Finding orthogonal vectors in discrete structures

Ryan Williams* Huacheng Yu†

Abstract

Hopcroft's problem in d dimensions asks: *given n points and n hyperplanes in $\mathbb{R}^d$, does any point lie on any hyperplane?* Equivalently, if we are given two sets of n vectors each in $\mathbb{R}^{d+1}$, is there a pair of vectors (one from each set) that are orthogonal? This problem has a long history and a multitude of applications. It is widely believed that for large d, the problem is subject to the *curse of dimensionality*: all known algorithms need at least $f(d) \cdot n^{2-1/O(d)}$ time for fast-growing functions f, and at the present time there is little hope that a $n^{2-\varepsilon} \cdot \mathrm{poly}(d)$ time algorithm will be found.

We consider Hopcroft's problem over finite fields and integers modulo composites, leading to both surprising algorithms and hardness reductions. The algorithms arise from studying the communication problem of determining whether two lists of vectors (one list held by Alice, one by Bob) contain an orthogonal pair of vectors over a discrete structure (one from each list). We show the randomized communication complexity of the problem is closely related to the sizes of matching vector families, which have been studied in the design of locally decodable codes. Letting $\mathbf{HOPCROFT}_{\mathcal{R}}$ denote Hopcroft's problem over a ring $\mathcal{R}$, we give randomized algorithms and almost matching lower bounds (modulo a breakthrough in SAT algorithms) for $\mathbf{HOPCROFT}_{\mathcal{R}}$, when $\mathcal{R}$ is the ring of integers modulo m or a finite field.

Building on the ideas developed here, we give a very simple and efficient output-sensitive algorithm for matrix multiplication that works over any field.

1 Introduction

A well-known problem popularized by John Hopcroft asks: given two sets $S_1, S_2 \subseteq \mathbb{R}^d$ of size n, are there $u \in S_1$ and $v \in S_2$ such that $\langle u, v\rangle = 0$? (Hopcroft originally posed it as: given n points in the plane and n lines, can we detect whether some point is on some line? This is equivalent to the above problem, with $d = 3$.) Geometrically speaking, the problem asks if there is an *incidence* among a set of n points in $\mathbb{R}^d$ and a set of n hyperplanes in $\mathbb{R}^d$ which pass through the origin. Computing the ℓ_2-nearest neighbor between S_1 and S_2 can often be reduced to Hopcroft's problem.[1] In data structures, Hopcroft's problem captures the difficulty of the infamous subset query and partial match query problems, restricted to 0-1 vectors: given two sets S_1, S_2 of subsets over a universe of d elements, is there a set $S \in S_1$ and $T \in S_2$ such that $S \subset T$?[2] The problem also has statistical applications; namely, in finding a pair of random variables with maximum correlation.[3]

The problem is not only foundational and pervasive, but it is also one where the trivial $O(n^2 \cdot d)$ time algorithm is not far from the fastest known, for large d. The $d = 3$ case has been known for 20 years to be solvable in $n^{4/3}2^{O(\log^* n)}$ time, and the general case is in $\tilde{O}(n^{2-2/d})$ time [Cha93, Mat93]. (Note this big-O hides exponential dependencies on d.) Erickson [Eri95] showed that several related problems are "Hopcroft-hard" in the sense that faster algorithms for them would improve the $d = 3$ case for Hopcroft; in other work [Eri96], he proved an $\Omega(n^{4/3})$ lower bound in a specialized model of computation.

When $|S_1| = |S_2| = n$ and $d = \Theta(\log n)$, there is no known algorithm running in $n^{2-\epsilon}$ time, even for 0-1 vectors in $O(\log n)$ dimensions. In fact, if such an algorithm

*Computer Science Department, Stanford University. Email: rrw@cs.stanford.edu. Supported in part by a David Morgenthaler II Faculty Fellowship and NSF CCF-1212372. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

†Computer Science Department, Stanford University. Email: yuhch123@stanford.edu. Supported by an Enlight Foundation Engineering Fellowship.

[1] After normalizing all vectors to unit length, computing the closest pair is equivalent to finding the pair with maximum inner product, which can be efficiently reduced to Hopcroft's problem when the alphabet size of the vectors is small (see footnote 3).

[2] Taking the complements of all sets in S_2, this is equivalent to asking if there are $S \in S_1$ and $T \in S_2$ such that $S \cap T = \varnothing$, which is equivalent to finding a pair of 0-1 vectors u_S and v_T such that $\langle u_S, v_T\rangle = 0$. Note that subset queries can be simulated with partial match queries as well.

[3] For instance, the problem of finding vectors $u \in S_1 \subseteq \{0,1\}^d$ and $v \in S_2 \subseteq \{0,1\}^d$ with minimum inner product can be reduced to Hopcroft's problem in $O(d^2)$ dimensions: for each possible inner product value $K = 0, 1, \ldots, d$, map each d-vector u and v to $O(d^2)$-vectors u' and v' such that $\langle u', v'\rangle = (\langle u, v\rangle - K)^2$. Finding the smallest K such that an orthogonal u' and v' exist amounts to finding the u, v with minimum inner product.

did exist, then k-CNF-SAT would be solvable in about $2^{(1-\epsilon/2)n}$ time, refuting the Strong Exponential Time Hypothesis (SETH) [IP01, IPZ01] (see Lemma A.1 in the appendix). Last year, G. Valiant [Val12] presented a new algorithm for the *random planted* case: suppose we have n uniform random vectors from $\{0,1\}^d$, along with a "planted" pair of vectors with statistically significant inner product. Valiant shows that the planted pair can be discovered in less than $n^{1.7} \cdot \text{poly}(d)$ time, giving several applications to learning theory. Unfortunately, Valiant's algorithm does not seem to be extendable to refuting SETH, even for the case of random k-SAT.

In this paper we study the variant of Hopcroft's problem where $\mathbb{R}$ is replaced with a discrete structure, such as a finite field, or a ring of integers modulo a composite. There are several motivations for studying these variants. First, the general study of incidences in finite geometry has increased in recent years in TCS, due to its strong relationships with coding theory and randomness extractors which are still far from being well understood [DGY11, Dvi12, BDL13]. Secondly, as indicated above, Hopcroft's problem on Boolean vectors is quite powerful: as the 0-1 d-dimensional case can be perfectly modeled in $\mathbb{Z}_{d+1}$, an efficient algorithm for this special case would already yield a breakthrough in SAT algorithms. Understanding the limits and possibilities over different structures should lead to a better understanding of the overall problem. (Our line of reasoning has also led to new thoughts about matrix multiplication, as shown below.)

1.1 Our Results

The aforementioned hardness results show that Hopcroft's problem retains much of its difficulty when studied over fields of high characteristic (greater than d). Our main results show that the problem is surprisingly easy in fields of low characteristic, and surprisingly difficult for some *constant-size* rings such as $\mathbb{Z}_6$.

The positive results are best presented in the context of a communication game between two players. Letting $\mathcal{R}$ be a ring, the **two-party Hopcroft problem over** $\mathcal{R}$ is defined as follows. Alice holds a set $U \subseteq \mathcal{R}^d$ and Bob holds a set $V \subseteq \mathcal{R}^d$ such that $|U| = |V| = n$. Their goal is to determine if there are $u \in U$ and $v \in V$ such that $\sum_i u_i \cdot v_i = 0$ over $\mathcal{R}$, with minimum communication between the two parties.

As we think of n as being much larger than $|\mathcal{R}|$ and d, we are interested in the case where communication complexity is a function of d and $|\mathcal{R}|$ only. Hence we define the **(randomized) communication complexity** of the two-party Hopcroft problem as follows: given $\mathcal{R}$ and d, it is the maximum (over all n) communication required by any (randomized public-coin) protocol computing two-party Hopcroft with $U, V \subseteq \mathcal{R}^d$ such that $|U| = |V| = n$. (A priori, note this quantity could be *infinite* if the communication complexity was an increasing function of n.) Therefore, all our communication bounds will in fact be of the form $O(f_{\mathcal{R}}(d))$ or $\Omega(f_{\mathcal{R}}(d))$ for some function $f_{\mathcal{R}}$, and of course this means we assume $f_{\mathcal{R}}(d) \leq n \cdot d$.

