On some batch code properties of the simplex code

Henk D.L. Hollmann[1], Karan Khathuria[1], Ago-Erik Riet[2], and Vitaly Skachek[1]

[1] Institute of Computer Science, University of Tartu, Tartu 50409, Estonia
{henk.d.l.hollmann,karan.khathuria,vitaly.skachek}@ut.ee
[2] Institute of Mathematics and Statistics, University of Tartu, Tartu 50409, Estonia
ago-erik.riet@ut.ee

Abstract. The binary k-dimensional simplex code is known to be a 2^{k-1}-batch code and is conjectured to be a 2^{k-1}-functional batch code. Here, we offer a simple, constructive proof of a result that is "in between" these two properties. Our approach is to relate the required property to an additive problem in finite abelian groups.

Keywords: Batch codes · Simplex code · Switch codes · Finite abelian groups

1 Introduction

A *t-batch code* is a method to store a data record in encoded form on multiple servers in such a way that the bit-values in any batch of t positions from the record can be retrieved by decoding the bit-values in t disjoint groups of positions.

Batch codes were initially introduced in [2] as a method to improve load-balancing in distributed data storage systems. Later, so-called *switch codes* (a special case of batch codes) were proposed in [6] as a method to increase the throughput rate in network switches.

In [4, 5], it was shown that the well-known binary simplex code, a code of length 2^k-1, dimension k, and minimum distance 2^{k-1} ($k \geq 1$ integer) is a 2^{k-1}-batch code. The proof of that result is somewhat cumbersome, and the algorithm resulting from the proof requires to store and use a database containing all the solutions for the cases where $k \leq 7$. More recently, in [9, 10] the authors conjecture that the k-dimensional simplex code is even a 2^{k-1}-functional batch code. (For precise definitions of this and other used notions, we refer to Section 2.)

In this paper, we give a simple, algorithmic proof of a result that falls halfway between the known result for the simplex code in [4, 5] and the conjecture in [9, 10]. Our approach is to relate the required properties of the simplex code to an additive problem in finite abelian groups.

The contents of this paper are as follows. In Section 2, we provide precise definitions of all the notions mentioned above, together with precise statements of some known results and conjectures. In our approach, we deal with certain reformulations of these statements, as derived in Section 3. In Section 4 we describe a variation of an algorithm in abelian groups first discovered by Marshall Hall, Jr., with slightly simpler proofs than those given in [1], that we then use to demonstrate our main result. We end with some conclusions in Section 5.

2 Preliminaries

All codes in this paper are binary and linear. We use $\mathbb{F}_2$ to denote the finite field of two elements 0,1, with addition and multiplication modulo 2, and we write $\mathbb{F}_2^k$ for the vector space of dimension k over $\mathbb{F}_2$. Thus $\mathbb{F}_2^k$ consists of all binary vectors of length k. Vectors and matrices will be denoted by boldface italic symbols. The ith unit vector $\boldsymbol{e}_i$ is the binary vector that has a one in position i and a zero in all other positions. We will write E_k to denote the set of the k unit vectors of length k. For convenience, we commonly number the positions with the integers $0, 1, \ldots, k-1$. The *support* of a vector $\boldsymbol{v} \in \mathbb{F}_2^k$, written as $\mathrm{supp}(\boldsymbol{v})$, is the collection of positions where $\boldsymbol{v}$ has a 1, and the (Hamming) *weight* $w(\boldsymbol{v})$ of $\boldsymbol{v}$ is the size of $\mathrm{supp}(\boldsymbol{v})$.

Definition 1 We say that a binary $k \times n$ matrix $\boldsymbol{G}$ can *serve* a *request sequence* $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t$ of (not necessarily distinct) nonzero vectors in $\mathbb{F}_2^k$ if we can find pairwise disjoint subsets $I_1, \ldots, I_t$ of the columns of $\boldsymbol{G}$ such that for $j = 1, \ldots, t$, the columns in I_j sum to $\boldsymbol{r}_j$.

