

## The Distance to an Irreducible Polynomial

Michael J. Mossinghoff

**ABSTRACT.** An old problem of P. Turán asks if every polynomial with integer coefficients lies close to an irreducible polynomial of the same degree or less, where the distance between two polynomials  $f$  and  $g$  is measured as the sum of the absolute values of the coefficients of  $f - g$ . We develop some algorithms to answer this question in the affirmative for monic polynomials of degree at most 34, showing that an irreducible polynomial can always be found within distance 4 in this case, and in fact distance 3 suffices for degree at most 18. This improves some results of Bérczes and Hajdu. We also employ a probabilistic model to make some predictions for polynomials with larger degree, and conjecture that distance 4 suffices in general.

### 1. Introduction

For a polynomial  $f(x) = \sum_{k=0}^n a_k x^k$  with degree  $n$  and integer coefficients, let  $L(f)$  denote its *length*,

$$L(f) := \sum_{k=0}^n |a_k|,$$

and let

$$\|f\|^2 := \sum_{k=0}^n a_k^2.$$

More than 40 years ago, Turán [5] asked if every polynomial in  $\mathbb{Z}[x]$  is near an irreducible polynomial, where the distance between polynomials is measured by the length. Specifically, he asked if there exists an absolute constant  $C$  such that for every  $f \in \mathbb{Z}[x]$  there exists an irreducible polynomial  $g \in \mathbb{Z}[x]$  with  $\deg(g) \leq \deg(f)$  and  $L(f - g) \leq C$ . Note that certainly  $C \geq 2$ , since one may take  $f(x) = x^n$  when  $n$  is odd or  $f(x) = x^{n-2}(x^2 + x - 1)$  when  $n \geq 4$  is even.

Turán's problem remains unsolved, although a number of partial results are known. In 1970, Schinzel [6] proved that such a constant does exist if one allows the neighboring irreducible polynomial to have larger degree. In particular, Schinzel showed that one may take  $C = 3$  if one permits

$$\deg(g) \leq e^{(5n+7)(\|f\|^2+3)},$$

---

2000 *Mathematics Subject Classification.* Primary: 11C08; Secondary: 11R09, 11Y40.

*Key words and phrases.* Turán's problem, irreducible polynomial, distance.

Research supported in part by NSA grant number H98230-08-1-0052.

and in fact  $C = 2$  suffices for polynomials with nonzero constant term. Recently, Banerjee and Filaseta [1] improved this by showing that an irreducible polynomial  $g$  with distance at most 3 always exists with the bound on  $\deg(g)$  growing only linearly in  $n$ ; an exponential dependence on  $\|f\|^2$  remains. More precisely, they showed that one may take  $C = 3$  provided that one allows

$$\deg(g) \leq 8 \max\{n + 3, n_0\} 5^{8\|f\|^2 + 9},$$

where  $n_0$  is an effectively computable constant.

In another direction, in 1997 and 1998 Bérczes and Hajdu [2, 3] answered Turán's question for polynomials with small degree through explicit computations. They showed that one may take  $C = 4$  for monic polynomials  $f$  with degree  $n \leq 24$ , and that  $C = 3$  suffices for  $n \leq 12$ . Their method in fact establishes the former bound for polynomials whose leading coefficient  $a_n$  is odd, and the latter one whenever  $3 \nmid a_n$ .

In this article, we implement more efficient algorithms for investigating Turán's problem, and use them to answer this question for polynomials up to degree 34. In particular, we prove the following theorem.

**THEOREM 1.** *If  $f \in \mathbb{Z}[x]$  has odd leading coefficient and degree  $n \leq 34$ , then there exists an irreducible polynomial  $g \in \mathbb{Z}[x]$  with  $\deg(g) = n$  and  $L(f - g) \leq 4$ . Further, if the leading coefficient of  $f$  is not a multiple of 3 and  $n \leq 18$ , then such a polynomial  $g$  exists with  $L(f - g) \leq 3$ .*

Section 2 describes some additional notation and summarizes the algorithm of Bérczes and Hajdu. Section 3 details the new algorithms, and section 4 summarizes the results we obtain. Last, section 5 analyzes our results and compares them with a probabilistic model for Turán's problem. In particular, based on the experimental evidence and this model, we conjecture that  $C = 4$  suffices in Turán's problem.

## 2. The Method of Bérczes and Hajdu

Following [2], we introduce some additional notation. First, for a positive integer  $n$ , let  $c_n$  denote the minimal integer with the property that for every monic polynomial  $f \in \mathbb{Z}[x]$  of degree  $n$  there exists a monic, irreducible polynomial  $g \in \mathbb{Z}[x]$  of degree  $n$  with  $L(f - g) \leq c_n$ . Using Eisenstein's criterion with  $p = 2$ , one easily verifies that  $c_n$  exists, and that in fact  $c_n \leq n + 1$ . Second, define  $c_n^*$  for monic polynomials  $f$  of degree  $n$  in a similar way, but this time require only that  $\deg(g) \leq n$ , and do not demand that  $g$  be monic. Clearly then  $c_n^* \leq c_n$ .

Next, consider a local version of Turán's problem. For a prime number  $p$  and a polynomial  $h \in \mathbb{F}_p[x]$ , let  $L_p(h)$  denote a local version of the length function. This is defined just like the global length function  $L$ , provided we choose each coefficient of  $h$  from the interval  $(-p/2, p/2]$ . This way,  $L_p(f - g)$  measures the minimal number of changes to the coefficients of  $f$  needed to create  $g$ . Then define  $c_n(p)$  in the same way as  $c_n$ , but for Turán's problem modulo  $p$ . Thus  $c_n(p)$  is the minimal integer with the property that for every monic  $f \in \mathbb{F}_p[x]$  with degree  $n$  there exists a monic, irreducible  $g \in \mathbb{F}_p[x]$  satisfying  $L_p(f - g) \leq c_n(p)$ . Since a monic polynomial  $g$  is irreducible in  $\mathbb{Z}[x]$  if it is irreducible in  $\mathbb{F}_p[x]$  for a prime  $p$ , it follows immediately that  $c_n \leq c_n(p)$ , for any prime  $p$ . We may therefore study Turán's problem for monic polynomials, where one wishes to bound  $c_n^*$ , by investigating various local versions of this problem, and determine bounds on  $c_n(p)$ , for several primes  $p$ .

