MATHEMATICS OF COMPUTATION
Volume 78, Number 268, October 2009, Pages 2259–2267
S 0025-5718(09)02237-6
Article electronically published on March 26, 2009

AN ALGORITHM FOR FINDING A NEARLY MINIMAL BALANCED SET IN $\mathbb{F}_p$

ZHIVKO NEDEV

ABSTRACT. For a prime p, we call a non-empty subset S of the group $\mathbb{F}_p$ balanced if every element of S is the midterm of a three-term arithmetic progression, contained in S. A result of Browkin, Diviš and Schinzel implies that the size of a balanced subset of $\mathbb{F}_p$ is at least $\log_2 p+1$. In this paper we present an efficient algorithm which yields a balanced set of size $(1 + o(1)) \log_2 p$ as p grows.

1. INTRODUCTION

Let p be an odd prime. In this paper, we are interested in small subsets S of $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$, where each element of S is a midpoint between two other elements from S.

Definition. If for $S \subseteq \mathbb{F}_p$, $x \in \mathbb{F}_p$, $\{y, z\} \subseteq S \setminus \{x\}$, we have that $2x = y + z \pmod p$, then we say that x is *balanced* with respect to S. If $x \in S$, we also say that x is *balanced in* S. We call $\{y, z\}$ a *balancing pair* for x with respect to S.

Definition. We say that a set S of residues modulo p is *balanced* if all of its elements are balanced with respect to the set. *Unbalanced* sets have at least one element without a balancing pair.

It is easy to construct large balanced sets, but constructing balanced sets of small size is much more challenging. Small balanced sets are required in strategies for the family of combinatorial games analyzed in [6, 3, 4, 5].

Problem 1. Let $\alpha(p)$ denote the minimum cardinality of a balanced set modulo p. For a given prime p, what is the value of $\alpha(p)$ and how can we construct algorithmically a balanced set of small size?

From the definition of $\alpha(p)$ it is clear that every subset of $\mathbb{F}_p$ of size less than $\alpha(p)$ has at least one unbalanced element. In [1], Browkin, Diviš, and Schinzel prove that for any subset $S \subset \mathbb{F}_p$ of size $|S| < \log_2 p + 1$, there exists a $v \in \mathbb{F}_p$ represented uniquely as $w + x$, $w, x \in S$. Since Browkin et al. work with ordered representations, unique sums must be of the form $x + x$, $x \in S$. If S (and thus x) were balanced, we would have $2x = y+z, \{y, z\} \subseteq S\setminus\{x\}$, contradicting uniqueness. Thus, they prove that every S with $|S| < \log_2 p + 1$ is unbalanced. It follows that $\alpha(p) \geq \log_2 p + 1$.

In [7], Straus considers sets of residues modulo p with unique differences. Although his paper does not deal with balanced sets, a simple construction from

Received by the editor April 25, 2008 and, in revised form, October 29, 2008.
2000 *Mathematics Subject Classification.* Primary 11Y16.

his paper gives a balanced set of size $2\lfloor \log_2 p \rfloor + 1$. A second more complicated construction in the same paper provides a balanced set of size $(2 + o(1)) \log_3 p$.

In [6], Z. Nedev and A. Quas gave a proof of the lower bound that is more specific (for the field $\mathbb{F}_p$) and shorter than in [1], and presented an alternative simple construction of a balanced set with size $2\lfloor \log_2 p \rfloor + 1$. Furthermore, a better lower bound was discovered in [3].

In this paper, we present a polynomial algorithm for finding a balanced subset of $\mathbb{F}_p$ of a size slightly larger than the lower bound. Our algorithm has the following input and output:

Input: A prime number p and a real number $\epsilon > 0$.

Output: A balanced subset S of $\mathbb{F}_p$ of size $(1+\epsilon)\log_2 p + constant$.

By algorithmic construction we will then prove an upper bound: For every $\epsilon > 0$ there exists a positive integer n_ϵ such that for every prime $p > n_\epsilon$, $\alpha(p) < (1+\epsilon)\log_2 p$.

