$ and $A \approx B$, we find vectors whose length is at most the bound given by Minkowski's celebrated theorem [32] (up to a factor of $1 + 1/n^C$ for an arbitrarily large constant $C > 0$), and when $L = \text{poly}(n)$ and $A \approx LB$, we find shorter vectors, corresponding to a stronger bound due to Blichfeldt [7]. (Blichfeldt proved his bound in the ℓ_2 norm, but we generalize it. Again, we match Blichfeldt's bound up to a factor of $1 + 1/n^C$ for an arbitrarily large constant $C > 0$.)

To that end, we first prove the following technical proposition. We then derive the above results as corollaries.

▶ **Proposition 25.** *There is an algorithm that takes as input $p \geq 1$, a radius $r \geq 1$, an integer $q \geq 1$, integers $A > B \geq 1$, an integer $L \geq 1$, and a basis $\mathbf{B} \in \mathbb{Z}^{n \times n}$ for a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, makes a single query to an (A, B)-Pigeon^L oracle, and outputs a lattice vector $\boldsymbol{y} \in \mathcal{L}(\mathbf{B})$ with the following behavior. If $L \leq \lceil A/B \rceil$, $q \geq 100pn/r$ and*

$$q^n \det(\mathcal{L}) \leq B < A \leq \left(1 - \frac{20n}{rq}\right) \cdot q^n \cdot \text{vol}(\mathcal{B}_p^n(r)) \,,$$

then $0 < \|\boldsymbol{y}\|_p \leq d_{\ell_p^n}(r, L)$. Furthermore, the algorithm runs in time $\text{poly}(n, L, \log A, \log r, \log \|\mathbf{B}\|)$.

Proof. See full version [4] ◀

Recall that we are interested in γ-HSVP$_p$ for $\gamma \approx \text{vol}(\mathcal{B}_p^n(1))^{-1/n}$.

▶ **Corollary 26.** *Let $p \geq 1$ be a constant integer, and let $A = A(n), B = B(n), L = L(n) \in \mathbb{Z}^+$ be such that $2^{s(n)} = B < A \leq 2^{\text{poly}(n)}$ and $L \leq \lceil A/B \rceil$ for a sufficiently large polynomial $s(n)$. Then γ-HSVP$_p \in (A, B)$-Pigeon^L for*

$$\gamma := (1 + o(1)) \cdot \text{vol}(\mathcal{B}_p^n(1))^{-1/n} \cdot (A/B)^{1/n} \cdot d_{\ell_p^n}(1, L) \,,$$

where n is the rank of the lattice $\mathcal{L}$ in the γ-HSVP$_p$ instance.

In particular, $\gamma_M := 2 \cdot \mathrm{vol}(\mathcal{B}_p^n(1))^{-1/n}$ *corresponds to Minkowski's theorem , and* $((1 + o(1))\gamma_M)$-$\mathrm{HSVP}_p \in \mathsf{PWPP}$. *Furthermore, for the case of* $p = 2$, $\gamma_B := \sqrt{2} \cdot \mathrm{vol}(\mathcal{B}_2^n(1))^{-1/n}$ *corresponds to Blichfeldt's theorem , and* $((1 + o(1))\gamma_B)$-$\mathrm{HSVP}_2 \in (2LB, B)$-$\mathsf{PMPP}^L$ *for any* $L = \mathrm{poly}(\log B)$.

Proof. See full version [4]. ◀

5 Inclusions

See full version [4].

6 Black-box separations

See full version [4].

7 A non-black-box non-separation

See full version [4].

References

1 Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. A $2^{n/2}$-time algorithm for $\sqrt{n}$-SVP and $\sqrt{n}$-Hermite SVP, and an improved time-approximation tradeoff for (H)SVP. In *Eurocrypt*, 2021. URL: `http://arxiv.org/abs/2007.09556`.

2 Frank Ban, Kamal Jain, Christos H. Papadimitriou, Christos-Alexandros Psomas, and Aviad Rubinstein. Reductions in PPP. *Information Processing Letters*, 145:48–52, 2019. `doi:10.1016/j.ipl.2018.12.009`.

3 L. A. Bassalygo. New upper bounds for error-correcting codes. *Problems of Information Transmission*, pages 32–35, 1965.

4 Huck Bennett, Surendra Ghentiyala, and Noah Stephens-Davidowitz. The more the merrier! on total coding and lattice problems and the complexity of finding multicollisions. In *ITCS*, 2025. URL: `https://eccc.weizmann.ac.il/report/2024/018`.

5 Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In *Eurocrypt*, 2018.

6 Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: A paradigm for keyless hash functions. In *STOC*, 2018.

7 H. F. Blichfeldt. The minimum value of quadratic forms, and the closest packing of spheres. *Mathematische Annalen*, 101(1):605–608, 1929.

8 Romain Bourneuf, Lukáš Folwarczný, Pavel Hubáček, Alon Rosen, and Nikolaj I. Schwartzbach. PPP-completeness and extremal combinatorics. In *ITCS*, 2023.