We will be interested in various properties of such matrices defined in terms of the particular request sequences that they can serve.

Definition 2 The binary $k \times n$ matrix $\boldsymbol{G}$ (as well as the binary linear code generated by $\boldsymbol{G}$) is (i) a *t-PIR code*, (ii) a *t-batch code*, (iii) a *t-odd batch code*, or (iv) a *t-functional batch code* if $\boldsymbol{G}$ can serve any request sequence of length t consisting of (i) the t-fold repetition of a unit vector in E_k, (ii) unit vectors in E_k only, (iii) vectors in $\mathbb{F}_2^k$ of odd weight only, or (iv) nonzero vectors in $\mathbb{F}_2^k$, respectively.

The notions of t-PIR code and t-batch code are well known (but note that some authors employ a more general definition and refer to these codes as *multiset primitive*), and together with t-functional batch codes are defined, for example, in [9, 10]. For a recent overview of these and related types of codes, see [3]. The notion of t-odd batch code is new and is introduced here for convenience.

The (binary) *simplex code* of length $n = 2^k - 1$ has a $k \times (2^k - 1)$ generator matrix $\boldsymbol{G}_k$ whose columns are the distinct nonzero vectors in $\mathbb{F}_2^k$. In [4] (see also [5]) it was shown that $\boldsymbol{G}_k$ is a 2^{k-1}-batch code, but the proof is somewhat cumbersome. Recently [9, 10], it was conjectured that $\boldsymbol{G}_k$ is even a 2^{k-1}-functional batch code, and it was shown that $\boldsymbol{G}_k$ is a t-functional batch code for $t = 2^{k-2} + 2^{k-4} + \lfloor 2^{k/2}/\sqrt{24} \rfloor$, again with a rather involved proof. After completion of this paper, we learned that this result was further improved in [7] and [8], where it was shown that $\boldsymbol{G}_k$ is a t-functional batch code for $t = \lfloor (2/3) \cdot 2^{k-1} \rfloor$ and $t = \lfloor (5/6) \cdot 2^{k-1} \rfloor - k$, respectively.

In this paper, we will provide a simple algorithmic proof that $\boldsymbol{G}_k$ is a 2^{k-1}-odd batch code. In fact, we will prove slightly more.

Theorem 3 For every integer $k \geq 1$, the binary simplex code of length $n = 2^k - 1$ and dimension k, with generator matrix $\boldsymbol{G}_k$ as above, is a 2^{k-1}-odd batch code. In addition, every sequence of 2^{k-1} odd-weight vectors from $\mathbb{F}_2^k$ can be served with column subsets of size at most two.

Note that since every unit vector has odd weight (in fact, a weight equal to 1), Theorem 3 implies that the k-dimensional simplex code generated by $\boldsymbol{G}_k$ is in fact a 2^{k-1}-batch code, a fact that was first proved in [4, 5].

In [9, 10], it was conjectured that the simplex code of length $2^k - 1$ is in fact a 2^{k-1}-functional batch code. We believe that even a slightly stronger result may be true.

Conjecture 4 For every integer $k \geq 1$, the binary simplex code of length $n = 2^k - 1$ and dimension k, with generator matrix $\boldsymbol{G}_k$ as above, can serve every sequence of 2^{k-1} vectors from $\mathbb{F}_2^k$ *with column subsets of size at most two*.

Now a request $\boldsymbol{r} \neq \boldsymbol{0}$ is served by a set of columns $I = \{\boldsymbol{x}, \boldsymbol{y}\}$ of $\boldsymbol{G}_k$, where we may assume $\boldsymbol{x} \neq \boldsymbol{0}$, and possibly $\boldsymbol{y} = \boldsymbol{0}$, precisely when there are vectors $\boldsymbol{x}, \boldsymbol{y}$ with $\boldsymbol{x} \neq \boldsymbol{0}$ such that $\boldsymbol{x} + \boldsymbol{r} = \boldsymbol{y}$. So it is easily seen that the above conjecture is in fact equivalent to the following.