As in [2], we remark that while one may define  $c_n^*(p)$  in an analogous way to  $c_n^*$  by relaxing the restrictions on  $g$ , one can no longer transfer irreducibility so easily from a local setting to the global one. For example, consider  $f(x) = x^6 + 2x^5 - 2x^4 + x^2 - x - 1$ , which is reducible in  $\mathbb{Z}[x]$ . Working modulo 2, we see  $g(x) = x^2 + x + 1$  is irreducible and  $L_2(f - g) = 1$ , but lifting  $g$  to the integer polynomial nearest  $f$  produces  $2x^5 - 2x^4 + x^2 - x - 1 = (x^2 - x + 1)(2x^3 - 2x - 1)$ . Thus, it is not apparent if  $c_n^* \leq c_n^*(p)$  must necessarily hold, and in this paper we restrict to monic polynomials  $g$  with the same degree as  $f$  in the local version of Turán's problem.

Bérczes and Hajdu computed  $c_n(2)$  for  $n \leq 24$  and  $c_n(3)$  for  $n \leq 12$ . We describe their algorithm briefly. For  $p = 2$ , they employ two sizable tables to determine the parity of the number of monomials of a polynomial  $f \in \mathbb{F}_2[x]$  of degree  $n$  in constant time. One table in essence provides the parity of the high-degree terms; the other handles the low-degree terms. Since an irreducible polynomial in  $\mathbb{F}_2[x]$  must necessarily consist of an odd number of monomials, it suffices to test only single- and triple-coefficient adjustments to  $f$  for irreducibility if  $f$  has an even number of monomials, and to test if  $f$  itself is irreducible, or double-coefficient adjustments to  $f$ , if  $f$  has odd parity. Any polynomial failing these tests then necessarily has distance greater than 3 to an irreducible polynomial. They then apply this test to each polynomial  $f$  of prescribed degree  $n$  having constant term 1. Clearly, the distance for a polynomial  $f$  with  $f(0) = 0$  is one larger than the distance for  $f + 1$ , so this method determines if there are any polynomials of degree  $n$  in  $\mathbb{F}_2[x]$  with distance greater than 4 to an irreducible polynomial modulo 2.

Their algorithm implements one additional optimization involving a time-space trade-off: Each time a polynomial  $g$  is tested for irreducibility, the result is stored in a table. This prevents testing the same polynomial for irreducibility several times in the course of a search, but necessitates the creation of a table with  $2^{n-2}$  entries, since only polynomials with odd parity and constant term 1 are tested.

Bérczes and Hajdu implemented a similar algorithm for  $p = 3$ , although no special effort was made in this case to filter out polynomials having a linear factor. All the methods were coded in Maple, which was used for the irreducibility tests mod  $p$ . Only one timing benchmark was supplied in [2, 3]: The case  $n = 22$  with  $p = 2$  required 180 hours of CPU time on a SUN SPARCstation 10, whose processor typically ran at 36 MHz.

In the prior work, no calculations were performed for  $p > 3$ , but the authors opined in [2] that using additional primes would likely produce better bounds.

### 3. New Algorithms

In this section, we describe some new algorithms for computing  $c_n(p)$  more efficiently. The new algorithms offer several improvements over the prior method, including:

- a more efficient mechanism for determining the parity of  $L_2(f)$  without using auxiliary storage tables,
- a more efficient strategy for remembering prior irreducibility tests,
- a native irreducibility tester whose amortized cost is much less than the cost of testing each polynomial for irreducibility independently, and
- a method for computing  $c_n(p)$  for an arbitrary small prime  $p$ .

Below, we first describe the specialized algorithm for  $p = 2$ , then the general method for larger  $p$ . In each case, the algorithm has two principal phases, given a positive integer  $n$ . First, we determine all monic irreducible polynomials of degree  $n$  in  $\mathbb{F}_p[x]$ . Second, for each monic  $f \in \mathbb{F}_p[x]$  with degree  $n$ , we compute the distance from  $f$  to an irreducible polynomial modulo  $p$ .

In each method, we represent a polynomial  $f \in \mathbb{F}_p[x]$  with the integer whose base- $p$  expansion is precisely the sequence of coefficients of  $f$ . This is simply the integer  $f(p)$  (performing the arithmetic in  $\mathbb{Z}$ ), assuming that each coefficient of  $f$  is the least nonnegative residue mod  $p$ . This representation allows rapid comparison of two polynomials, and computing  $f(x) \pm x^k$  is also a fast operation, assuming relevant powers of  $p$  have been pre-computed. When  $p = 2$ , adding two polynomials is also very fast, as the coefficients can be summed in parallel by computing the exclusive or (xor) of the corresponding integer values.

**3.1. The Case  $p = 2$ .** To determine the set of irreducible polynomials in  $\mathbb{F}_2[x]$  of degree  $n$ , we consider each such  $f$  with  $f(0) = 1$  in turn, and test all possible irreducible polynomials  $g$  of degree at most  $n/2$  as possible divisors. This test can be performed very rapidly for divisors  $g$  of small degree by arranging the computation in an appropriate way.