We use $\textbf{HOPCROFT}_{\mathcal{R}}$ to denote the typical computation version of the problem, where an algorithm is given both U and V at once, and the goal is to (efficiently) determine the existence of an orthogonal pair of vectors from U and V over $\mathcal{R}$.

1. Orthogonal Vectors and Matching Vectors. First we show that the communication complexity of the two-party Hopcroft problem is closely tied to the sizes of Matching Vector (MV) families, which have been recently studied for their relationship to locally decodable codes [DGY11, Efr12, BDL13].

DEFINITION 1.1. *Let $U = (u_1, u_2, \ldots, u_k)$, $V = (v_1, v_2, \ldots, v_k)$ be sequences of vectors over $\mathcal{R}^d$. (U, V) is called a* matching vector (MV) family of size k *if for all $i, j = 1, \ldots, k$, $\langle u_i, v_j \rangle = 0$ over $\mathcal{R}$ if and only if $i = j$. Define $\mathbf{MV}(\mathcal{R}, d)$ to be the maximum size of a matching vector family over $\mathcal{R}^d$.*[4]

THEOREM 1.1. *The two-party Hopcroft problem over $\mathcal{R}$ requires at least $\Omega(\mathbf{MV}(\mathcal{R}, d))$ communication in the randomized setting, while there is a deterministic protocol for two-party Hopcroft over $\mathcal{R}$ with (simultaneous) communication complexity $O(d \cdot \mathbf{MV}(\mathcal{R}, d))$.*

Note, by our *definition* of the communication complexity of two-party Hopcroft, we assume that $n \gg d$; in particular, $n \geq \mathbf{MV}(\mathcal{R}, d)$. Therefore, communication upper and lower bounds for our problem yield upper and lower bounds on MV families. We consider the problem of closing the above gap to be an interesting open problem. Moreover, executing the protocol in Theorem 1.1 requires solving an NP-hard problem in general, so it cannot be directly converted into an efficient algorithm – to do that, we use other ideas.

2. Orthogonal vectors over $\mathbb{Z}_{p^k}$. Over the ring $\mathbb{Z}_{p^k}$, we determine the randomized communication complexity of the two-party problem, and give an interesting algorithm for $\textbf{HOPCROFT}_{\mathbb{Z}_{p^k}}$.

THEOREM 1.2. *For every fixed prime power $m = p^k$, the randomized communication complexity of the two-party Hopcroft problem in $\mathbb{Z}_m$ is $\Theta\left(d^{m-1}\right)$.*

[4] In the literature, there is also a more general definition of S-matching vector family, which specifies an $S \subseteq \mathcal{R}$ such that for $u_i \in U$, $v_j \in V$, $\langle u_i, v_j \rangle \notin S$ if and only if $i = j$. Here we set $S = \mathcal{R} \backslash \{0\}$ as the default.

The upper bound uses a connection between orthogonal vectors and polynomial evaluation problems, some modular polynomial tricks [BBR94], and the ideas behind Freivalds' matrix multiplication checker [Fre77]. The lower bound follows from Theorem 1.1 applied to existing lower bounds on MV families. Directly interpreting Theorem 1.2 algorithmically, we obtain:

THEOREM 1.3. *For every prime power $m = p^k$, there is a randomized algorithm for* $\mathbf{HOPCROFT}_{\mathbb{Z}_m}$ *running in $O(n \cdot d^{m-1})$ time.*[5]

For instances where $n \geq d^{m-2}$ and $m \leq d$, Theorem 1.3 would be preferred over the trivial $O(n^2 \cdot d)$ or $O(n \cdot m^d)$ time algorithms.

It is natural to wonder if the large exponent of m can be reduced. We can show that a considerable reduction in the exponent would yield a breakthrough in SAT algorithms, contradicting the following conjecture, known as the Strong Exponential Time Hypothesis (SETH):

CONJECTURE 1.1. (SETH [IP01, IPZ01]) *For every constant $\delta < 1$ there is a clause width k such that k-SAT cannot be solved in $2^{\delta n}$ time.*

Although SETH looks rather strong, all known SAT algorithms are consistent with it, and a string of recent papers [CIP09, DW10, PW10, LMS11, CNP+11, CDL+12, PP12, HKN12, Cyg12, CKN13, RW13, FHV13] have shown that the SETH stands in the way of improving many algorithms for many diverse problems: if the best known algorithm can be improved, then SETH is false. We show that a reduction of the exponent in Theorem 1.3 from $m-1$ to $o(m/\log m)$ would refute SETH:

THEOREM 1.4. *Suppose there is an $\varepsilon > 0$ and a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(m)/(m/\log m) \to 0$ and for infinitely many m,* $\mathbf{HOPCROFT}_{\mathbb{Z}_m}$ *can be solved in time $n^{2-\varepsilon} \cdot d^{f(m)}$. Then SETH is false.*

3. Orthogonal vectors over $\mathbb{F}_{p^k}$. Over finite fields, we find that the communication complexity of two-party Hopcroft can be low:

THEOREM 1.5. *For every fixed prime p and constant k, the randomized communication complexity of the two-party Hopcroft problem over $\mathbb{F}_{p^k}$ is $\Theta(d^{(p-1)k})$.*

For fields of small characteristic, the exponent is only logarithmic in the size of the field. The protocol achieving the upper bound exploits the fact that multiplication by an element in $\mathbb{F}_{p^k}$ is a linear transformation on vectors in $(\mathbb{F}_p)^k$. We use this fact to efficiently reduce the problem over $\mathbb{F}_{p^k}$ to the problem over $\mathbb{F}_p$, which can then be solved using the techniques of Theorem 1.3. We also obtain an algorithm with a matching exponent:

THEOREM 1.6. *For every prime p and $k \in \mathbb{N}$, there is a randomized algorithm for* $\mathbf{HOPCROFT}_{\mathbb{F}_{p^k}}$ *running in $O(n \cdot d^{(p-1)k})$ time.*

In the case of the field $\mathbb{F}_{p^k}$, the complexity of exhaustive search takes either $O(n^2 \cdot d)$ or $O(n \cdot p^{kd})$ time, so Theorem 1.6 is a significant speedup when $p < d$ and $d^{(p-1)k-1} < n$; that is, when the field characteristic is smaller than the dimension, and the dimension is much smaller than the number of vectors.

Similar to Theorem 1.4, improving the exponent much further would refute Strong ETH:

THEOREM 1.7. *Suppose there is an $\varepsilon > 0$ and a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(x)/(x/\log x) \to 0$ and for infinitely many $\mathbb{F} = \mathbb{F}_{p^k}$,* $\mathbf{HOPCROFT}_{\mathbb{F}}$ *is solvable in time $n^{2-\varepsilon} \cdot d^{f((p-1)k)}$. Then SETH is false.*

4. Orthogonal vectors over $\mathbb{Z}_6$. Given the above results, it is natural to wonder if they extend to $\mathbb{Z}_m$ where m is not a prime power (the smallest case is $m = 6$). Here, the picture is startlingly different:

THEOREM 1.8. *Every randomized protocol for the two-party Hopcroft problem over $\mathbb{Z}_6$ requires $d^{\Omega\left(\frac{\log d}{\log\log d}\right)}$ bits of communication.*

That is, unlike the previous two cases, we cannot hope to find a communication protocol with complexity polynomial in d. The proof is immediate from Theorem 1.1 and existing MV family constructions.