Conjecture 5 Let $k \geq 1$ be an integer. For every sequence of nonzero vectors $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^{k-1}}$ in $\mathbb{F}_2^k$, there are pairwise distinct nonzero vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^{k-1}}$ in $\mathbb{F}_2^k$ such that the nonzero vectors among $\boldsymbol{y}_1 = \boldsymbol{x}_1 + \boldsymbol{r}_1, \ldots, \boldsymbol{y}_{2^{k-1}} = \boldsymbol{x}_{2^{k-1}} + \boldsymbol{r}_{2^{k-1}}$ are pairwise distinct and distinct from $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^{k-1}}$.

3 A reformulation

We will in fact prove the following slight generalization of Theorem 3.

Theorem 6 Let $k \geq 1$ be an integer, and let H be a $(k-1)$-dimensional subspace of $\mathbb{F}_2^k$. The binary simplex code of length $n = 2^k - 1$ and dimension k, with generator matrix $\boldsymbol{G}_k$ as above, can serve every sequence of 2^{k-1} vectors from the complement $\mathbb{F}_2^k \setminus H$ of H with column subsets of size at most two.

In fact, as one of the referees remarked, Theorem 3 and Theorem 6 can easily seen to be equivalent, by applying a suitable invertible linear transformation to the request sequence. Below, the equivalence will be obtained in another way.

For later use, we now derive several equivalent formulations of Theorem 6. Note that if $I = \{\boldsymbol{x}, \boldsymbol{y}\}$ with $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{r}$ and $\boldsymbol{r} \in \mathbb{F}_2^k \setminus H$, then without loss of generality we may assume that $\boldsymbol{x} \in \mathbb{F}_2^k \setminus H$ (hence nonzero) and $\boldsymbol{y} \in H$. So it is easily seen that Theorem 6 is in fact equivalent to the following.

Theorem 7 Let $k \geq 1$ be an integer, and let H be a $(k-1)$-dimensional subspace of $\mathbb{F}_2^k$. For every sequence of vectors $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^{k-1}}$ in $\mathbb{F}_2^k \setminus H$, there are pairwise distinct vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^{k-1}}$ in $\mathbb{F}_2^k \setminus H$ such that the nonzero vectors among the vectors $\boldsymbol{x}_1 + \boldsymbol{r}_1, \ldots, \boldsymbol{x}_{2^{k-1}} + \boldsymbol{r}_{2^{k-1}}$ in H are also pairwise distinct.

Of special interest is the case where H and $\mathbb{F}_2^k \setminus H$ are the collection of even and odd weight vectors in $\mathbb{F}_2^k$, respectively. Note that for this case, Theorem 6 reduces to Theorem 3.

We will now show the equivalence of Theorem 7 and the following.

Theorem 8 Let $k \geq 1$ be an integer. Given any sequence $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^k}$ in $\mathbb{F}_2^k$, there exists a numbering $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^k}$ of the vectors in $\mathbb{F}_2^k$ such that the nonzero vectors in the sequence $\boldsymbol{x}_1 + \boldsymbol{r}_1, \ldots, \boldsymbol{x}_{2^k} + \boldsymbol{r}_{2^k}$ are pairwise distinct.

To show this equivalence, we need some preparation. Let H be a $(k-1)$-dimensional subspace of $\mathbb{F}_2^k$. Then H is of the form

$$H = \boldsymbol{u}^{\perp} := \{\boldsymbol{h} \in \mathbb{F}_2^k \mid \boldsymbol{h}_1\boldsymbol{u}_1 + \cdots + \boldsymbol{h}_k\boldsymbol{u}_k = \boldsymbol{0}\}$$