Let  $S$  be a set of irreducible polynomials of small degree in  $\mathbb{F}_2[x]$ . The precise contents of  $S$  can be varied to tune the performance of the algorithm for different  $n$ , but usually  $S$  contains all the irreducible polynomials mod 2 for a particular range of degrees. Before beginning the search for the irreducible polynomials of degree  $n$ , we compute the remainder of  $x^k$  mod  $g$ , for each  $g \in S$  and each  $k$  with  $0 \leq k \leq n$ . Each remainder has degree less than  $\deg(g)$ , and so can be encoded in  $\deg(g)$  bits. We pack these bit sequences into a number of 64-bit long words. For example, the first word for a particular dividend  $x^k$  has two bits to encode its remainder modulo  $x^2 + x + 1$ , then three bits each for  $x^3 + x + 1$  and  $x^3 + x^2 + 1$ , then twelve bits for the three irreducible polynomials of degree 4, and so on. The first vector holds remainders for fourteen polynomials; the second, ten additional polynomials; the third, nine more, etc. (It is not necessary to store remainders for the two linear irreducible polynomials.)

We employ these remainder vectors when testing each polynomial for irreducibility. We begin with  $f(x) = x^n + 1$ , and compute the remainder of  $f$  modulo each  $g \in S$  by computing the xor of the bit vectors for  $x^n$  and 1. We then use a Gray code on the middle  $n$  bits of  $f$  to iterate over the  $2^n$  polynomials of degree  $n$  with  $f(0) = 1$ . In this way, each polynomial we consider differs from its predecessor in a single bit position. Thus, at each iteration we update the remainder sequence by simply computing the xor of the current remainder vectors with the bit sequences corresponding to the single altered monomial in  $f$ .

Each candidate polynomial  $f$  can then be tested for divisibility by some  $g \in S$  by scanning the remainder vectors. This is quite fast with the use of appropriate mask vectors that isolate the fields of interest. As an added benefit, using a Gray code allows us to maintain the parity of  $L_2(f)$  with no auxiliary data structures, since this parity simply alternates with each iteration. Thus, there is no need to test for divisibility by  $x + 1$ .

We use a number of different sets  $S$  for different values of  $n$ , but for many larger runs,  $S$  consisted of all the irreducible polynomials in  $\mathbb{F}_2[x]$  with degree between 2 and 11. The remainders for these 410 polynomials pack into 70 long words. For

$n = 34$ , the irreducible polynomials of degree 12 were added to  $S$  as well, bringing the total to 745 polynomials, packed into 137 long words.

For each polynomial that survives the divisibility test by polynomials in  $S$ , we then use ordinary trial division to check for other possible factors up to degree  $n/2$ . The irreducible polynomials of degree up to  $n/2$  that are not in  $S$  are computed first by using this same method. The integer representation for polynomials in  $\mathbb{F}_2[x]$  helped to speed the checks here, since trial division can be encoded by using simple bit shifts and xor operations. We also experimented with replacing this trial division step with the computation of the greatest common divisor with  $x^{2^k-1} + 1$  for  $k \leq n/2$ , but this alternative strategy was not as efficient in practice.

All the irreducible polynomials of degree  $n$  constructed in the first phase of the algorithm must be saved for use in the second part, where distances are calculated. The data structure housing these polynomials must ensure fast insertion and search times, and we also require efficient use of space. Since the number of irreducible polynomials mod 2 is well known, a hash table satisfies all these requirements. Using an open-addressing scheme with a load factor of  $2/3$ , and employing a double-hashing scheme to resolve collisions, on average a polynomial can be inserted or tested for membership in the table in just three probes. Storing all the irreducible polynomials mod 2 of degree  $n$  then requires approximately  $3 \cdot 2^{n+1}/n$  bytes of memory. Other data structures are much less efficient in their memory usage: A balanced binary tree would need 2.5 times as much space, owing to the overhead for storing the pointers, and a `set` from the C++ Standard Template Library would need 5 times the space.

Since we store only the irreducible polynomials, and not the results of irreducibility tests for all polynomials with odd length, our strategy uses significantly less space than the method of [2, 3].

We use our hash table during the first phase of the algorithm as well, in order to exploit some symmetry. For a polynomial  $f(x) = \sum_{k=0}^n a_k x^k$  in  $\mathbb{F}_2[x]$ , let  $f^*$  denote its *reciprocal*, obtained by reversing the order of the coefficients, so  $f^*(x) = \sum_{k=0}^n a_{n-k} x^k$ . Clearly,  $f^*$  is irreducible if and only if  $f$  is irreducible, so we can avoid the trial divisions on  $f$  if  $f^*$  is already in the hash table. This optimization saves nearly half the computation time of the first phase of the algorithm for sizable  $n$ .

Some special considerations apply for larger values of  $n$ .

- (1) When  $n \geq 32$ , the integer corresponding to a polynomial of degree  $n$  no longer fits in a 32-bit word. Of course, we could simply employ a long 64-bit word instead, since our computers have a 64-bit architecture, but it is best to avoid this for two reasons. First, processors are often significantly more efficient using 32-bit operations, and our experience affirms this for this algorithm. Second, our storage requirement would double, and memory is already critical for the calculations when  $n$  is large.

However, every polynomial we consider has leading and trailing coefficient 1, so there is no need to store these bits. This allows us to handle degrees 32 and 33 using 32-bit arithmetic with only minor changes to our code. For degree 34, we maintain two hash tables: one for irreducible polynomials that contain the monomial  $x^{33}$ , the other for those where this term is absent. Each table then stores the remaining 32 bits of each

polynomial. (Here, it is helpful that  $x^{34} + x^{33} + 1$  is reducible modulo 2, as  $x^4 + x^3 + 1$  is a factor, so the value 0 can be used to indicate an empty location in both hash tables.)

- (2) The search for irreducible polynomials can be distributed across multiple computers, with each processor handling the polynomials with a certain prescribed sequence of high-order monomials. We split the searches for  $n = 32$  and  $n = 33$  across 16 computers each, and the one for  $n = 34$  across 64 machines. Each irreducible polynomial is simply printed to a file as it is found. In fact, we need only print one of  $f$  or its reciprocal—whichever has the smaller representative integer value. The hash table is constructed from this output in the second phase of the algorithm. (The second phase is executed on a single machine.)
- (3) There are simply too many irreducible polynomials of degree 34 for a computer with two gigabytes of RAM to store in real memory. For this case, then, we store only one of  $f$  or  $f^*$  (whichever one has the smaller corresponding integer value). This halves the space requirement, while increasing the computation time of the second phase of the method, since now two hash table lookups may be required when testing a polynomial for irreducibility. Also in this case, we test only one of  $f$  and  $f^*$  for its distance to an irreducible polynomial, in order to speed the second phase of the algorithm.