Using the polynomials of Beigel, Barrington and Rudich [BBR94], we prove that efficient algorithms for Hopcroft's problem over $\mathbb{Z}_6$ could be used to derive algorithms for the case of $\mathbb{Z}$ which break the usual "curse of dimensionality" hypotheses:

THEOREM 1.9. *If* $\mathbf{HOPCROFT}_{\mathbb{Z}_6}$ *is solvable in $n^{2-\varepsilon} \cdot d^{o_d\left(\frac{\log d}{(\log\log d)^2}\right)}$ time for some $\varepsilon < 1$, then* $\mathbf{HOPCROFT}_{\mathbb{Z}}$ *over vectors in $\{0,1\}^d$ can be solved in time $n^{2-\delta} 2^{o(d)}$ for some $\delta < 1$.*

An algorithm for $\mathbf{HOPCROFT}_{\mathbb{Z}}$ with such a running time would have several new applications to pattern matching and subset problems, as well as a refutation of SETH:

COROLLARY 1.1. *If* $\mathbf{HOPCROFT}_{\mathbb{Z}_6}$ *is solvable in $n^{2-\varepsilon} \cdot d^{o_d\left(\frac{\log d}{(\log\log d)^2}\right)}$ time for some $\varepsilon < 1$, then SETH is false.*

[5] In all statements we make about randomized algorithms, the qualifier "with probability at least 99%" is implied.

5. Output-sensitive and communication-efficient matrix multiplication. Building on the ideas developed to solve the above problems, we give a very simple output-sensitive communication protocol for matrix multiplication over *any* field $\mathbb{F}$. Suppose Alice holds a matrix $A \in \mathbb{F}^{n\times d}$, Bob holds a matrix $B \in \mathbb{F}^{d\times n}$, and they wish to compute $A \cdot B$.

THEOREM 1.10. *Let $\mathbb{F}$ be an arbitrary finite field, let $A \in \mathbb{F}^{n\times d}$, and let $B \in \mathbb{F}^{d\times n}$. After $O(nd\log(|\mathbb{F}|\, n))$ preprocessing time (and no communication), Alice and Bob can compute every nonzero entry of $A\cdot B$ whp, with only $O(d(\log n)(\log(|\mathbb{F}|\, n)))$ communication and time delay per nonzero.*

That is, for "skinny" matrices, the communication and time cost of computing the nonzero entries of the problem can be surprisingly low. As a corollary, we obtain a new $\tilde{O}((m+n)\cdot d)$ time combinatorial algorithm for $n \times d \times n$ matrix multiplication with m nonzeroes in the output; after $\tilde{O}(nd)$ preprocessing, the algorithm outputs nonzero entries with $\tilde{O}(d)$ delay per nonzero.

Algorithms with similar running times were known for Boolean matrix multiplication over the (OR, AND) semiring [Lin11] and over the reals [Ind05, Pag12] using compressed sensing techniques, but not for other fields. (Our algorithm can also be extended to $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{C}$; the running time obviously becomes dependent on the bit-complexities of the entries.) Pagh's matrix multiplication [Pag12] is closest in spirit to ours: his algorithm achieves similar time and communication bounds, but is algorithmically more complicated, using multiple hash functions and FFTs. Low-space streaming models for matrix multiplication (which also yield communication-efficient protocols) have been studied in the past, but with a focus on approximate solutions; for example, [DKM06, Sar06, CW09].

Another consequence is that the above algorithms for Hopcroft's problem can be extended to algorithms that can efficiently *list* all orthogonal pairs of vectors, for example:

COROLLARY 1.2. *There is an algorithm for listing all incidences over a given set of n points and n hyperplanes in $(\mathbb{F}_{p^k})^d$ that runs in $\tilde{O}((n+m)\cdot d^{k(p-1)})$ time, where m is the total number of incidences.*

Problems of incidence detecting and counting have long been studied in computational and discrete geometry (cf. the survey of Dvir [Dvi12] and the book of Matousek [Mat02]).

2 Orthogonal Vectors and Matching Vectors

In what follows, let $\mathcal{R}$ be a commutative ring.

REMINDER OF THEOREM 1.1 *The two-party Hopcroft problem over $\mathcal{R}$ requires at least $\Omega(\mathbf{MV}(\mathcal{R}, d))$ communication in the randomized setting, while there is a deterministic protocol for two-party Hopcroft over $\mathcal{R}$ with (simultaneous) communication complexity $O(d \cdot \mathbf{MV}(\mathcal{R}, d))$.*

Proof. Recall in the communication problem DISJOINTNESS, two parties each hold sets $S_1, S_2 \subseteq [\ell]$, and they want to determine if $S_1 \cap S_2 = \varnothing$. To prove the communication lower bound, we give a reduction from DISJOINTNESS to the two-party Hopcroft problem. Fix $\mathcal{R}$ and d, and suppose there is a MV family over $\mathcal{R}$ with vectors of dimension d, where $U = (u_1, u_2, \ldots, u_t)$, $V = (v_1, v_2, \ldots, v_t)$.

Let Alice and Bob have sets $S_1, S_2 \subseteq [t]$, respectively. Alice and Bob can determine disjointness of S_1 and S_2 by individually computing the sets of vectors $A = \{u_i \mid i \in S_1\}$ and $B = \{v_j \mid j \in S_2\}$, then solving the two-party Hopcroft problem in $\mathcal{R}^d$. By the definition of MV family, there are vectors $u \in A$ and $v \in B$ such that $\langle u, v\rangle = 0$ over $\mathcal{R}$ if and only if there is some $i \in S_1 \cap S_2$. However, it is known that every randomized protocol for DISJOINTNESS over $[\ell]$ requires $\Omega(\ell)$ communication [KS92, Raz92]. Therefore the two-party Hopcroft problem over $\mathcal{R}^d$ requires $\Omega(\mathbf{MV}(\mathcal{R}, d))$ communication.

Next, we describe a deterministic protocol for the two-party Hopcroft problem over commutative rings. Alice computes the following procedure with her set of vectors $A = \{a_1, \ldots, a_n\}$:

> Set $A' = \varnothing$. Set $A'' = A$.
> For all $i = 1, \ldots, n$,
> Remove a_i from A''.
> If there exists a $v_i \in \mathcal{R}^d$ such that $\langle a_i, v_i\rangle = 0$ and for all $a_j \in A' \cup A''$ we have $\langle a_j, v_i\rangle \neq 0$,
> then add a_i to A'.
> Send A' to Bob.

Bob then checks by himself whether there are $u \in A'$ and $v \in B$ such that $\langle u, v\rangle = 0$ over $\mathcal{R}$, returning the result.

Now we prove the protocol is correct. First, if A, B do not contain an orthogonal pair of vectors, then A', B does not either, as $A' \subseteq A$. The interesting case is when A, B do contain an orthogonal pair. A vector a_i is not added to A' in the ith stage of the for-loop, only when for every $v \in \mathcal{R}^d$ orthogonal to a_i, there is another $a_j \in A' \cup A''$ (either a_j was added earlier, or $j > i$) that is also orthogonal to v. Therefore at the end of stage i in the for-loop, $(A' \cup A'', B)$ has an orthogonal pair if and only if $((A' \cup A'') \cup \{a_i\}, B)$ has an orthogonal pair. By induction, (A, B) has an orthogonal pair if and only if $(A' \cup A'', B)$ has such a pair, at the end of every stage

in the for-loop. It follows that the protocol correctly computes the function.