for some $\boldsymbol{u} = (u_1, \ldots, u_k) \in \mathbb{F}_2^k \setminus \{\boldsymbol{0}\}$. Now H is a subgroup of $(\mathbb{F}_2^k, +)$ isomorphic to $(\mathbb{F}_2^{k-1}, +)$. Indeed, if $u_j \neq 0$, then it is easily verified that $\varphi : H \to \mathbb{F}_2^{k-1}$ defined by $\varphi(\boldsymbol{h}) = (h_1, \ldots, h_{j-1}, h_{j+1}, \ldots, h_k)$ is an isomorphism. Fix some $\boldsymbol{a} \in \mathbb{F}_2^k \setminus H$, and extend φ to a linear map $\varphi : \mathbb{F}_2^k \to \mathbb{F}_2^{k-1}$ by defining $\varphi(\boldsymbol{a}+\boldsymbol{h}) = \varphi(\boldsymbol{h})$ for $\boldsymbol{h} \in H$. Note that φ also sets up a one-to-one correspondence between $\mathbb{F}_2^k \setminus H$ and $\mathbb{F}_2^{k-1}$.

Now let $k \geq 2$ be an integer. First, suppose Theorem 8 holds for $k-1$, let H be a $(k-1)$-dimensional subspace of $\mathbb{F}_2^k$, and let $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^{k-1}}$ be a sequence in $\mathbb{F}_2^k \setminus H$. Let $\varphi : \mathbb{F}_2^k \to \mathbb{F}_2^{k-1}$ be a linear map, 1-1 on both H and $\mathbb{F}_2^k \setminus H$, constructed as discussed above. Put $\boldsymbol{r}_i' = \varphi(\boldsymbol{r}_i)$ $(i = 1, \ldots, 2^{k-1})$. Applying Theorem 8 for $k-1$, we conclude that there is a numbering $\boldsymbol{x}_1', \ldots, \boldsymbol{x}_{2^{k-1}}'$ of the vectors in $\mathbb{F}_2^{k-1}$ such that the nonzero vectors among $\boldsymbol{y}_1' = \boldsymbol{x}_1' + \boldsymbol{r}_1', \ldots, \boldsymbol{y}_{2^{k-1}}' = \boldsymbol{x}_{2^{k-1}}' + \boldsymbol{r}_{2^{k-1}}'$ are pairwise distinct. Let $\boldsymbol{x}_i \in \mathbb{F}_2^k \setminus H$, $\boldsymbol{r}_i \in \mathbb{F}_2^k \setminus H$, and $\bar{y}_i \in H$ $(i = 1, \ldots, 2^{k-1})$ be the unique vectors such that $\varphi(\boldsymbol{x}_i) = \boldsymbol{x}_i'$, $\varphi(\boldsymbol{r}_i) = \boldsymbol{r}_i'$, and $\varphi(\boldsymbol{y}_i) = \boldsymbol{y}_i'$, respectively. By the linearity of φ, we have that $\boldsymbol{x}_i + \boldsymbol{r}_i = \boldsymbol{y}_i$ for all i; moreover, since φ is one-to-one on H and on $\mathbb{F}_2^k \setminus H$, with $\boldsymbol{y}_i = \boldsymbol{0}$ if and only if $\varphi(\boldsymbol{y}_i) = \boldsymbol{y}_i' = \boldsymbol{0}$, the $\boldsymbol{x}_i$ are pairwise distinct and the nonzero $\boldsymbol{y}_i$ are also pairwise distinct. So we conclude that Theorem 7 holds for k.