2. Algorithmic construction of small balanced sets, and an upper bound

We will demonstrate an algorithm for constructing small balanced sets, and thus prove the following theorem.

Theorem 1. *We have $\alpha(p) = (1 + o(1)) \log_2 p$ as $p \to \infty$.*

The following lemma is an obvious restatement of Theorem 1.

Lemma 2. *For every $\epsilon > 0$, there exists a constant positive integer m such that for every prime number p, there exists a balanced subset of $\mathbb{F}_p$ with size less than $(1+\epsilon)\log_2 p + m$.*

Throughout the algorithm we work in $\mathbb{F}_p$ except where otherwise specified. We first give a special case of the input p where the algorithm simplifies and the main idea is clearly seen. We then give the general algorithm and prove Lemma 2.

2.1. The simplified algorithm.

2.1.1. *The case when the input p is a "lucky" prime.* Let $n = \lfloor \log_2 p \rfloor$. We say that a prime $p > 2$ is a lucky input for our algorithm if either $\left\lceil \frac{3}{4}p \right\rceil$ or $\left\lfloor \frac{3}{4}p \right\rfloor$ equals 2^n.

To construct a balanced set, we start with the set $S = \{0, \left\lfloor \frac{p}{2} \right\rfloor, \left\lceil \frac{p}{2} \right\rceil, 2^n\}$. Notice that 0 is balanced by the pair $\{\left\lfloor \frac{p}{2} \right\rfloor, \left\lceil \frac{p}{2} \right\rceil\}$. Since p is an odd prime, we have two cases: If $p = 4 \cdot i + 1, i \in \mathbb{N}$, then $n = \log_2(3i+1)$, $2^n = \left\lceil \frac{3}{4}p \right\rceil$, and 2^n is balanced by $\{0, \left\lceil \frac{p}{2} \right\rceil\}$. If $p = 4 \cdot i - 1, i \in \mathbb{N}$, then $n = \log_2(3i)$, $2^n = \left\lfloor \frac{3}{4}p \right\rfloor$, and 2^n is balanced by $\{0, \left\lfloor \frac{p}{2} \right\rfloor\}$. Thus, in our initial S, the only unbalanced elements are $\left\lfloor \frac{p}{2} \right\rfloor$ and $\left\lceil \frac{p}{2} \right\rceil$, which we call the *core*, $C \stackrel{\text{def}}{=} \{\left\lfloor \frac{p}{2} \right\rfloor, \left\lceil \frac{p}{2} \right\rceil\}$. Our goal is to balance the core.

Example. $p = 43$ is a lucky prime because $\left\lfloor \frac{3}{4} \cdot 43 \right\rfloor = 2^5 = 32$. $S = \{0, 21, 22, 32\}$, $C = \{21, 22\}$, and $2^n = 32$ is balanced by $\{0, \left\lfloor \frac{p}{2} \right\rfloor\} = \{0, 21\}$.

At the beginning of the algorithm we let $L = 0$ and $R = 2^n$. We consider only the integers in the interval $[L, R]$. Letting $M = L + \frac{1}{2}(L+R)$, we divide the interval into two subintervals, $[L, M]$ and $[M, R]$. We then continue with two steps:

1) Add the midpoint M to S.
2) Update L and R to be the endpoints of the subinterval that contains the core C.

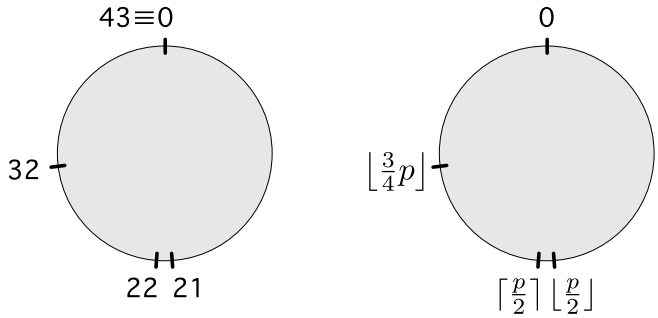

FIGURE 1. The initial S

Example (Continued). $L = 0$ and $R = 32$, so $M = 16$. The subinterval $[16, 32]$ contains the core $\{21, 22\}$, so $L \leftarrow 16$, $R \leftarrow 32$ and $S \leftarrow \{0, 21, 22, 32\} \cup \{16\}$.