For the second phase of the algorithm, we again employ a Gray code to iterate over the  $2^{n-1}$  polynomials in  $\mathbb{F}_2[x]$  with degree  $n$  and constant term 1. We thus automatically maintain the parity of each polynomial tested. For a polynomial  $f$  of odd length, we test if  $f$  is irreducible, then check two-bit changes to  $f$  if needed, then four-bit changes after that if required. A similar strategy is employed if  $f$  has even length. We use the revolving door algorithm of Nijenhuis and Wilf [4] to enumerate the subsets of monomials of the various required sizes in an efficient way. This method constructs the collection of subsets of fixed size from a parent set in such a way that each subset built differs from its predecessor in a minimal way—one element is removed from the subset, and another is added to take its place.

**3.2. The Case  $p \geq 3$ .** The algorithm for larger primes has the same overall strategy. We determine all the monic, irreducible polynomials in  $\mathbb{F}_p[x]$  of prescribed degree  $n$ , store their corresponding integer values in a hash table, and then for each  $f \in \mathbb{F}_p[x]$  of degree  $n$ , compute its distance to an irreducible polynomial. Since the number of monic irreducible polynomials we need to store now grows like  $p^n/n$ , space requirements are now critical much sooner, so there is less need to optimize the computation times.

In the first phase of the algorithm, we do not account for divisibility by linear factors in the enumeration, and we use the `DetIrredTest` method of the NTL library [7] to test irreducibility of each polynomial independently. This method implements an algorithm of Shoup [8]. In the second phase of the algorithm, we use a  $p$ -ary Gray code to enumerate the polynomials, and revolving door to enumerate subsets of particular sizes. However, the optimization exploiting the symmetry with  $f^*$  is no longer available, since we now restrict to monic polynomials. In addition, we can no longer assume that the maximum distance to an irreducible polynomial occurs for polynomials with constant term 0 (this is true for the case  $p = 3$ , but

need not be for  $p \geq 5$ .) The second phase must therefore test all  $p^n$  polynomials in  $\mathbb{F}_p[x]$  of degree  $n$ .

It is helpful to perform the distance check on a family of  $p$  polynomials at a time. If  $f \in \mathbb{F}_p[x]$  has degree  $n$  and  $f(0) = 0$ , we consider the polynomials  $f(x) + k$  with  $0 \leq k < p$  as a group. First, we use the hash table to identify any irreducible polynomials in this group, and mark these with the value 0. Then set  $i = 0$  and perform each of the following actions.

- (1) If for some  $k$  the polynomial  $f(x) + k \pm 1$  is marked with the integer  $i$ , then mark  $f(x) + k$  with the integer  $i + 1$ .
- (2) For any unmarked polynomial  $f(x) + k$ , test if it has distance  $i + 1$  from an irreducible polynomial, and mark it with the integer  $i + 1$  if this is the case.

Then increment  $i$  and repeat these steps, halting when each polynomial in the group has been marked.

For  $p \geq 5$ , the space requirement on the hash table dictates the largest degree  $n$  we are able to handle. However, for  $p = 3$ , where we searched through degree 18, the hash table requires only 125 megabytes of memory, but our computation time was already 59 hours for this case. Thus, it seems possible that a specialized algorithm for  $p = 3$ , which takes advantage of the fact that we need to find all the irreducible polynomials in  $\mathbb{F}_3[x]$  of a particular degree, would be able to search somewhat further. It may be possible to account for the linear factors in an efficient way too, similar to the specialized code for  $\mathbb{F}_2[x]$ .

#### 4. Results

We use our algorithm for  $p = 2$  to verify that  $c_n(2) \leq 4$  for  $n \leq 34$ . Tables 1 and 2 summarize the results of this computation. Table 1 shows the number of polynomials in  $\mathbb{F}_2[x]$  of fixed degree  $n$  with distance  $k$  from an irreducible polynomial, for  $0 \leq k \leq 4$ .

Table 2 displays some extremal polynomials, and indicates in a certain sense how close we come to finding a polynomial with distance greater than 4 to an irreducible polynomial. For  $f \in \mathbb{F}_2[x]$  of degree  $n$ , let  $m_n(k, f)$  denote the number of monic irreducible polynomials  $g$  of degree  $n$  having  $L_2(f - g) = k$ , and let  $m_n(k)$  designate the minimal value of  $m_n(k, f)$  over all  $f$  of degree  $n$  that have distance  $k$  from an irreducible polynomial. Owing to parity considerations, the value of  $m_n(3)$  can then be viewed as a measure of proximity to detecting a polynomial with distance 5 from an irreducible polynomial, and likewise  $m_n(4)$  is an indication of how close we come to finding a polynomial with distance 6.

Table 2 shows the values of  $m_n(k)$  for  $k = 3$  and  $k = 4$  over the degrees we consider. The last column of the table shows polynomials of degree  $n$  that have maximal distance from an irreducible polynomial, and for which the number of irreducible polynomials at this distance is minimized. All such polynomials are shown for each degree, except the corresponding reciprocal polynomial  $(f + 1)^* + 1$  is not shown whenever  $f$  is listed. Thus, the polynomials listed of degree  $n$  with  $4 \leq n \leq 7$  or  $n = 9$  have distance 3 from an irreducible polynomial, and  $m_n(3)$  irreducible polynomials at distance 3. The polynomials exhibited for  $n = 8$  and  $n \geq 10$  have distance 4, and the minimal number of irreducible polynomials at this distance. (For these degrees we do not display the extremal polynomials at

TABLE 1. Number of polynomials  $\mathbb{F}_2[x]$  of degree  $n$  with distance  $k$  from an irreducible polynomial.