Conversely, suppose that the statement in Theorem 7 holds for *some* hyperplane H in $\mathbb{F}_2^k$ (for example, with H the collection of even-weight vectors in $\mathbb{F}_2^k$), and let $\boldsymbol{r}_1', \ldots, \boldsymbol{r}_{2^{k-1}}'$ be in $\mathbb{F}_2^{k-1}$. Let $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^{k-1}}$ be the unique vectors in $\mathbb{F}_2^k \setminus H$ for which $\varphi(\boldsymbol{r}_i) = \boldsymbol{r}_i'$ for all i, where φ is as defined in the beginning of the proof. We conclude that there are pairwise distinct vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^{k-1}}$ in $\mathbb{F}_2^k \setminus H$ such that the nonzero vectors among $\boldsymbol{y}_1 = \boldsymbol{x}_1 + \boldsymbol{r}_1, \ldots, \boldsymbol{y}_{2^{k-1}} = \boldsymbol{x}_{2^{k-1}} + \boldsymbol{r}_{2^{k-1}}$ are also pairwise distinct. Now let $\boldsymbol{x}_j' = \varphi(\boldsymbol{x}_j)$ and $\boldsymbol{y}_j' = \varphi(\boldsymbol{y}_j)$ for all j. By linearity of φ, we have $\boldsymbol{x}_j' + \boldsymbol{r}_j' = \boldsymbol{y}_j'$ for all j. Moreover, since φ is one-to-one both on H and on $\mathbb{F}_2^k \setminus H$, the $\boldsymbol{x}_j'$ are pairwise distinct, hence they form a numbering of the vectors in $\mathbb{F}_2^{k-1}$, and the nonzero $\boldsymbol{y}_j'$ are also pairwise distinct. So we conclude that Theorem 8 holds for $k-1$. We have proved the following.

Theorem 9 For every integer $k \geq 2$, Theorem 6, Theorem 7, and Theorem 3 are all equivalent to Theorem 8 (with k replaced by $k-1$).

4 A servicing algorithm

We will now describe an algorithm to solve the numbering problem inherent in Theorem 8, in the more general context of finite abelian groups. We first introduce some terminology.

Definition 10 Let $(G, +)$ be a finite abelian group. A *service* for a given sequence $r_1, \ldots, r_m$ in G is a collection of pairwise distinct $x_1, \ldots, x_m \in G$ such that the m elements $y_1 = x_1 + r_1, \ldots, y_m = x_m + r_m$ are also pairwise distinct in G.

We will often think of such a service as a collection of ordered triples

$$(x_1, y_1, r_1), \ldots, (x_m, y_m, r_m)$$

with $x_1, \ldots, x_m$ pairwise distinct in G, $y_1, \ldots, y_m$ pairwise distinct in G, and $x_i + r_i = y_i$ for $i = 1, \ldots, m$. The next result is crucial for our approach.

Theorem 11 Let $(G, +)$ be a finite abelian group. Given a service $x_1, \ldots, x_m$ for the sequence $r_1, \ldots, r_m$ in G with $0 \leq m \leq |G| - 2$, and some additional element $r_0 \in G$, we can find some $x \in G$ distinct from $x_1, \ldots, x_m$ such that some permutation of $x, x_1, \ldots, x_m$ is a service for $r_0, r_1, \ldots, r_m$.

Proof. We give a constructive proof of this theorem by describing an algorithm that extends a given service consisting of the triples $(x_1, y_1, r_1), \ldots, (x_m, y_m, r_m)$ in G of length at most $|G| - 2$ as in the theorem, so with $x_j + r_j = y_j$ for $j = 1, \ldots, m$. To this end, let y_{-1}, y_0 be two distinct elements outside $\{y_1, \ldots, y_m\}$ (which is possible by our assumption that $m \leq |G| - 2$), define $x_0 = y_0 - r_0$, and set $c = x_0 + y_{-1}$. (Note that we may assume that $x_0 \in \{x_1, \ldots, x_m\}$ since otherwise we could extend the service with the new triple (x_0, y_0, r_0); however, we will not use this information below.)

Assume that after t steps of our algorithm ($t = 0, 1, \ldots$), we have found t triples $(x_1, y_1, r_1), \ldots, (x_t, y_t, r_t)$ (after renumbering triples if necessary) from the given service such that the relations $x_j + r_{j-1} = y_{j-2}$ hold for $j = 1, \ldots, t$, and in addition,

$$x_j + y_{j-1} = c \tag{1}$$

holds for $j = 0, \ldots, t$, see Figure 1.