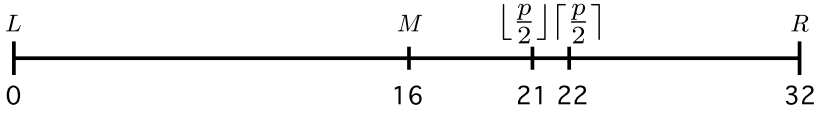

FIGURE 2. The interval $[L, R]$ before updating

Notice that in step 1, M is balanced by the pair $\{L, R\}$, which are themselves balanced. Thus in step 2, both the new L and R are balanced.

We repeat the above subdivision process until $R - L = 1$. At that point, we have $L = \lfloor \frac{p}{2} \rfloor$ and $R = \lceil \frac{p}{2} \rceil$. Since L and R begin balanced and remain so as we subdivide, we have balanced the core and constructed a balanced set S.

Example (Continued). For $p = 43$, we add to S (in the order produced by the algorithm): 16, 24, 20, 22, and 21. (Note that 21 and 22 were already in S.) The final balanced set $S = \{0, 16, 20, 21, 22, 24, 32\}$.

2.1.2. *What is the size of S?* Initially, we count the two elements L and R. We have that $R - L = 2^n$, so we execute n steps, adding one point to S each time. The core elements become midpoints, so they are counted among these n elements.

Therefore

$$|S| = n + 2 = \lfloor \log_2 p \rfloor + 2.$$

Thus, when p is a lucky prime, we obtain a stronger result than Lemma 2: m does not depend on ϵ but $m = 2$. Furthermore, it follows from the main theorem in [3] that $\log_2 p + 3 - \log_2 3 \leq |S|$. Thus we have

$$\log_2 p + 1.41503749 \leq \log_2 p + 3 - \log_2 3 \leq |S| = \lfloor \log_2 p \rfloor + 2 \leq \log_2 p + 2$$

and since $|S|$ must be an integer, there is only one possible value for $|S|$. We have therefore constructed a balanced set of minimum size.

2.2. **The algorithm for an arbitrary prime p.** Our objective is to prove Lemma 2 by means of algorithmic construction.

Proof of Lemma 2. Let $\epsilon > 0$.

We first consider the case when $p = 2$. Then $S = \{0, 1\}$ is the only balanced set, and $|S| = \log_2 p + 1$. Thus, in this case, Lemma 2 is satisfied so long as we let $m \geq 1$.

We now consider the general case when p is prime and $p > 2$. □

2.2.1. *Initial setup for the algorithm.* Let t be the smallest positive integer such that $\frac{1}{t+1} \leq \epsilon$. Thus $t = \left\lfloor \frac{1}{\epsilon} \right\rfloor$.

Our general algorithm for producing a small size balanced set begins almost identically as when p is lucky. We take $L = 0$ and let $R = \left\lceil \frac{3}{4}p \right\rceil$ if $p = 4 \cdot i + 1, i \in \mathbb{N}$, or $R = \left\lfloor \frac{3}{4}p \right\rfloor$ if $p = 4 \cdot i - 1, i \in \mathbb{N}$. For now, we take the core $C' = \{\left\lfloor \frac{p}{2} \right\rfloor, \left\lceil \frac{p}{2} \right\rceil\}$, and let $S = \{L, R, \left\lfloor \frac{p}{2} \right\rfloor, \left\lceil \frac{p}{2} \right\rceil\}$. As before, L and R are balanced from the beginning.

We also compute a real number $Q \stackrel{\text{def}}{=} \frac{2}{3}R$, dividing $[L, R]$ in a ratio of $2 : 1$. Q remains fixed throughout the algorithm. Notice that $\left\lfloor \frac{p}{2} \right\rfloor < Q < \left\lceil \frac{p}{2} \right\rceil$.