The communication cost of the protocol is exactly $d \cdot |A'|$. We claim that there is a set of vectors B' such that (A', B') is a MV family over $\mathcal{R}$. By construction, for every $a_i \in A'$, there is a vector v_i such that $\langle a_i, v_i \rangle = 0$ but v_i is not orthogonal to any other vectors in $A' \cup A''$ at that point. It follows that v_i cannot be orthogonal to any vector in the final set A'. Letting $B' = (v_1, v_2, \ldots)$, we have that (A', B') is a MV family over $\mathcal{R}$. Therefore $|A'| \leq \mathbf{MV}(\mathcal{R}, d)$, and the communication cost is at most $O(d \cdot \mathbf{MV}(\mathcal{R}, d))$. □

We remark that, although Alice's deterministic protocol above can be executed in polynomial time when $\mathcal{R} = \mathbb{F}_2$, it is probably not possible to execute the protocol efficiently in general, even for $\mathcal{R} = \mathbb{F}_3$:

PROPOSITION 2.1. *Given $\{v_0, v_1, \ldots, v_n\} \subseteq \mathbb{F}_3^d$, it is* NP*-hard to find an $x \in \mathbb{F}_3^d$ that satisfies $\langle v_0, x \rangle = 0$, $\langle v_1, x \rangle \neq 0, \ldots, \langle v_n, x \rangle \neq 0$ over $\mathbb{F}_3$.*

Proof. (Sketch) We reduce from the 3-coloring problem. Consider a graph G with d nodes and n edges for which we wish to check 3-colorability. For each edge $e_i = \{u_j, u_k\}$, we set $v_i \in \mathbb{F}_3^d$ to be the vector with $v_{ij} = 1$, $v_{ik} = -1$, and 0 in all other entries. Finally, set v_0 to be the all-zero vector. Every $x \in \mathbb{F}_3^d$ corresponds to a 3-coloring of the d nodes in G. The vector v_i corresponds to the constraint that the endpoints of e_i cannot have the same color. It is not hard to show that there is a vector x satisfying $\langle v_0, x \rangle = 0$, $\langle v_1, x \rangle \neq 0, \ldots, \langle v_n, x \rangle \neq 0$ over $\mathbb{F}_3$ if and only if G is 3-colorable. □

3 Hopcroft's problem over $\mathbb{Z}_{p^k}$

As a simple warm-up, we start by showing that finding *non-orthogonal* vectors is easy. Many of our results consist in finding novel ways to reduce Hopcroft's problem in one structure to finding non-orthogonal pairs in another structure.

PROPOSITION 3.1. (FOLKLORE) *For any m, given two sets of vectors $U, V \subseteq \mathbb{Z}_m^d$, there is a randomized algorithm which runs in time $\tilde{O}((|U| + |V|)d)$ and determines whether there is a pair $u \in U$, $v \in V$ such that $\langle u, v \rangle \neq 0$ over $\mathbb{Z}_m$, with high probability.*

Proof. Let M_U be a $|U| \times d$ matrix such that the i-th row is the i-th vector $u_i \in U$, M_V be a $d \times |V|$ matrix such that the j-th column is the j-th vector $v_j \in V$. Note that $(M_U M_V)_{i,j} = \langle u_i, v_j \rangle$, so the goal is to test whether $M_U M_V$ is the all-zero matrix. By the analysis of Freivalds' algorithm [Fre77], for random vectors $u_0 \in (\mathbb{Z}_m^\star)^{|U|}$, if $M_U M_V$ is not the all-zero matrix, then $u_0^T M_U M_V$ will not be the all-zero vector with probability at least $1 - 1/|\mathbb{Z}_m^\star|$. (Of course, if $M_U M_V$ is the all-zero matrix, then $u_0^T M_U M_V$ is always 0.) Thus this problem can be solved in linear time, since computing $u_0^T M_U M_V$ requires only linear time. Repeated trials increase the probability of success. □

The above immediately leads to a fast communication protocol:

COROLLARY 3.1. *For any m, if Alice and Bob hold sets $U, V \subseteq \mathbb{Z}_m^d$ respectively, there is a randomized protocol which costs $O(d \log m)$ communication and determines whether there are $u \in U$, $v \in V$ such that $\langle u, v \rangle \neq 0$ over $\mathbb{Z}_m$, with high probability.*

Proof. Alice and Bob construct M_U and M_V as in Proposition 3.1, Alice generates a random u_0, computes $u_0^T M_U$, and sends it to Bob who can compute the result by himself. □

The following very useful lemma reduces the problem of finding solutions to polynomials to finding orthogonal (or non-orthogonal) vector pairs.

LEMMA 3.1. *Let $\mathcal{R}$ be a commutative ring and Q be a polynomial in $d_1 + d_2$ variables with m monomials. Given $S_1 \subseteq \mathcal{R}^{d_1}$ and $S_2 \subseteq \mathcal{R}^{d_2}$, there is a reduction from the problem of finding $u \in S_1$, $v \in S_2$ such that $Q(u_1, \ldots, u_{d_1}, v_1, \ldots, v_{d_2})$ is nonzero (respectively, is zero) to the problem of finding a non-orthogonal (resp., orthogonal) pair of vectors in $\mathcal{R}^m$ with sets S_1', S_2' of size equal to that of S_1 and S_2. The reduction takes $\tilde{O}(n \cdot m \cdot (d_1 + d_2))$ additions and multiplications in $\mathcal{R}$, where $n = |S_1| + |S_2|$.*

Proof. Since the polynomial Q has m monomials, without loss of generality, we may assume that

$$Q(x_1, \ldots, x_{d_1}, y_1, \ldots, y_{d_2}) = \sum_{i=1}^{m} c_i \prod_{j=1}^{d_1} x_j^{\alpha_{ij}} \prod_{j=1}^{d_2} y_j^{\beta_{ij}}.$$

For every $u \in \mathcal{R}^{d_1}$, define $f : \mathcal{R}^{d_1} \to \mathcal{R}^m$ to be $f(u)_i = c_i \prod_{j=1}^{d_1} u_j^{\alpha_{ij}}$. Similarly for $v \in \mathcal{R}^{d_2}$, define $g : \mathcal{R}^{d_2} \to \mathcal{R}^m$ to be $g(v)_i = \prod_{j=1}^{d_2} v_j^{\beta_{ij}}$. Observe that $Q(u, v) = \langle f(u), g(v) \rangle$.

For all $u \in S_1, v \in S_2$, our reduction puts $f(u)$ in S_1' and $g(v)$ in S_2', in $\tilde{O}(n \cdot m \cdot (d_1 + d_2))$ additions and multiplications. We have converted the problem of finding u, v such that $Q(u, v) \neq 0$ (resp. $= 0$) to finding a non-orthogonal (resp. orthogonal) pair of vectors from S_1' and S_2'. □

Next, we show how to map arbitrary vectors in $\mathbb{Z}_m^d$ into binary vectors, while preserving the inner product modulo m.

LEMMA 3.2. *For all positive integers m, d, there are functions $f, g : \mathbb{Z}_m^d \to \{0,1\}^{m^2 d}$, such that for all $x, y \in \mathbb{Z}_m^d$, $\langle x, y\rangle \equiv \langle f(x), g(y)\rangle \bmod m$. Moreover, they can be both computed in $O(m^2 d)$ time.*

Proof. We first show that the theorem holds for $d = 1$. For $x, y \in \mathbb{Z}$, we construct vectors $f(x)$ and $g(y)$ of length m^2 which have m "blocks" and each block has m components. For $f(x)$, the first x blocks are filled with all ones, and put zeroes in the remaining blocks. For $g(y)$, each block has the same form: we put ones in the first y entries of the block, and zeroes in the other entries. It is not hard to verify $\langle f(x), g(y)\rangle = x \cdot y$, and both f and g can be computed efficiently.