$n$	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
2	1	2	1		
3	2	4	2		
4	3	7	5	1	
5	6	14	10	2	
6	9	25	23	7	
7	18	50	46	14	
8	30	93	97	35	1
9	56	184	200	72	
10	99	354	412	158	1
11	186	696	836	328	2
12	335	1355	1709	693	4
13	630	2662	3450	1434	16
14	1161	5209	6983	2983	48
15	2182	10291	14119	6093	83
16	4080	20296	28520	12472	168
17	7710	40144	57492	25392	334
18	14532	79263	115735	51809	805
19	27594	157191	233075	104953	1475
20	52377	311095	468485	213193	3426
21	99858	617282	941854	431294	6864
22	190557	1224987	1892449	872165	14146
23	364722	2432502	3800210	1761802	29372
24	698870	4830908	7627472	3557700	62266
25	1342176	9605110	15309366	7172106	125674
26	2580795	19096115	30711741	14458317	261896
27	4971008	37992980	61605396	29115884	532460
28	9586395	75616382	123552456	58601346	1078877
29	18512790	150521773	247713921	117913683	2208745
30	35790267	299734269	496589191	237136643	4491454
31	69273666	597046041	995369621	476695783	9098537
32	134215680	1189342142	1994652606	958141506	18615362
33	260300986	2369913037	3996794713	1925054259	37871597
34	505286415	4723495045	8007889511	3866439547	76758666

distance 3.) The polynomials shown for  $n = 2$  and  $n = 3$  have distance 2 from an irreducible polynomial, and in both cases  $m_n(2) = 2$ .

The entire calculation for  $n = 22$  with  $p = 2$  required just 2.4 seconds on a 2.4 GHz Intel-based Apple compute with two gigabytes of memory. For  $n = 34$ , the first phase of the algorithm required about 280 hours of CPU time, distributed across 64 PowerPC-based Apple computers, each running at 2.5 GHz, and the second phase completed in about 22 hours on the Intel-based Apple computer. The first phase of the program required about 37.5 hours for degree 32 and 79 hours for degree 33.

For  $p = 3$ , we verify that  $c_n(3) \leq 3$  for  $n \leq 18$ . Tables 3 and 4 summarize our results here in the same manner as Tables 1 and 2, with  $m_n(k)$  defined in the

TABLE 2. Extremal polynomials in  $\mathbb{F}_2[x]$ .

$n$	$m_n(3)$	$m_n(4)$	Extremal polynomials
2	—	—	$x^2$
3	—	—	$x^3, x^3 + x^2 + x$
4	3	—	$x^4 + x^2$
5	5	—	$x^5 + x$
6	5	—	$x^6 + x^4 + x^3 + x^2$
7	9	—	$x^7 + x^2, x^7 + x^4 + x^2 + x, x^7 + x^5 + x^4 + x,$ $x^7 + x^6 + x^4 + x^3 + x^2 + x$
8	5	17	$x^8$
9	10	—	$x^9 + x^7 + x^6 + x, x^9 + x^8 + x^7 + x^5 + x^4 + x^3 + x^2 + x$
10	10	35	$x^{10} + x^8 + x^7 + x^6 + x^4 + x^3 + x^2$
11	12	51	$x^{11} + x^6 + x^4 + x^3 + x^2$
12	10	48	$x^{12} + x^9 + x^7 + x^2 + x$
13	13	66	$x^{13}$
14	11	69	$x^{14} + x^{12} + x^8 + x^6 + x^2$
15	11	89	$x^{15} + x^{12} + x^{11} + x^6 + x^5 + x^3 + x,$ $x^{15} + x^{13} + x^{12} + x^9 + x^8 + x^6 + x^5 + x^3 + x^2$
16	9	92	$x^{16} + x^8 + x^4$
17	13	113	$x^{17} + x^{15} + x^{14} + x^{11} + x^5 + x^3 + x^2$
18	6	119	$x^{18} + x^{14} + x^{12} + x^9 + x^6 + x^4 + x^2$
19	13	144	$x^{19} + x^{16} + x^{14} + x^{12} + x^7 + x^4 + x^3 + x^2 + x$
20	13	153	$x^{20} + x^{14} + x^{10} + x^8 + x^6$
21	12	166	$x^{21} + x^{13} + x^{12} + x^9 + x^8$
22	13	195	$x^{22} + x^{18} + x^4, x^{22} + x^{21} + x^{20} + x^{19} + x^{14} +$ $+x^{13} + x^{12} + x^{10} + x^9 + x^8 + x^3 + x^2 + x$
23	14	214	$x^{23} + x^{20} + x^{18} + x^{17} + x^{13} + x^{12} + x^5 + x^4 + x^2$
24	10	196	$x^{24} + x^{16} + x^8$
25	15	258	$x^{25} + x^{23} + x^{20} + x^{18} + x^{17} + x^{15} + x^{14} +$ $+x^{12} + x^{11} + x^{10} + x^6 + x^5 + x^3 + x^2 + x$
26	14	277	$x^{26} + x^{25} + x^{24} + x^{20} + x^{19} + x^{18} + x^{16} +$ $+x^{14} + x^{12} + x^{10} + x^8 + x^7 + x^6 + x^2 + x$
27	15	291	$x^{27} + x^{24} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{17} +$ $+x^{14} + x^{13} + x^{12} + x^9 + x^7 + x^6 + x^5 + x^4 + x$
28	15	323	$x^{28} + x^{25} + x^{24} + x^{23} + x^{21} + x^{20} + x^{18} +$ $+x^{16} + x^{14} + x^{12} + x^9 + x^7 + x^6 + x^4 + x^2$
29	17	336	$x^{29} + x^{28} + x^{26} + x^{23} + x^{22} + x^{21} + x^{17} +$ $+x^{16} + x^{13} + x^{12} + x^8 + x^7 + x^6 + x^3 + x$
30	17	374	$x^{30} + x^{28} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} +$ $+x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^2$
31	16	406	$x^{31} + x^{28} + x^{27} + x^{26} + x^{23} + x^{20} +$ $+x^{18} + x^{13} + x^{11} + x^8 + x^5 + x^4 + x^3$
32	14	347	$x^{32} + x^{16} + x^4$
33	17	464	$x^{33} + x^{30} + x^{29} + x^{28} + x^{26} + x^{25} + x^{23} + x^{22} + x^{20} +$ $+x^{18} + x^{14} + x^{12} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3$
34	17	493	$x^{34} + x^{30} + x^{28} + x^{26} + x^{22} + x^{20} + x^{18} + x^{14} + x^8 + x^4 + x^2,$ $x^{34} + x^{31} + x^{27} + x^{26} + x^{25} + x^{22} + x^{21} + x^{18} +$ $+x^{15} + x^{13} + x^{10} + x^9 + x^6 + x^5 + x^4 + x^3 + x^2$