The lengths of the intervals $[L, R]$ will generally not be powers of 2 (and in fact may be odd numbers), so we will not be able to subdivide as before. However, at the cost of a small increase in the size of S, we can find intervals whose lengths are powers of 2 times a number.

Definition. An interval $[l, r]$ with $l, r \in \mathbb{N}$ is *t-even*, where t is a positive integer, if $r - l = 2^t c$, where $c \in \mathbb{N}$.

The algorithm's main part consists of the repetition of two main steps. In step A, we find a t-even interval at most half the size of the current interval, with the cost of adding two elements to S. Then during step B, we proceed as in the lucky case, subdividing the t-even interval t times and adding t elements to S. Because we want to repeat steps A and B, the ratio in which Q divides the interval found in each step A ideally should remain $2 : 1$. This is not always possible, but we will see that it is enough if the interval found in each step A is divided by Q in ratios of approximately either $2 : 1$ or $1 : 2$.

To achieve step A we will expand the core to $8k$ elements, where k is the smallest integer such that $2k - 1 > 2^t$ (thus $k = \left\lceil 2^{t-1} + 1/2 \right\rceil = 2^{t-1} + 1$), and add these new elements to S. These extra elements allow us to find t-even intervals. We denote these elements as follows:

$$C_{-4k} = \left\lfloor \frac{p}{2} \right\rfloor - 4k + 1, C_{-4k+1} = \left\lfloor \frac{p}{2} \right\rfloor - 4k + 2, \ldots, C_{-1} = \left\lfloor \frac{p}{2} \right\rfloor,$$
$$C_1 = \left\lceil \frac{p}{2} \right\rceil, C_2 = \left\lceil \frac{p}{2} \right\rceil + 1, \ldots, C_{4k} = \left\lceil \frac{p}{2} \right\rceil + 4k - 1$$

and let $C = \{C_{-4k}, C_{-4k+1}, \ldots, C_{-1}, C_1, \ldots, C_{4k-1}, C_{4k}\}$ be the new core. Notice that C has $4k$ elements on the left and right of Q.

When the new core does not fit within $[L, R]$, we take $S = \{L, L + 1, \ldots, R\}$ as our balanced set. A simple calculation shows that $|S| \leq R - L + 1 \leq 12k + 4$, and Lemma 2 is satisfied with $m = 12k + 4$.

Otherwise, let $S = \{L, R\} \cup C$. Since the elements in C are consecutive integers, only C_{-4k} and C_{4k} are unbalanced. As in the lucky case, our goal is to balance these two unbalanced elements. We now proceed to step A.

2.2.2. *Step A of the algorithm.*

Definition. Let $[l, r]$ be an interval, let $a, b \in \mathbb{N} \setminus \{0\}$, and let $\hat{Q} = l + \frac{a}{a+b}(r - l)$ so that $\hat{Q}$ divides $[l, r]$ in a ratio of $a : b$. We say that a real number Q divides $[l, r]$ in a ratio of *approximately* $a : b$ if $\left\lfloor \hat{Q} \right\rfloor = \lfloor Q \rfloor$ and $\left\lceil \hat{Q} \right\rceil = \lceil Q \rceil$.

Note. Initially, Q divides $[L, R]$ *exactly* in a ratio of $2 : 1$. After the first execution of step A, Q will divide the new interval $[L, R]$ in a ratio of *approximately* $2 : 1$, rather than exactly.

Moreover, after step B, Q will divide each new interval $[L, R]$ in a ratio of approximately either $1 : 2$ or $2 : 1$. The below procedure is for the ratio of $2 : 1$, but it is easily adapted when the ratio is $1 : 2$; all references to $2 : 1$ should be read as $1 : 2$, and $\hat{Q}$ should be redefined accordingly.

Our goal is to find a t-even interval no more than half the size of $[L, R]$ where Q divides the t-even interval in a ratio of approximately $2 : 1$.