For general d, we just map each entry of the vector to a 0-1 vector of length m^2, and concatenate these d vectors. □

REMINDER OF THEOREM 1.2 *For every fixed prime power $m = p^k$, the randomized communication complexity of the two-party Hopcroft problem in $\mathbb{Z}_m$ is $\Theta\left(d^{m-1}\right)$.*

Proof. For the upper bound, by Lemma 3.2, we can assume the inputs are 0-1 vectors without loss of generality. (In general, this increases the complexity from d^{m-1} to $(dm^2)^{m-1} \le O(d^{m-1})$.) Consider the following polynomial over d Boolean variables:

$$P(x_1, \ldots, x_d) = \sum_{i=0}^{p^k - 1} (-1)^i \left(\sum_{S \subseteq [d], |S| = i} \prod_{j \in S} x_j \right).$$

Beigel, Barrington, and Rudich [BBR94] proved that P has the following property:

$$P(x_1, \ldots, x_d) \equiv \begin{cases} 1 \bmod p, & \text{if } \sum_{i=1}^d x_i \equiv 0 \bmod p^k \\ 0 \bmod p, & \text{if } \sum_{i=1}^d x_i \not\equiv 0 \bmod p^k \end{cases}$$

Define the polynomial $Q(u_1, u_2, \ldots, u_d, v_1, \ldots, v_d) = P(u_1 \cdot v_1, u_2 \cdot v_2, \ldots, u_d \cdot v_d)$. It is easy to verify that, for $u, v \in \{0,1\}^d$, $Q(u, v) \not\equiv 0$ over $\mathbb{Z}_p$ if and only if $\langle u, v\rangle = 0$ over $\mathbb{Z}_{p^k}$. Note that the number of monomials in Q is the same as P, which is $\binom{d}{\le m-1} = O(d^{m-1})$.[6] By Lemma 3.1, we can reduce the problem of finding a pair u, v such that $Q(u, v) \neq 0$ to finding a non-orthogonal pair of vectors over $\mathbb{Z}_m$ in $O(d^{m-1})$ dimensions. The reduction itself costs no communication at all. Applying Corollary 3.1, we get an protocol using $O(d^{m-1})$ communication.

Now we turn to the lower bound. Dvir, Gopalan, and Yekhanin [DGY11] proved that

$$\mathbf{MV}(\mathbb{Z}_m, d) \ge \binom{d}{m-1},$$

by letting $u_{ij} = 1$ if $j \in S_i$, $u_{ij} = 0$ otherwise, and $v_{ij} = 1 - u_{ij}$, where $\{S_i\}$ are $(m-1)$-sized subsets of $[d]$. Hence the communication lower bound follows from by applying Theorem 1.1. □

REMINDER OF THEOREM 1.3 *For every prime power $m = p^k$, there is a randomized algorithm for $\mathbf{HOPCROFT}_{\mathbb{Z}_m}$ running in $O(n \cdot d^{m-1})$ time.*

Proof. As in the proof of Theorem 1.2, by Lemma 3.2, we can first assume the inputs are 0-1 vectors. Then we use the same polynomials P and Q, and apply Lemma 3.1 to reduce the problem to finding a non-orthogonal pair in $O(d^{m-1})$ dimensions. The reduction is also time-efficient, running in time $O(n \cdot d^{m-1})$. Applying Proposition 3.1, the theorem holds. □

Let $\mathcal{R}$ be a commutative ring, and P be a polynomial over k 0-1 variables $x_1, x_2, \ldots, x_k$, such that

$$P(x_1, \ldots, x_k) = 0 \text{ iff } x_1 = \cdots = x_k = 0 \text{ over } \mathcal{R}.$$

Let $\deg_{\mathcal{R}}(\mathrm{OR}_k)$ be the minimum possible degree of such a polynomial that can be efficiently constructed (e.g., in time that is linear in the number of monomials). We have the following theorem.

THEOREM 3.1. *Suppose there is an infinite sequence $\{\mathcal{R}_m\}$ of rings and $\epsilon > 0$ such that for all sufficiently large m and all constant c, there is an algorithm solving $\mathbf{HOPCROFT}_{\mathcal{R}_m}$ on n vectors in d dimensions in time $n^{2-\epsilon} 2^{d_0/c}$, for d_0 satisfying $d \ge \binom{d_0}{\le \deg_{\mathcal{R}_m}(\mathrm{OR}_{d_0})}$. Then SETH is false.*

Proof. We give a reduction from $\mathbf{HOPCROFT}_{\mathbb{Z}}$ on 0-1 inputs and apply Lemma A.1 (in the appendix). Consider an instance of $\mathbf{HOPCROFT}_{\mathbb{Z}}$ on 0-1 vectors in d_0 dimensions. We have sets of vectors U, V and want to determine if there are $u \in U, v \in V$ such that $\langle u, v\rangle = 0$. For every m, by the definition of $\deg_{\mathcal{R}_m}$, we can efficiently construct a polynomial P of degree $\deg_{\mathcal{R}_m}$ on d_0 0-1 variables computing the OR function over $\mathcal{R}_m$. Note the number of monomials in P is at most $\binom{d_0}{\le \deg_{\mathcal{R}_m}(\mathrm{OR}_{d_0})} \le d$. Let $Q(u_1, \ldots, u_{d_0}, v_1, \ldots, v_{d_0}) = P(u_1 v_1, \ldots, u_{d_0} v_{d_0})$. By Lemma 3.1, we can reduce the problem to an instance of $\mathbf{HOPCROFT}_{\mathcal{R}_m}$ in d dimensions.

To apply Lemma A.1, let c be an arbitrary constant and let $d_0 = c \log n$. We first find m large enough so that $\mathbf{HOPCROFT}_{\mathcal{R}_m}$ can be solved within time $O(n^{2-\epsilon} \cdot 2^{\frac{\epsilon}{2c} \cdot d_0})$, then perform the above reduction and apply the fast algorithm for Hopcroft's problem on $\mathcal{R}_m$. We obtain an algorithm for $\mathbf{HOPCROFT}_{\mathbb{Z}}$ on vectors in $\{0,1\}^{c \log n}$ with running time $O(n^{2-\epsilon} \cdot 2^{\frac{\epsilon}{2c} \cdot d_0}) = O(n^{2-\epsilon/2})$. As this holds for every c, Lemma A.1 applies and SETH is false. □

[6] We use the notation $\binom{n}{\le k} = \sum_{i=1}^k \binom{n}{i}$.

Reminder of Theorem 1.4 *Suppose there is an $\varepsilon > 0$ and a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(m)/(m/\log m) \to 0$ and for infinitely many m, $\mathbf{HOPCROFT}_{\mathbb{Z}_m}$ can be solved in time $n^{2-\varepsilon} \cdot d^{f(m)}$. Then SETH is false.*

Proof. This is a corollary of Theorem 3.1. Let $\mathcal{R}_m = \mathbb{Z}_m$; note that $\deg_{\mathbb{Z}_m}(\mathrm{OR}_{d_0}) \le \lceil d_0/(m-1) \rceil$.[7] Applying a similar argument as in Theorem 3.1, we only need $d \le \binom{d_0}{\lceil d_0/(m-1) \rceil} = O(2^{H(1/(m-1))d_0})$, where $H(\cdot)$ is the binary entropy function. Observe that

$$d^{o(\frac{m}{\log m})} \le 2^{o(H(1/(m-1))d_0 \cdot \frac{m}{\log m})} \le 2^{o(d_0)}$$

where we applied the inequality $H(1/(m-1))m \le 2 \log m$ for integer $m \ge 2$. If we have an algorithm for $\mathbf{HOPCROFT}_{\mathbb{Z}_m}$ that runs in time $n^{2-\epsilon} \cdot d^{o(\frac{m}{\log m})}$, it satisfies the condition of Theorem 3.1, and SETH is false. □

4 Hopcroft's problem over finite fields

Reminder of Theorem 1.5 *For every fixed prime p and constant k, the randomized communication complexity of the two-party Hopcroft problem over $\mathbb{F}_{p^k}$ is $\Theta(d^{(p-1)k})$.*

Proof. Let $\mathbb{F} = \mathbb{F}_{p^k}$. First we give the protocol achieving the upper bound. Fix an element $\alpha \in \mathbb{F}$ and consider the mapping $T_\alpha : \mathbb{F} \to \mathbb{F}$ defined as $T_\alpha(\beta) = \alpha\beta$. Recall $\mathbb{F}$ is a vector space over $\mathbb{F}_p$ of dimension k, and T_α is a linear transformation in this vector space. Fixing a basis in $\mathbb{F}$, T_α corresponds to a matrix M_α of dimension $k \times k$, and each element $\beta \in \mathbb{F}$ corresponds a vector X_β of dimension $1 \times k$.