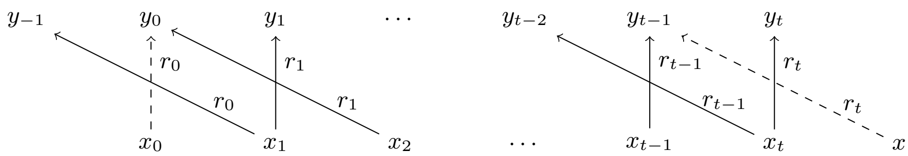

Fig. 1. The extension algorithm

Note that the case $t = 0$ describes the initial situation where we have no triples yet, no relations, and where $x_0 + y_{-1} = c$ by definition of c.

Then in step $t + 1$, to extend the list of triples from the given service, we proceed as follows. Define $x = y_{t-1} - r_t$. Note that from the "crossing edges", we obtain the relation $r_t = y_t - x_t = y_{t-1} - x$, hence

$$x + y_t = x_t + y_{t-1} = c \tag{2}$$

by our assumptions. Depending on where x is situated, we now distinguish several cases.

Case 1: $x \notin \{x_1, \ldots, x_m\}$. Then the t new triples $(x_1, y_{-1}, r_0), \ldots, (x_t, y_{t-2}, r_{t-1})$, the triple (x, y_{t-1}, r_t), and the $m-t$ old triples $(x_{t+1}, y_{t+1}, r_{t+1}), \ldots, (x_m, y_m, r_m)$ together constitute a service for $r_0, r_1, \ldots, r_m$, and we are done.

Case 2: $x \in \{x_{t+1}, \ldots, x_m\}$. Then after renumbering triples if necessary, we may assume that $x = x_{t+1}$, where $x_{t+1} + r_{t+1} = y_{t+1}$. By (2), we have $x_{t+1} + y_t = c$, and we have extended the configuration in Figure 1 from t to $t + 1$ triples.

Case 3: $x \in \{x_1, \ldots, x_t\}$. We will show that this case cannot occur. Indeed, suppose that $x = x_j$ with $1 \le j \le t$. From (1) and (2), we have that $x_j + y_{j-1} = c = x_j + y_t$, hence $y_t = y_{j-1}$ for some $j = 1, \ldots, t$, contradicting our assumptions.

Since in any configuration, $t \le m$ must hold, the algorithm will eventually end in case 1, and thus will produce an extended service.

Theorem 11 has an interesting consequence.

Theorem 12 Let $(G, +)$ be a finite abelian group of size n and let $r_1, \ldots, r_n$ be a sequence in G. Then there is a numbering $x_1, \ldots, x_n$ of the elements of G such that $x_1 + r_1, \ldots, x_n + r_n$ form a permutation of the elements of G if and only if $r_1 + \cdots + r_n = 0$.

Proof. If (x_i, r_i, y_i) $(i = 1, \ldots, n)$ is a service for r, then the x_i and the y_i are both a permutation of G, so if $x_i + r_i = y_i$ for all i, then $\sum r_i = \sum y_i - \sum x_i = 0$. So the condition on the r_i is necessary. Conversely, suppose that $\sum r_i = 0$. By Theorem 11, we can construct a service $(x_1, r_1, y_1), \ldots, (x_{n-1}, r_{n-1}, y_{n-1})$ for $r_1, \ldots, r_{n-1}$. Let g denote the sum of all the elements of G. If x_n and y_n are the elements in G that do not occur among x_i and y_i $(i = 1, \ldots, n-1)$, respectively, then $x_n = g - \sum_{i \neq n} x_i$, $y_n = g - \sum_{i \neq n} y_i$ and $r_n = -\sum_{i \neq n} r_i$, and since $x_i + r_i = y_i$ for $i = 1, \ldots, n-1$, we conclude that in addition $x_n + r_n = y_n$.

This theorem (with a slightly different proof, but employing essentially the same algorithm) was first stated in [1]. We can also employ Theorem 11 to prove Theorem 8.