For convenience we introduce two temporary markers: $Q_l \stackrel{\text{def}}{=} L + \frac{1}{2}(Q - L)$ and $Q_r \stackrel{\text{def}}{=} Q + \frac{1}{2}(R - Q)$.

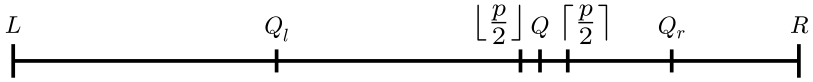

FIGURE 3. Q_l and Q_r

Notice that:

1) $Q_r - Q_l = \frac{1}{2}(R - L)$.
2) Q divides $[Q_l, Q_r]$ in a $2 : 1$ ratio.

We consider the $8k$ intervals $[L, C_{-4k}], [L, C_{-4k+1}], \ldots, [L, C_{4k}]$ listed in order of *increasing* length. Half of these have even lengths, and therefore their midpoints are integers. We denote these $4k$ integer midpoints by $l_1, l_2, \ldots, l_{4k}$, and note that each $l_{i+1} = l_i + 1$. There are $2k$ of these midpoints on the left and right of Q_l. Note also that, since $L \in S, C \subseteq S$, all of these midpoints are balanced with respect to S.

Similarly, we consider the $8k$ intervals $[C_{-4k}, R], [C_{-4k+1}, R], \ldots, [C_{4k}, R]$ listed in order of *decreasing* length. We obtain $4k$ midpoints $r_1, r_2, \ldots, r_{4k}$ with the property that each $r_{j+1} = r_j + 1$. There are $2k$ of these midpoints on the left and right of Q_r, and they are balanced with respect to S.

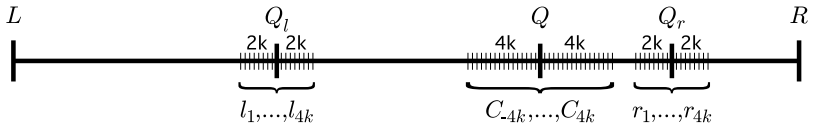

FIGURE 4. Midpoints and core

We want to update L and R such that $L \in \{l_{2k+1}, l_{2k+2}, \ldots, l_{4k}\}$ and $R \in \{r_1, r_{k+2}, \ldots, r_{2k}\}$, to achieve a reduction of the current interval by at least half.

We consider the set of intervals obtained by choosing a left endpoint $l \in \{l_{2k+1}, l_{2k+2}, \ldots, l_{3k}\}$ and a right endpoint $r \in \{r_{k+1}, r_{k+2}, \ldots, r_{2k}\}$. (We restrict ourselves to $l \leq l_{3k}$ and $r \geq r_{k+1}$ as we will later translate the interval $[l, r]$, and want freedom to translate while maintaining $l \in \{l_1, l_2, \ldots, l_{4k}\}$ and $r \in \{r_1, r_2, \ldots, r_{4k}\}$.) Among this set of intervals there are $2k-1$ distinct consecutive lengths. Since $2k-1 > 2^t$, at least one of these lengths will be a multiple of 2^t. Thus, we may choose a t-even interval $[l, r]$.

FIGURE 5. Possible endpoints for $[l, r]$

In order for Q to divide $[l, r]$ by a ratio of approximately $2:1$, we will translate the interval. Let $\hat{Q} = l + \frac{2}{3}(r - l)$ so that $\hat{Q}$ divides $[l, r]$ in a ratio of $2:1$. We can translate l, r, and $\hat{Q}$ in parallel in integer increments, maintaining $l \in \{l_1, l_2, \ldots, l_{4k}\}$ and $r \in \{r_1, r_2, \ldots, r_{4k}\}$.

Translating to the extreme right ($l = l_{4k}$), we have that $Q_l < l < Q < Q_r < r$. Since Q divides $[Q_l, Q_r]$ in a $2:1$ ratio and $\hat{Q}$ divides $[l, r]$ in a $2:1$ ratio, we have that $\hat{Q} > Q$. Similarly, translating to the extreme left ($r = r_1$), we have that $\hat{Q} < Q$.