This motivates the following reduction. For vector $u \in U$, $u = (u_1, u_2, \ldots, u_d)$, we convert it to $u' = (M_{u_1}, M_{u_2}, \ldots, M_{u_d})$, which can be seen as a matrix of dimension $k \times dk$. Similarly for vector $v \in V$, $v = (v_1, v_2, \ldots, v_d)$, we convert it to $v' = (X_{v_1}, X_{v_2}, \ldots, X_{v_d})$, which can be seen as a matrix of dimension $1 \times dk$. The goal becomes to determine whether there is a pair u' and v' such that $u' \cdot v'^T$ is the all-zero $k \times 1$ vector. Letting the k rows of u' be $u'_1, u'_2, \ldots, u'_k$, consider the polynomial:

$$\begin{aligned} &Q(u'_{1,1}, u'_{1,2}, \ldots, u'_{k,1}, \ldots, u'_{k,dk}, v'_1, \ldots, v'_{dk}) \\ =&\prod_{i=1}^{k} \left(1 - \langle u'_i, v' \rangle^{p-1}\right) \\ =&\prod_{i=1}^{k} \left(1 - \left(\sum_{j=1}^{dk} u'_{i,j} v'_j\right)^{p-1}\right) \end{aligned}$$

It is not hard to verify that $u' \cdot v'^T$ is the all-zero vector if and only if the above polynomial expression evaluates to a nonzero value over $\mathbb{F}_p$. The number of monomials in Q is at most $\left((dk)^{p-1} + 1\right)^k = O(d^{(p-1)k})$.[8] Now by Lemma 3.1, the problem of finding vectors on which Q is nonzero can be reduced to finding non-orthogonal pair in $\mathbb{F}_p$ of $O(d^{(p-1)k})$ dimensions. By Corollary 3.1, we get a protocol using $O(d^{(p-1)k})$ communication bits for this problem.

To obtain the lower bound, recall that by Theorem 1.1 it is sufficient to construct a MV family over $\mathbb{F}_{p^k}$ with dimension d of size $\Omega\left(d^{(p-1)k}\right)$. Recall the finite field $\mathbb{F}_{p^k}$ is isomorphic to polynomials over $\mathbb{F}_p$ modulo an irreducible polynomial of degree k. In the following, we use polynomials in x of degree at most $k-1$ to represent elements of $\mathbb{F}_{p^k}$.

For any $S \subseteq [d]$ with $(p-1)k$ elements, define $u_S \in \{0,1\}^d$ to be the complement of the indicator vector of S: $u_S[i] = 0$ if and only if $i \in S$. Define $v_S \in \mathbb{F}^d$ as follows: start with the indicator vector v of S, then put 1 in the first $(p-1)$ nonzero entries of v, the polynomial x in the next $(p-1)$ nonzero entries, the polynomial x^2 in the next $(p-1)$ nonzero entries, and so on, up to x^{k-1} if necessary. (For example, for $p = 3$, the vector $[0,1,0,1,1,0,1,1]$ becomes $[0,1,0,1,x,0,x,x^2]$.)

Observe that $\langle u_S, v_S \rangle = 0$ over $\mathbb{F}_{p^k}$, because the non-zero entries are disjoint in the two vectors. For arbitrary $S \ne S' \subseteq [d]$, we have $\langle u_{S'}, v_S \rangle = \sum_{i \in T} v_S[i]$ for some non-empty subset T of S. By construction, every non-empty subset of components of v_S sums to a nonzero value over $\mathbb{F}_{p^k}$. Therefore the sequence of vectors $\{u_S\}$ and $\{v_S\}$ is a MV family of size $\binom{d}{(p-1)k} \ge \Omega(d^{k(p-1)})$, and the proof is complete. □

Remark 4.1. *For $d \ge (p-1)k$, the lower bound on $\mathbf{MV}\left(\mathbb{F}_{p^k}, d\right)$ is $\binom{d}{(p-1)k} = \Omega((\frac{d}{(p-1)k})^{(p-1)k})$, and the upper bound is $O(\binom{dk+(p-1)k}{(p-1)k}) = O((\frac{e \cdot (k+1)d}{(p-1)k})^{k(p-1)})$. That is, for non-constant p and k, the upper and lower bounds on communication are off by a $(e(k+1))^{k(p-1)}$ factor.*

Reminder of Theorem 1.6 *For every prime p and $k \in \mathbb{N}$, there is a randomized algorithm for $\mathbf{HOPCROFT}_{\mathbb{F}_{p^k}}$ running in $O(n \cdot d^{(p-1)k})$ time.*

Proof. Similar to the proof of Theorem 1.5, we use the same reduction and polynomial and apply Lemma 3.1 to reduce it to finding a non-orthogonal pair. By Proposition 3.1, we have an algorithm for this problem which runs in $O(n \cdot d^{(p-1)k})$ time. □

[7] The idea is to simply partition the d_0 variables into at most $m-1$ groups of $q = \lceil d_0/(m-1) \rceil$ variables each, then compute the OR function exactly on each group with a polynomial of degree q, summing the result [Bar92].

[8] By combining some of the terms with identical v-values, the number of terms can be bounded from above by $\binom{dk+(p-1)k}{(p-1)k}$.

REMINDER OF THEOREM 1.7 *Suppose there is an $\varepsilon > 0$ and a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(x)/(x/\log x) \to 0$ and for infinitely many $\mathbb{F} = \mathbb{F}_{p^k}$, $\mathbf{HOPCROFT}_{\mathbb{F}}$ is solvable in time $n^{2-\varepsilon} \cdot d^{f((p-1)k)}$. Then SETH is false.*

Proof. In order to apply Theorem 3.1, we only need to construct low degree polynomials computing the OR function in finite fields.

We first use a folklore construction of a low-degree polynomial computing the OR function over $\mathbb{F}$. Recall $\mathbb{F}_{p^k}$ is isomorphic to polynomials over $\mathbb{F}_p$, modulo an irreducible polynomial of degree k. We use polynomials in x of degree at most $k-1$ to represent elements of $\mathbb{F}_{p^k}$. To construct a polynomial for OR on d_0 variables $y_1, \ldots, y_{d_0}$, we divide the variables into $(p-1)k$ groups, with at most $\lceil d_0/(p-1)k \rceil$ variables per group. For each group i, we construct a polynomial P_i of degree at most $\lceil d_0/(p-1)k \rceil$ which equals to 0 if all variables in group i are 0, and equals to 1 otherwise. Define the polynomial P with degree $\lceil d_0/(p-1)k \rceil$ (in y), as:

$$P(y_1, \ldots, y_{d_0}) = \sum_{i=0}^{k-1} \left(\sum_{j=i(p-1)+1}^{(i+1)(p-1)} P_j \right) x^i.$$

Then $P(y_1, \ldots, y_{d_0}) = 0$ if and only if all y_i equal 0. Same as the proof for Theorem 1.4, we can apply Theorem 3.1, and SETH is false. □

5 Hopcroft's problem over $\mathbb{Z}_6$

REMINDER OF THEOREM 1.8 *Every randomized protocol for the two-party Hopcroft problem over $\mathbb{Z}_6$ requires $d^{\Omega\left(\frac{\log d}{\log\log d}\right)}$ bits of communication.*

Proof. [Gro00, DGY11] proved that $\mathbf{MV}(6, d) \geq 2^{\Omega\left(\frac{\log^2 d}{\log\log d}\right)}$, so the result follows from Theorem 1.1. □

We remark that an communication-efficient protocol for two-party Hopcroft problem over $\mathbb{Z}_6$ will actually give an upper bound for $\mathbf{MV}(\mathbb{Z}_6, d)$, which still has a large gap between lower and upper bounds.