9 Jan Buzek and Stefano Tessaro. Collision resistance from multi-collision resistance for all constant parameters. In *CRYPTO*, 2024.

10 Ivan Bjerre Damgård. A design principle for hash functions. In *CRYPTO*, 1989.

11 Thomas Debris-Alazard, Léo Ducas, and Wessel P. J. van Woerden. An algorithmic reduction theory for binary codes: LLL and more. *IEEE Transactions on Information Theory*, 68(5):3426–3444, 2022. `doi:10.1109/TIT.2022.3143620`.

12 Philippe Delsarte. *An algebraic approach to the association schemes of coding theory.* Thesis, Universite Catholique de Louvain, 1973.

13 Itai Dinur. Tight time-space lower bounds for finding multiple collision pairs and their applications. In *Eurocrypt*, 2020.

14 I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003. doi:10.1109/TIT.2002.806118.

15 Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC*, 2008.

16 J. H. Griesmer. A bound for error-correcting codes. *IBM Journal of Research and Development*, 4(5):532–542, 1960. doi:10.1147/RD.45.0532.

17 Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inf. Theory*, 54(1):135–150, 2008. doi:10.1109/TIT.2007.911222.

18 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. self-published, 2023. October 3rd, 2023 book version. URL: https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/web-coding-book.pdf.

19 R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950. doi:10.1002/j.1538-7305.1950.tb00463.x.

20 Yassine Hamoudi and Frédéric Magniez. Quantum time–space tradeoff for finding multiple collision pairs. *ACM Trans. Comput. Theory*, 15(1-2):3:1–3:22, 2023. doi:10.1145/3589986.

21 Siddhartha Jain, Jiawei Li, Robert Robere, and Zhiyang Xun. On pigeonhole principles and Ramsey in TFNP. In *FOCS*, 2024.

22 Emil Jeřábek. Integer factoring and modular square roots. *Journal of Computer and System Sciences*, 82(2):380–394, 2016. doi:10.1016/J.JCSS.2015.08.001.

23 Antoine Joux. Multicollisions in iterated hash functions. application to cascaded constructions. In *CRYPTO*, 2004.

24 Grigorii A. Kabatjanskiĭ and Vladimir I. Levenšteĭn. Bounds for packings on the sphere and in space. *Problemy Peredači Informacii*, 14(1):3–25, 1978.

25 Ilan Komargodski, Moni Naor, and Eylon Yogev. Collision resistant hashing for paranoids: Dealing with multiple collisions. In *Eurocrypt*, 2018.

26 Ilan Komargodski and Eylon Yogev. Personal communication, 2023.

27 Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.

28 Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In *Eurocrypt*, 2019.

29 R. McEliece, E. Rodemich, H. Rumsey, and L. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, 1977. doi:10.1109/TIT.1977.1055688.

30 Ralph C. Merkle. A certified digital signature. In *CRYPTO*, 1989.

31 Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In *Eurocrypt*, 2016. URL: http://eprint.iacr.org/2015/1123.

32 Hermann Minkowski. *Geometrie der Zahlen*. B.G. Teubner, 1910. URL: http://books.google.com/books?id=MusGAAAAYAAJ.

33 Mridul Nandi and Douglas R. Stinson. Multicollision attacks on some generalized sequential hash functions. *IEEE Transactions on Information Theory*, 53(2):759–767, 2007. doi:10.1109/TIT.2006.889721.

34 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

35 Amol Pasarkar, Christos Papadimitriou, and Mihalis Yannakakis. Extremal combinatorics, iterated pigeonhole arguments and generalizations of PPP. In *ITCS*, 2023.

36 M. Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. doi:10.1109/TIT.1960.1057584.

37 Ron D. Rothblum and Prashant Nalini Vasudevan. Collision-resistance from multi-collision-resistance. In *CRYPTO*, 2022.

38 Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53(23):201–224, 1987. doi:10.1016/0304-3975(87)90064-8.

39 R. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964. doi:10.1109/TIT.1964.1053661.

40 Aikaterini Sotiraki. *New Hardness Results for Total Search Problems and Non-Interactive Lattice-Based Protocols*. Thesis, Massachusetts Institute of Technology, 2020. URL: https://dspace.mit.edu/handle/1721.1/129310.

41 Katerina Sotiraki, Manolis Zampetakis, and Giorgos Zirdelis. PPP-completeness with connections to cryptography. In *FOCS*, 2018.

42 Alexander Vardy. Algorithmic complexity in coding theory and the Minimum Distance Problem. In *STOC*, 1997.

43 Hongbo Yu and Xiaoyun Wang. Multi-collision attack on the compression functions of MD4 and 3-pass HAVAL. In *ICISC*, 2007.

44 Victor Zyablov. An estimate of the complexity of constructing binary linear cascade codes. *Probl. Peredachi Inf.*, 1971.