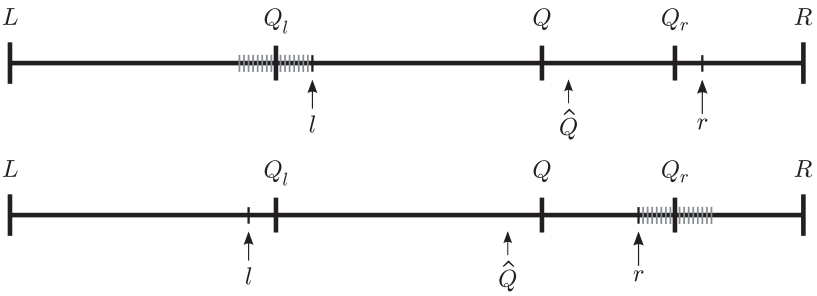

FIGURE 6. Extreme right and left translations

Thus, we can translate l, r, and $\hat{Q}$ such that $\hat{Q}$ and Q are in the same interval of length one with integer endpoints, i.e. $\lfloor \hat{Q} \rfloor = \lfloor Q \rfloor$ and $\lceil \hat{Q} \rceil = \lceil Q \rceil$. We achieve this by the translation:

$$\delta = \lfloor \hat{Q} \rfloor - \lfloor Q \rfloor,$$
$$l \leftarrow l - \delta; \quad r \leftarrow r - \delta; \quad \hat{Q} \leftarrow \hat{Q} - \delta.$$

Therefore, we have obtained l and r such that:

1) $(r - l) \leq \frac{1}{2}(R - L)$.
2) $(r - l)$ is t-even.
3) $\hat{Q}$ divides $[l, r]$ in a ratio of $2:1$.
4) Q divides $[l, r]$ in a ratio of approximately $2:1$, so $\lfloor \hat{Q} \rfloor = \lfloor Q \rfloor = \left\lfloor \frac{p}{2} \right\rfloor$ and $\lceil \hat{Q} \rceil = \lceil Q \rceil = \left\lceil \frac{p}{2} \right\rceil$.

If at this point $l \in C$ or $r \in C$ we do not perform step A, and instead proceed to section 2.2.4 to *finish the construction of* S. Otherwise, note that $C \subset [l, r]$.

Finally, we update $S \leftarrow S \cup \{l, r\}$; $L \leftarrow l$; $R \leftarrow r$ to obtain a new interval $[L, R]$ that is at most half the size of the original interval.

2.2.3. *Step B of our algorithm.* Since $[L, R]$ is t-even, we may perform at least t subdivision steps as in the lucky case. However, as it simplifies the proof of our lemma, we will only subdivide t times. If at any time during the subdivision process we have the midpoint $M \in C$, we immediately cease subdividing and proceed to section 2.2.4 to finish the construction of S. Otherwise, we obtain a new interval $[L, R]$ that is $\frac{1}{2^t}$ times smaller than the original. We then repeat step A.

Depending on the total number of subdivisions performed since the start of the algorithm, $\hat{Q}$ is always one of $L + \frac{1}{3}(R-L)$ (ratio $1:2$) or $L + \frac{2}{3}(R-L)$ (ratio $2:1$). If we have performed an even number of subdivisions, $\hat{Q}$ will divide the resultant interval in a $2:1$ ratio, and Q will divide it in *approximately* a $2:1$ ratio. If we have performed an odd number of subdivisions, the ratios will be $1:2$.

2.2.4. *Finishing the construction.* We say that we reach the core when the new element to be added to S is in C, and therefore already in S. This can happen in two cases; we will show that in both cases we have $R - L \leq 36k + 9$. We suppose that $\hat{Q}$ divides $[L, R]$ in a $2:1$ ratio; the results are analogous when the ratio is $1:2$.

Case 1) During step A we found the desired interval $[l, r]$, but $l \in C$ or $r \in C$.

Suppose that $r \in C$; the result is analogous when $l \in C$. We must have $r \in \{C_1, \ldots, C_{4k}\} \cap \{r_1, \ldots, r_{2k}\} \neq \emptyset$, so $Q_r - Q \leq 6k + 1$.