REMINDER OF THEOREM 1.9 *If $\mathbf{HOPCROFT}_{\mathbb{Z}_6}$ is solvable in $n^{2-\varepsilon} \cdot d^{o_d\left(\frac{\log d}{(\log\log d)^2}\right)}$ time for some $\varepsilon < 1$, then $\mathbf{HOPCROFT}_{\mathbb{Z}}$ over vectors in $\{0,1\}^d$ can be solved in time $n^{2-\delta} 2^{o(d)}$ for some $\delta < 1$.*

REMINDER OF COROLLARY 1.1 *If $\mathbf{HOPCROFT}_{\mathbb{Z}_6}$ is solvable in $n^{2-\varepsilon} \cdot d^{o_d\left(\frac{\log d}{(\log\log d)^2}\right)}$ time for some $\varepsilon < 1$, then SETH is false.*

Proof. [of Theorem 1.9 and Corollary 1.1] As before, it suffices to give a low degree polynomial computing the OR function over $\mathbb{Z}_6$. Beigel, Barrington, and Rudich [BBR94] constructed a multilinear polynomial P of degree $O(\sqrt{d_0})$ on d_0 binary variables, computing the OR function modulo 6. In this case, the number of monomials $d \leq \binom{d_0}{\sqrt{d_0}} = 2^{O(\sqrt{d_0}\log d_0)}$.

Therefore if we could solve $\mathbf{HOPCROFT}_{\mathbb{Z}_6}$ in time $n^{2-\varepsilon} \cdot d^{o\left(\frac{\log d}{(\log\log d)^2}\right)} = n^{2-\epsilon} \cdot 2^{o(d_0)}$, then by Theorem 3.1, SETH is false. □

6 Output sensitive and communication efficient matrix multiplication

Our final contribution is a surprisingly simple algorithm for matrix multiplication which has low communication complexity and is output-sensitive, building on some of our ideas for detecting orthogonal pairs.

REMINDER OF THEOREM 1.10 *Let $\mathbb{F}$ be an arbitrary finite field, let $A \in \mathbb{F}^{n \times d}$, and let $B \in \mathbb{F}^{d \times n}$. After $O(nd \log(|\mathbb{F}|\, n))$ preprocessing time (and no communication), Alice and Bob can compute every nonzero entry of $A \cdot B$ whp, with only $O(d(\log n)(\log(|\mathbb{F}|\, n)))$ communication and time delay per nonzero.*

Proof. Without loss of generality, assume $|\mathbb{F}| \geq n^5$, otherwise we can take a sufficiently large extension field of $\mathbb{F}$ in the following algorithm (recall that an irreducible polynomial of degree k can be generated with randomness in $\tilde{O}(k)$ time, so such extension fields can be efficiently obtained).

We wish to find all pairs $(i, j) \in [n]^2$ such that the inner product of i-th row of A (call it a_i) and j-th column of B (call it b_j) is nonzero. Let $i_1, i_2, j_1, j_2 \in [n]$, with $i_1 < i_2$, $j_1 < j_2$. Alice and Bob can test whether the submatrix of $A \cdot B$ with indices in $[i_1, i_2] \times [j_1, j_2]$ contains a nonzero entry, by having Alice choose $u = (u_{i_1}, \ldots, u_{i_2}) \in \mathbb{F}^{i_2 - i_1 + 1}$ uniformly at random, Bob choose random $v = (v_{j_1}, \ldots, v_{j_2}) \in \mathbb{F}^{j_2 - j_1 + 1}$, having Alice compute $a = \sum_{k=i_1}^{i_2} u_k \cdot a_k$, Bob compute $b = \sum_{k=j_1}^{j_2} v_k \cdot b_k$, and test if $\langle a, b \rangle = 0$ with $O(d(\log|\mathbb{F}| + \log n))$ communication. If the submatrix of $A \cdot B$ with indices in $[i_1, i_2] \times [j_1, j_2]$ is all-zero, then $\langle a, b \rangle = 0$; if there is a nonzero, then $\langle a, b \rangle \neq 0$ with probability at least $1 - 1/|\mathbb{F}|$.

Therefore, in $\tilde{O}((|i_2 - i_1 + 1| + |j_2 - j_1 + 1|) \cdot d)$ time, we can detect a nonzero in a submatrix of $|i_2 - i_1 + 1| \cdot |j_2 - j_1 + 1|$ entries. We can amortize the work over all possible intervals, as follows. First, observe that by the union bound, the guarantee that the submatrix of $A \cdot B$ with indices in $[i_1, i_2] \times [j_1, j_2]$ is all-zero if and only if $\langle a, b \rangle = 0$ holds, with probability at least $1 - n^4/|\mathbb{F}| \geq 1 - 1/n$, for all intervals $[i_1, i_2]$, $[j_1, j_2]$. Alice and Bob generate $u = (u_1, \ldots, u_n), v = (v_1, \ldots, v_n) \in \mathbb{F}^n$ at random, and compute all prefix sums

$$S_i = \sum_{k=1}^{i} u_k \cdot a_k, \quad T_j = \sum_{k=1}^{j} v_k \cdot b_k,$$

for $i, j = 0, \ldots, n$ (we set $S_0 = T_0 = 0$). This can be done in $O(nd \log |\mathbb{F}|)$ time. (Note these could be efficiently computed in parallel [Ble93].) Now for any desired intervals $[i_1, i_2], [j_1, j_2]$, Alice and Bob can detect a nonzero in the $[i_1, i_2] \times [j_1, j_2]$ submatrix of AB using $O(d(\log |\mathbb{F}| + \log n))$ communication and time: Alice sends $a = S_{i_2} - S_{i_1 - 1}$, Bob sends $b = T_{j_2} - T_{j_1 - 1}$, and they compute $\langle a, b \rangle$.

Given the ability to query nonzeroes in arbitrary intervals, we use binary search to recursively find all nonzero entries in $A \cdot B$. At each stage of the recursion, Alice and Bob are considering some interval $[i_1, i_2] \times [j_1, j_2]$; initially, $i_1 = j_1 = 1$ and $i_2 = j_2 = n$. Letting $i_m = \lfloor (i_2 - i_1 + 1)/2 \rfloor$ and $j_m = \lfloor (j_2 - j_1 + 1)/2 \rfloor$, they use four queries to detect nonzeroes among the four intervals

$$[i_1, i_m] \times [j_1, j_m], \ [i_1, i_m] \times [j_m + 1, j_2]$$

$$[i_m + 1, i_2] \times [j_1, j_m], \ [i_m + 1, i_2] \times [j_m + 1, j_2],$$

recursing on those intervals containing nonzeroes. For each nonzero entry (i, j), at most $\log n$ queries are needed to isolate the entry; then Alice and Bob can simply send a_i and b_j in extra $O(d \log |\mathbb{F}|)$ communication and time. □

Observe the above algorithm makes sense over $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{C}$ as well, with comparable running time (in terms of the number of arithmetic operations).

Reminder of Corollary 1.2 *There is an algorithm for listing all incidences over a given set of n points and n hyperplanes in $(\mathbb{F}_{p^k})^d$ that runs in $\tilde{O}((n+m) \cdot d^{k(p-1)})$ time, where m is the total number of incidences.*

Proof. Using the construction in the proof of Theorem 1.5, the problem of listing all incidences in $(\mathbb{F}_{p^k})^d$ can be reduced to listing all pairs $u_i \in U, v_j \in V$ such that $\langle u_i, v_j \rangle \neq 0$ for some $U, V \subseteq (\mathbb{F}_p)^{d^{(p-1)k}}$ with $|U| = |V| = n$. The reduction takes $\tilde{O}(n \cdot d^{k(p-1)})$ time. Let A be the $n \times d^{k(p-1)}$ matrix with ith row equal to $u_i \in U$, and B be the $d^{k(p-1)} \times n$ matrix with jth column equal to $v_j \in V$. The algorithm of Theorem 1.10 finds all nonzeroes in $A \cdot B$, corresponding to all pairs u_i, v_j that $\langle u_i, v_j \rangle \neq 0$, in time $\tilde{O}((n + m) d^{k(p-1)})$. □

7 Conclusion

In this paper, we have shown how Hopcroft's problem, a fundamental problem in computational geometry, reveals interesting layers of tractability and difficulty when we vary the algebraic structures involved. Along the way, we have given a simple combinatorial and output-sensitive method for matrix multiplication. There are still gaps in the overall picture which are worth further investigation. We conclude with a few open questions:

- *What is the exact relationship between the two-party Hopcroft communication problem over $\mathcal{R}$ and the sizes of matching vector families over $\mathcal{R}$?* Currently, we know they are related to within a factor of d, but perhaps tighter bounds can be proved. The only case we know how to completely settle is where $\mathcal{R} = \mathbb{F}_2$: we can prove that the deterministic communication complexity of two-party Hopcroft is at least $d^2/4 - o(d)$, and by bounds on MV families, it is $O(d^2)$.