TABLE 3. Number of polynomials  $\mathbb{F}_3[x]$  of degree  $n$  with distance  $k$  from an irreducible polynomial.

$n$	$k = 0$	$k = 1$	$k = 2$	$k = 3$
2	3	6		
3	8	17	2	
4	18	53	10	
5	48	156	39	
6	116	460	153	
7	312	1411	462	2
8	810	4158	1583	10
9	2184	12477	5007	15
10	5880	37175	15942	52
11	16104	111045	49820	178
12	44220	331657	154857	707
13	122640	993839	475964	1880
14	341484	2968254	1466447	6784
15	956576	8891503	4480747	20081
16	2690010	26612560	13674807	69344
17	7596480	79665489	41684316	193878
18	21522228	238347569	126891552	659140

same way for polynomials in  $\mathbb{F}_3[x]$ . However, since the parity of the distance is not germane in this case, Table 4 displays just one value of  $m_n(k)$  for each  $n$ : the value where  $k$  is the maximum distance attained for that degree, so  $k = c_n(3)$ . All the extremal polynomials are shown for each degree, after accounting for certain symmetries. For example,  $f(-x)$  is not listed if  $f(x)$  is shown.

Tables 1 and 3 show that the calculations with  $p = 3$  produce improved upper bounds on  $c_n$  and  $c_n^*$  for  $n \in \{2, 4, 5, 6, 8\}$  and  $10 \leq n \leq 18$ , so one might hope that using additional primes would improve some of these bounds further. However, extensive computations with several larger primes always produce bounds that are the same as, or slightly worse than, those obtained using  $p = 3$ . Table 5 summarizes the results of our calculations for odd primes  $p \leq 31$ . Here, the integer  $N_2(p)$  denotes the largest degree  $n$  for which  $c_n(p) = 2$  for each prime  $p$ . The last row shows the largest degree  $N(p)$  tested for each of these primes. Throughout, we find that  $c_n(p) = 3$  for  $N_2(p) < n \leq N(p)$ .

Last, we note that Bérczes and Hajdu conjectured in [2, 3] that for each  $n \geq 10$  there exists a polynomial  $f \in \mathbb{F}_2[x]$  of degree  $n$  having maximal distance from an irreducible polynomial and for which the polynomial  $f(x) + x^n + 1$  is irreducible mod 2. They verified this for  $n \leq 24$ ; our data affirm this conjecture for  $n \leq 34$ .

## 5. Analysis

Let  $r_p(n, k)$  denote the proportion of monic polynomials in  $\mathbb{F}_p[x]$  having distance  $k$  from an irreducible polynomial. It is well known that the number of monic irreducible polynomials modulo  $p$  of degree  $n$  is given by

$$\frac{1}{n} \sum_{d|n} \mu(d) p^{n/d},$$

TABLE 4. Extremal polynomials in  $\mathbb{F}_3[x]$ .

$n$	$c_n(3)$	$m_n(c_n(3))$	Extremal polynomials
2	1	1	$x^2, x^2 + x$
3	2	4	$x^3, x^3 + x$
4	2	3	$x^4 + x^3 - x^2 + x$
5	2	3	$x^5 + x^4 - x^2 - x, x^5 + x^4 + x^3$
6	2	1	$x^6 + x^5 + x^4 - x^3 + x^2$
7	3	23	$x^7 + x^4 + x$
8	3	36	$x^8 + x^7 + x^6 - x^5 - x^3 + x^2 + x$
9	3	34	$x^9 + x^3 - x$
10	3	37	$x^{10} + x^9 + x^6 - x^5 - x^4 - x^3 + x^2$
11	3	36	$x^{11} - x^9 + x^8 + x^7 - x^6 - x^4 + x^3 - x$
12	3	35	$x^{12} - x^{10} + x^9 - x^8 - x^7 - x^5 - x^4 - x^2$
13	3	46	$x^{13} - x^{11} - x^9 + x^7 - x^3 - x$
14	3	45	$x^{14} + x^{12} + x^{11} + x^{10} + x^9 + x^7 + x^6 - x^4 + x^3 - x^2 - x$
15	3	42	$x^{15} + x^{13} + x^9 - x^7 + x^5 + x^3$
16	3	42	$x^{16} + x^{15} + x^{14} + x^{13} + x^{12} - x^{11} - x^{10} +$ $+x^9 - x^8 + x^6 - x^3 + x^2 + x$
17	3	47	$x^{17} + x^{15} - x^7 - x^5 + x^2$
18	3	48	$x^{18} + x^{17} - x^{15} + x^{14} + x^{13} - x^{12} + x^{10} - x^9 - x^5 + x^4 - x^3,$ $x^{18} + x^{15} + x^{14} - x^{13} + x^{12} - x^{11} +$ $+x^{10} + x^7 + x^6 - x^4 + x^3 - x^2 + x$

TABLE 5. Calculations with larger primes:  $c_n(p) = 2$  for  $2 \leq n \leq N_2(p)$  and  $c_n(p) = 3$  for  $N_2(p) < n \leq N(p)$ .