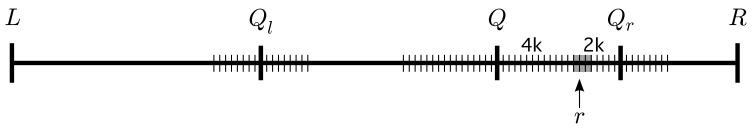

FIGURE 7. Case 1

Since Q divides $[L, R]$ in a ratio of approximately $2:1$, we have

$$\frac{1}{3}(R-L) = R - \hat{Q} \leq R - Q + 1 = 2(Q_r - Q) + 1 \leq 12k + 3.$$

Thus $R - L \leq 36k + 9$.

Case 2) During step B, $M \in C$.

We have $C_{-1} < Q < C_1$, so $C_{-1} \leq \hat{Q} \leq C_1$. Moreover, $C_{-4k} \leq M \leq C_{4k}$, so $|\hat{Q} - M| \leq 4k + 1$. Therefore

$$\frac{1}{6}(R-L) = \frac{2}{3}(R-L) - \frac{1}{2}(R-L) = |\hat{Q} - M| \leq 4k + 1.$$

Thus $R - L \leq 36k + 9$.

We can finish the construction by independently balancing C_{-4k} on the left and C_{4k} on the right. To balance C_{-4k}, we note that one of $[L, C_{-4k}]$ and $[L, C_{-4k+1}]$ will always have even length. Thus we may repeatedly subdivide: Let M be the midpoint of the even interval, and update $S \leftarrow S \cup \{M\}$; $L \leftarrow M$. We repeat until $C_{-4k} - L = 1$. Similarly, C_{4k} can be balanced using $[C_{4k-1}, R]$ and $[C_{4k}, R]$.

Since $R - L \le 36k + 9$, these two subdivision processes add no more than $36k + 9$ elements to S (and in fact add at most $O(\log_2 k)$ elements).

2.2.5. *What is the size of S?* We count as follows:

1) The two initial elements, $L = 0$ and $R \approx \frac{3}{4}p$.
2) The core, $|C| = 8k$.
3) The points added during steps A and B, calculated below.
4) At most $36k + 9$ elements from the final construction after reaching the core.

In order to calculate the required number of repetitions of step A followed by step B, we assume that we cease repetition when $R - L \le 36k + 9$, even if we have not yet reached the core.

Initially, $R - L \approx \frac{3}{4}p$. Step A reduces $R - L$ by a factor of at least $\frac{1}{2}$ and step B by exactly $\frac{1}{2^t}$. Let d be the smallest positive integer such that $(\frac{3}{4}p)(\frac{1}{2^{t+1}})^d \le 36k + 9$. Then

$$\frac{\log_2 p - \log_2(48k + 12)}{t + 1} \le d$$

and therefore

$$d = \left\lceil \frac{\log_2 p - \log_2(48k + 12)}{t + 1} \right\rceil \le \frac{\log_2 p}{t + 1} + 1.$$

The number of repetitions of step A followed by step B will be less than or equal to d. We add 2 points to S in step A, and t points in step B. Therefore we add at most $d(t+2)$ points during the repetitions of steps A and B. Note that we may have overcounted: if the subdivision midpoint reaches the core, the algorithm terminates before performing t subdivisions in the last step B.

Summing the above counts, we obtain the following bound:

$$\begin{aligned} |S| &\le d(t + 1) + d + 2 + 8k + (36k + 9) \\ &\le \left(\frac{\log_2 p}{t + 1} + 1\right)(t + 1) + \left(\frac{\log_2 p}{t + 1} + 1\right) + 44k + 11 \\ &< \left(\frac{\log_2 p}{t + 1} + 1\right)(t + 1) + (\epsilon \cdot \log_2 p + 1) + 44k + 11 \\ &= (1 + \epsilon)\log_2 p + m, \qquad m = t + 44k + 13. \end{aligned}$$

We now return to the two conditions on m that we mentioned earlier. When $p = 2$, we required that $m \ge 1$, and when $L \in C$ or $R \in C$ at the beginning of the algorithm, we required that $m \ge 12k + 4$. Since our choice of m satisfies these conditions, the lemma is satisfied in these cases.