- *Can we find algorithms for Hopcroft's problem that break the mold of the communication problem?* All of our algorithms stem from studying the two-party communication problem, then efficiently implementing the underlying protocols. It seems obvious that algorithms which can examine both sets of vectors multiple times before making decisions would be only more powerful. Of course, we know that significantly faster algorithms that go beyond the communication bounds would break the Strong Exponential Time Hypothesis.

- *Is the Strong Exponential Time Hypothesis actually true?* We find it hard to believe that Hopcroft's problem over (say) $\mathbb{F}_{31^{100}}$ would be significantly easier than the same problem over $\mathbb{Z}_6$, but that must be true if SETH holds. Just as studying $\mathbb{Z}_m$ for non-prime-power m was the key to developing subexponential locally decodable codes, studying Hopcroft's problem over $\mathbb{Z}_m$ may lead to faster SAT algorithms.

References

[Bar92] David A. Mix Barrington. Some problems involving Razborov-Smolensky polynomials. In *Proceedings of the London Mathematical Society Symposium on Boolean Function Complexity*, pages 109–128. Cambridge University Press, 1992.

[BBR94] David A. Mix Barrington, Richard Beigel, and Steven Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Comput. Complexity*, 4:367–382, 1994.

[BDL13] Abhishek Bhowmick, Zeev Dvir, and Shachar Lovett. New lower bounds for matching vector codes. In *STOC*, pages 823–832, 2013.

[Ble93] Guy E. Blelloch. Prefix sums and their applications. In J. Reif, editor, *Synthesis of Parallel Algorithms*. Morgan Kaufmann, 1993.

[CDL+12] Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. In *IEEE Conference on Computational Complexity*, pages 74–84, 2012.

[Cha93] Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete & Computational Geometry*, 9:145–158, 1993.

[CIP09] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Parameterized and Exact Computation*, pages 75–85. Springer, 2009.

[CKN13] Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast Hamiltonicity checking via bases of perfect matchings. In *STOC*, pages 301–310, 2013.

[CNP+11] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, J.M.M. van Rooij, and J.O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS*, pages 150–159, 2011.

[CW09] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *STOC*, pages 205–214, 2009.

[Cyg12] Marek Cygan. Deterministic parameterized connected vertex cover. In *Algorithm Theory–SWAT 2012*, pages 95–106. Springer, 2012.

[DGY11] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM J. Comput.*, 40(4):1154–1178, 2011.

[DKM06] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM J. Comput.*, 36:132–157, 2006.

[Dvi12] Zeev Dvir. Incidence theorems and their applications. *Foundations and Trends in Theoretical Computer Science*, 6(4):257–393, 2012.

[DW10] Evgeny Dantsin and Alexander Wolpert. On moderately exponential time for SAT. In *Proc. 13th International Conference on Theory and Applications of Satisfiability Testing*, pages 313–325, 2010.

[Efr12] Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, 41(6):1694–1703, 2012.

[Eri95] Jeff Erickson. On the relative complexities of some geometric problems. In *Proceedings of the 7th Canadian Conference on Computational Geometry*, pages 85–90, 1995.

[Eri96] Jeff Erickson. New lower bounds for Hopcroft's problem. *Discrete & Computational Geometry*, 16(4):389–418, 1996.

[FHV13] Henning Fernau, Pinar Heggernes, and Yngve Villanger. A multivariate analysis of some DFA problems. In *Proceedings of LATA*, pages 275–286, 2013.

[Fre77] Rusins Freivalds. Probabilistic machines can use less running time. In *IFIP Congress*, pages 839–842, 1977.

[Gro00] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20(1):71–86, 2000.

[HKN12] Sepp Hartung, Christian Komusiewicz, and André Nichterlein. Parameterized algorithmics and computational experiments for finding 2-clubs. In *Parameterized and Exact Computation*, pages 231–241. Springer, 2012.

[Ind05] Piotr Indyk. Output-sensitive algorithm for matrix multiplication. *unpublished manuscript*, 2005.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[KS92] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

[Lin11] Andrzej Lingas. A fast output-sensitive algorithm for boolean matrix multiplication. *Algorithmica*, 61(1):36–50, 2011.

[LMS11] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs on bounded treewidth are probably optimal. In *SODA*, pages 777–789, 2011.

[Mat93] Jirí Matousek. Range searching with efficient hiearchical cutting. *Discrete & Computational Geometry*, 10:157–182, 1993.

[Mat02] Jiri Matousek. *Lectures on Discrete Geometry*. Springer Graduate Texts in Mathematics, 2002.

[Pag12] Rasmus Pagh. Compressed matrix multiplication. In *ITCS*, pages 442–451, 2012.

[PP12] Marcin Pilipczuk and Michał Pilipczuk. Finding a maximum induced degenerate subgraph faster than 2^n. In *Parameterized and Exact Computation*, pages 3–12. Springer, 2012.

[PW10] Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In *SODA*, pages 1065–1075, 2010.

[Raz92] Alexander A. Razborov. On the distributional complexity of disjointness. *TCS*, 106(2):385–390, 1992.

[RW13] Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *STOC*, pages 515–524, 2013.

[Sar06] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, pages 143–152, 2006.

[Val12] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *FOCS*, pages 11–20, 2012.

[Wil05] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005.

A Appendix: Hopcroft's problem and the Strong Exponential Time Hypothesis

A variant of the following lemma appeared in [Wil05].

LEMMA A.1. ([WIL05]) *Suppose there is a $\delta > 0$ and an algorithm $\mathcal{A}$ such that for all $c \geq 1$, $\mathcal{A}$ solves Hopcroft's problem on n vectors in $\{0,1\}^{c \log n}$ over the integers, in $O(n^{2-\delta})$ time. Then the Strong Exponential Time Hypothesis is false.*

Proof. Recall that, to refute the Strong Exponential Time Hypothesis, we need to show that there is a $\delta' > 0$ such that for every k, k-SAT is in $(2-\delta')^n$ time.

Suppose we have a $\delta > 0$ and an algorithm for Hopcroft's problem as hypothesized. Given a k-CNF formula F on n variables, we first apply the Sparsification Lemma [IPZ01] to F, which for any desired $\varepsilon > 0$ generates $t \leq 2^{\varepsilon n}$ different k-CNF formulas $F_1, \ldots, F_t$ such that F is satisfiable if and only if some F_i is satisfiable, and the number of clauses of each F_i is at most $f(k,\varepsilon)n$ for a fixed function f. This reduction runs in $O(2^{\varepsilon n})$ time.

Setting $\varepsilon < \delta/2$ and $c = 2f(k,\varepsilon)$, we can embed the problem of satisfying a given F_i into Hopcroft's problem, as follows. We split the set of n variables into two sets of at most $\lceil n/2 \rceil$ variables each, and enumerate all partial assignments to the two sets, creating two lists L_1 and L_2 of $O(2^{n/2})$ size. For each partial assignment a from one of the lists, we associate a Boolean vector v_a with cn components, where the jth component of v_a is 1 if and only if the jth clause of F_i is not satisfied by a. Observe that there is a satisfying assignment to F_i if and only if there is a vector $v_a \in L_1$ and $w_{a'} \in L_2$ such that $\langle v_a, w_{a'} \rangle = 0$ (over the integers).

Applying the hypothesized algorithm for Hopcroft's problem on all $F_1, \ldots, F_t$, our satisfiability algorithm runs in $O(2^{\varepsilon n} \cdot (2^{n/2})^{2-\delta}) \leq O(2^{(\varepsilon+1-\delta/2)n})$ time. Since $\varepsilon < \delta/2$, this completes the proof. □