Proof of Theorem 8: Given a sequence $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^k}$ in $\mathbb{F}_2^k$, we can use Theorem 11 to construct pairwise distinct vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^k-1} \in \mathbb{F}_2^k$ such that the vectors $\boldsymbol{y}_1 = \boldsymbol{x}_1 + \boldsymbol{r}_1, \ldots, \boldsymbol{y}_{2^k-1} = \boldsymbol{x}_{2^k-1} + \boldsymbol{r}_{2^k-1}$ are also pairwise distinct. Let $\boldsymbol{x}_{2^k}$ denote the vector in $\mathbb{F}_2^k$ such that $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2^k}\} = \mathbb{F}_2^k$. For every vector $\boldsymbol{a} \in \mathbb{F}_2^k$, the triples $(\boldsymbol{x}_i + \boldsymbol{a}, \boldsymbol{y}_i + \boldsymbol{a}, \boldsymbol{r}_i)$ for $i = 1, \ldots, 2^k - 1$ again form a service for $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^k-1}$, so by choosing $\boldsymbol{a} = \boldsymbol{x}_{2^k} + \boldsymbol{r}_{2^k}$ and replacing $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ by $\boldsymbol{x}_i + \boldsymbol{a}$

and $\boldsymbol{y}_i + \boldsymbol{a}$, we may assume without loss of generality that $\boldsymbol{x}_{2^k} = \boldsymbol{r}_{2^k}$. Now define $\boldsymbol{y}_{2^k} = \mathbf{0}$ and add the triple $(\boldsymbol{r}_{2^k}, \mathbf{0}, \boldsymbol{r}_{2^k})$ as the last triple to complete the service to one for $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2^k}$.

We have now proved Theorem 8; in view of Theorem 9, this implies Theorem 7, Theorem 6, and our main result Theorem 3.

5 Conclusions

We related various known and conjectured batch-type properties of the simplex codes to certain additive problems in finite abelian groups. By applying known methods for these more general problems we obtained a simple, constructive proof of a generalization of the theorem that the k-dimensional binary simplex code is a 2^{k-1}-batch code.

Acknowledgments

We thank Lev Yohananov and Eitan Yaakobi for sharing their unpublished work [7] with us. This research was supported by the Estonian Research Council grants PRG49 and PSG114, and by the European Regional Development Fund via CoE project EXCITE.

References

1. M. Hall, Jr. A combinatorial problem on abelian groups. *Proc. Amer. Math. Soc.*, 3(4):584–587, 1952.
2. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Batch codes and their applications. In *Proc. 36th ACM Symposium on Theory of Computing (STOC), Chicago, IL*, pages 1057–1061, June 2004.
3. V. Skachek. Batch and PIR codes and their connections to locally repairable codes. In M. Greferath, M. O. Pavčević, N. Silberstein, and M. Ángeles Vázquez-Castro, editors, *Network Coding and Subspace Designs*, pages 427–442. Springer, 2018.
4. Z. Wang, H. M. Kiah, and Y. Cassuto. Optimal binary switch codes with small query size. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 636–640, 2015.
5. Z. Wang, H. M. Kiah, Y. Cassuto, and J. Bruck. Switch codes: Codes for fully parallel reconstruction. *IEEE Trans. on Inform. Theory*, 63(4):2061–2070, 2017.
6. Z. Wang, O. Shaked, Y. Cassuto, and J. Bruck. Codes for network switches. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 1057–1061, 2013.
7. L. Yohananov and E. Yaakobi. Almost optimal construction of functional batch codes using Hadamard codes. Submitted to *IEEE Trans. on Inform. Theory*, 2021.
8. L. Yohananov and E. Yaakobi. Almost optimal construction of functional batch codes using Hadamard codes. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 3139–3144, 2021.
9. Y. Zhang, T. Etzion, and E. Yaakobi. Bounds on the length of functional PIR and batch codes. *IEEE Trans. on Inform. Theory*, pages 4917–4934, 2020.

10. Y. Zhang, E. Yaakobi, and T. Etzion. Bounds on the length of functional PIR and batch codes. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 2129–2133, 2019.