$p$	3	5	7	11	13	17	19	23	29	31
$N_2(p)$	6	4	3	2	2	2	2	2	2	2
$N(p)$	18	12	10	8	7	7	7	6	6	6

where  $\mu(\cdot)$  is the Möbius function, so

$$r_p(n, 0) \approx \frac{1}{n}.$$

We can use this estimate to approximate the value of  $r_p(n, k)$  for various distances  $k$ , if we assume that the irreducible polynomials in  $\mathbb{F}_p[x]$  of fixed degree are evenly distributed, after accounting for some evident necessary conditions. We describe some of these approximations in this section, and compare the predictions of the model with the data that we obtained. We can then use our model to assess the probability that a polynomial exists with even larger distance to an irreducible polynomial.

We consider the case  $p = 2$ . Certainly every irreducible polynomial of degree  $n > 1$  in  $\mathbb{F}_2[x]$  has  $f(0) = 1$  and  $L_2(f)$  odd, and we suppose that the irreducible polynomials of degree  $n$  are distributed uniformly among the polynomials satisfying these simple constraints. Suppose  $f \in \mathbb{F}_2[x]$  has degree  $n$  and  $f(0) = 1$ . We may then compute the probability that  $f$  has distance  $k$  from an irreducible polynomial, for a fixed nonnegative integer  $k$ .

Suppose first that  $f$  has odd length. The conditional probability that  $f$  is irreducible, given that  $f(0) = 1$  and  $L_2(f)$  is odd, is approximately  $\frac{2^n}{n} \cdot \frac{1}{2^{n-2}} = 4/n$ . If  $f$  is reducible, then it has distance at least 2 to an irreducible polynomial. The probability that a polynomial of the form  $f(x) + x^i + x^j$  is reducible, with  $1 \leq i < j < n$ , is about  $1 - 4/n$ , so the probability that  $f$  has distance  $k \geq 4$ , assuming that  $f$  itself is reducible, is approximately

$$\left(1 - \frac{4}{n}\right)^{\binom{n-1}{2}} \approx e^{2-2n} \left(1 - \frac{8}{3n}\right).$$

Thus, the conditional probability that  $f$  has distance 2, assuming that  $f(0) = 1$  and  $L_2(f)$  is odd, is approximately  $1 - 4/n$ .

If  $f$  has even length, then the probability that  $f$  is not adjacent to an irreducible polynomial is about

$$\left(1 - \frac{4}{n}\right)^{n-1} \approx e^{-4} \left(1 - \frac{4}{n}\right).$$

Since the probability that  $f$  has distance greater than 3 is negligible at

$$\left(1 - \frac{4}{n}\right)^{n-1+\binom{n-1}{3}} \approx \exp\left(-\frac{2n^2}{3} + \frac{8n}{3} - \frac{62}{9}\right),$$

we estimate the probability that  $f$  has distance 1, conditioned on the assumptions that  $f(0) = 1$  and  $L_2(f)$  is even, by

$$1 - e^{-4} \left(1 - \frac{4}{n}\right).$$

We can now account for the polynomials with constant term 0 in a simple way. The probability that such a polynomial has distance  $k$  from an irreducible polynomial is the same as that for a polynomial with constant term 1 and opposite parity to have distance  $k - 1$ . We then obtain the following approximations for the proportions  $r_2(n, k)$ :

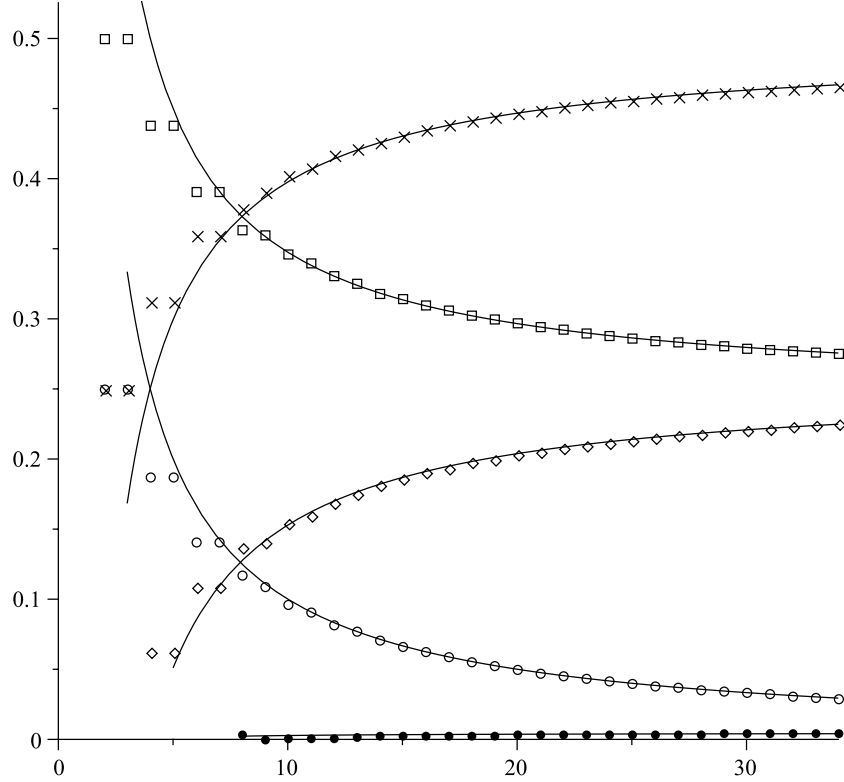
$$\begin{aligned} r_2(n, 0) &\approx \frac{1}{n}, \\ r_2(n, 1) &\approx \frac{1 - e^{-4}}{4} + \frac{1 + e^{-4}}{n}, \\ r_2(n, 2) &\approx \frac{2 - e^{-4}}{4} - \frac{1 - e^{-4}}{n}, \\ r_2(n, 3) &\approx \frac{1 + e^{-4}}{4} \left(1 - \frac{4}{n}\right), \\ r_2(n, 4) &\approx \frac{e^{-4}}{4} \left(1 - \frac{4}{n}\right). \end{aligned}$$

As  $n$  grows large, we thus expect about 24.54% of the polynomials of degree  $n$  to have distance 1, about 49.54% to have distance 2, approximately 25.46% to possess distance 3, and the remaining .46% to have distance 4.