Finally, we note that by our choice of t and k, $t = \lfloor \frac{1}{\epsilon} \rfloor$ and $k = 2^{t-1} + 1 = 2^{\left(\lfloor \frac{1}{\epsilon} \rfloor - 1\right)} + 1$. Thus $m = 11 \times 2^{t+1} + t + 57 = 11 \times 2^{\lceil \frac{1}{\epsilon} \rceil} + \lceil \frac{1}{\epsilon} \rceil + 56$. □

2.3. **Construction of a small balanced set for a given prime.** For a given prime p, we can iterate the algorithm to experimentally determine and construct the smallest possible balanced set obtainable by the algorithm. We perform the algorithm several times with $t = 1, 2, 3, \ldots$. For each value of t we will obtain a different balanced set. As t increases, the sizes of these sets will initially decrease. However, because the constant m (and the "core") grows exponentially with t, the sizes will eventually increase. Therefore, we can quickly find the optimal value of t for the given p.

3. Questions for further research

1) Experimental data suggests that for every prime $p > 2$, $\alpha(p) \leq \log_2 p + c$, and that c is slightly less than 3. Can this be proved?
2) Is there a polynomial algorithm for finding a balanced set of minimum size plus a constant, where the constant is independent of the input prime p?
3) Is there a polynomial algorithm for finding a balanced set of minimum size for any p, thus computing $\alpha(p)$? Conversely, is this problem provably NP-complete?
4) Can we find $\alpha(p)$ without constructing a minimum size balanced set?
5) By exhaustive search, we found minimum size balanced sets for small primes. The results consistently showed that $\alpha(p)$ is always either $\lceil \log_2 p \rceil + 2$ or $\lceil \log_2 p \rceil + 1$. Is it true that for each prime p there exists a balanced set S with size at most $\lceil \log_2 p \rceil + 2$?
6) Is a minimal (a smallest) balanced set S with m elements unique up to translation and scaling ($aS + b \bmod p$, with $a, b \in \mathbb{F}_p, a \neq 0$)?

Acknowledgments

We are grateful to the following people and institutions for their advice, discussion, and support during the development of this paper: Jeffrey O. Shallit, Noam Sturmwind for editing assistance, Valerie King, Uriel Feige, S. Muthu Muthukrishnan, Mario Szegedy, School of Computer Science at the University of Waterloo, and the DIMACS Center.

References

[1] Browkin, J., Diviš, B., and Schinzel, A., *Addition of sequences in general fields*, Monatshefte für Mathematik **82**, pp. 261-268, 1976. MR0432581 (55:5568)
[2] Johnson, Charles R. and Newman, Morris, *A surprising determinantal inequality for real matrices*, Math. Ann. **247**, pp. 179-186, 1980. MR568207 (83h:15005)
[3] Nedev, Zhivko, *Lower bound for balanced sets*, preprint.
[4] Nedev, Zhivko, *Universal sets and the vector game*, INTEGERS: The Electronic Journal of Combinatorial Number Theory, 8 (2008), #A45.
[5] Nedev, Zhivko and and Muthukrishnan, S., *The Magnus-Derek Game*, Theoretical Computer Science, Volume 393, Issues 1-3, 20 March 2008, pp. 124-132. MR2397246
[6] Nedev, Zhivko and Quas, Anthony, *Balanced sets and the vector game*, International Journal of Number Theory, 4 (2008), pp. 339-347. MR2424326
[7] Straus, E. G., *Differences of residues* (mod p), Journal of Number Theory **8**, pp. 40-42, 1976. MR0392876 (52:13689)

Department of Mathematics and Statistics, University of Victoria, P.O. Box 3060, STN CSC, Victoria, B.C., Canada V8W 3R4
E-mail address: znedev@gmail.com