Figure 1 shows that these predicted proportions fit our data reasonably well for  $n \leq 34$ . Here, the curves show the predicted proportion for each  $k$ , and the points display the experimental data, with a different symbol used for each value of  $k$ .

We can use our model to estimate the probability that a polynomial in  $\mathbb{F}_2[x]$  exists with distance  $k \geq 5$  from an irreducible polynomial. We expect that the total

FIGURE 1. Predicted proportions versus experimental data for distances in  $\mathbb{F}_2[x]$  ( $k = 0$ : open circles;  $k = 1$ : boxes;  $k = 2$ : crosses;  $k = 3$ : diamonds;  $k = 4$ : filled circles).



number of polynomials  $f \in \mathbb{F}_2[x]$  with  $f(0) = 1$ , odd distance  $k \geq 5$ , and degree  $n \geq 35$  is

$$\sum_{n \geq 35} 2^{n-2} \left( \left(1 - \frac{4}{n}\right)^{\binom{n-1}{1} + \binom{n-1}{3}} + \left(1 - \frac{4}{n}\right)^{1 + \binom{n-1}{2}} \right) < 10^{-18},$$

and that the total number of polynomials with even distance  $k \geq 6$  and degree  $n \geq 35$  is

$$\sum_{n \geq 35} 2^{n-2} \left( \left(1 - \frac{4}{n}\right)^{\binom{n-1}{1} + \binom{n-1}{3}} + \left(1 - \frac{4}{n}\right)^{1 + \binom{n-1}{2} + \binom{n-1}{4}} \right) < 10^{-306}.$$

It seems reasonable to conjecture then that  $c_n(2) = 4$  for  $n \geq 10$ , and thus that  $c_n^* \leq 4$  for  $n \geq 10$  in Turán's problem.

One may obtain estimates for  $r_p(n, k)$  for other fixed primes  $p$  in a similar way. We briefly discuss just the limiting case as  $p$  grows large. Since the polynomials with constant term 0 have diminishing influence on the values of  $r_p(n, k)$  as  $p \rightarrow \infty$ , we may ignore this special case in the asymptotic analysis. Clearly, we have  $r_p(n, 0) \approx 1/n$ . For distances  $k > 0$ , we must account for altering coefficients by

$\pm 1$ , so we expect the probability that a monic polynomial is reducible, and is not adjacent to an irreducible polynomial, to be approximately

$$\left(1 - \frac{1}{n}\right)^{2n+1} \approx e^{-2} \left(1 - \frac{2}{n}\right).$$

Further, we expect the probability that the distance exceeds 2 to be

$$\left(1 - \frac{1}{n}\right)^{n^2+n+1} \approx e^{-n-3/2} \left(1 - \frac{11}{6n}\right).$$

We therefore expect that

$$\begin{aligned} \lim_{n \rightarrow \infty} \lim_{p \rightarrow \infty} r_p(n, 1) &= 1 - e^{-2} = 0.8646\dots, \\ \lim_{n \rightarrow \infty} \lim_{p \rightarrow \infty} r_p(n, 2) &= e^{-2} = 0.1353\dots, \end{aligned}$$

and

$$\lim_{p \rightarrow \infty} r_p(n, 3) \approx e^{-n-3/2}.$$

Thus, for large  $p$ , our heuristics indicate that there should be about  $e^{-3/2} \left(\frac{p}{e}\right)^n$  monic polynomials with distance 3 from an irreducible polynomial as  $n$  grows large, and that distance  $k \geq 4$  is extremely unlikely for sizable  $n$ . This then supports a conjecture that in fact  $c_n^* \leq 3$  for large  $n$  in Turán's problem.

### Acknowledgements

I thank the University of South Carolina for their hospitality, as this research was performed during my visit there in 2008–09. I especially thank Michael Filaseta for many helpful suggestions. I also thank the Centre for Interdisciplinary Research in the Mathematical and Computational Sciences (IRMACS) at Simon Fraser University for computational resources, as the irreducible polynomials in  $\mathbb{F}_2[x]$  of degrees 32, 33, and 34 were calculated in distributed computations there.

### References

- [1] P. Banerjee and M. Filaseta, *On a polynomial conjecture of Pál Turán*, Acta Arith., to appear.
- [2] A. Bérczes and L. Hajdu, *Computational experiences on the distances of polynomials to irreducible polynomials*, Math. Comp. **66** (1997), no. 217, 391–398. MR1377660 (97c:11035)
- [3] ———, *On a problem of P. Turán concerning irreducible polynomials*, Number Theory: Diophantine, Computational and Algebraic Aspects (Eger, Hungary, 1996) (K. Györy, A. Pethő, and V. T. Sós, eds.), de Gruyter, Berlin, 1998, pp. 95–100. MR1628834 (99f:11032)
- [4] A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms*, 2nd ed., Academic Press, New York, 1978. MR510047 (80a:68076)
- [5] A. Schinzel, *Reducibility of polynomials and covering systems of congruences*, Acta Arith. **13** (1967), 91–101. MR0219515 (36 #2596)
- [6] ———, *Reducibility of lacunary polynomials, II*, Acta Arith. **16** (1970), 371–392. MR0265323 (42 #233)
- [7] V. Shoup, *NTL: A library for doing number theory*. [www.shoup.net/ntl](http://www.shoup.net/ntl).
- [8] ———, *Fast construction of irreducible polynomials over finite fields*, J. Symbolic Comput. **17** (1994), no. 5, 371–391. MR1289997 (95k:11156)

DEPARTMENT OF MATHEMATICS, BOX 6996, DAVIDSON COLLEGE, DAVIDSON, NORTH CAROLINA 28035-6996

*E-mail address*: [mimossinghoff@davidson.edu](mailto:mimossinghoff@davidson.edu)