Discrete Mathematics 312 (2012) 621–632

Contents lists available at SciVerse ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

Switching codes and designs

Patric R.J. Östergård *

Department of Communications and Networking, Aalto University School of Electrical Engineering, P.O. Box 13000, 00076 Aalto, Finland
Lehrstuhl Mathematik II, Universität Bayreuth, 95440 Bayreuth, Germany

ARTICLE INFO

Article history:
Available online 12 June 2011

Keywords:
Combinatorial design
Covering code
Error-correcting code
Switching
Trade

ABSTRACT

Various local transformations of combinatorial structures (codes, designs, and related structures) that leave the basic parameters unaltered are here unified under the principle of switching. The purpose of the study is threefold: presentation of the switching principle, unification of earlier results (including a new result for covering codes), and applying switching exhaustively to some common structures with small parameters.

1. Introduction

In combinatorics, as a branch of discrete mathematics concerning the study of finite or countable discrete structures, the question whether two given structures are essentially the same is a central issue. Indeed, the concept of *isomorphism* (or *equivalence*, *isotopy*, etc. depending on the type of structure) has been introduced in this context. Whereas this concept is well understood as such, there is still much work to be done for the related problems of existence, enumeration, and classification of combinatorial structures.

Isomorphism is essentially about transforming structures via relabeling. For many families of combinatorial structures, the number of isomorphism classes is large (for example, it may show exponential growth with respect to some parameter), so it is natural to think about other possible transformations that would merge isomorphism classes. It is further desirable that such transformations be local and precisely defined, to make any practical (analytical or computational) study feasible.

Through transformations of the sought kind one may indeed gain further understanding of issues like a plethora of isomorphism classes in certain cases. These transformations may also be applied to any incomplete classification to see whether additional isomorphism classes could be found (perhaps in the search for structures with some particular property). Further applications will be mentioned at the end of the paper, in Sections 3 and 4.

One aim of this work is to give a precise definition for a local transformation with variants for codes, designs, and related combinatorial structures. Through the definition of this transformation, termed switching, a variety of earlier methods and results for different structures can be unified. The earlier work on this topic can be traced back at least to work by Norton [71] and Fisher [28] in the 1930s.

Graphs are not in general considered in the current work; graph switching has been discussed and applied in a variety of places, [5,14,30,34,39,47,86,88,94,97,101] just to mention a few, and this topic desires a treatment of its own. Also some similar techniques in finite geometry, such as those considered in [7,65], fall beyond the scope of this work.

The paper is organized as follows. Switching of various types of structures is considered in Section 2, starting from error-correcting codes (Section 2.1), and proceeding via constant weight codes (Section 2.2), designs (Section 2.3), equidistant

* Corresponding address: Department of Communications and Networking, Aalto University School of Electrical Engineering, P.O. Box 13000, 00076 Aalto, Finland. Tel.: +358 9 470 22341; fax: +358 9 470 22474.
E-mail address: patric.ostergard@tkk.fi.

0012-365X/$ – see front matter
doi:10.1016/j.disc.2011.05.016

codes (Section 2.4), and group divisible designs, in particular, Latin squares (Section 2.5) to covering codes (Section 2.6). Possibilities of using switching when searching for combinatorial structures are briefly considered in Section 3. The paper is concluded in Section 4.

2. Switching

We here define *switching* and *switches* as certain local transformations that do not alter the basic parameters of a combinatorial structure. The locality aspect of such a transformation should be emphasized; allowing an arbitrary transformation, any structure with given parameters can be obtained from any other, which is not useful in practice.

Throughout the paper we shall consider switching as a transformation between isomorphism (or equivalence) classes of objects. Some studies, including [38,50,85], have also touched issues related to switching as a means for transforming a labeled structure into another.

The first structures to be considered here are binary error-correcting codes, for which there is a natural definition of a local transformation. An obvious modification that is necessary for such codes with constant weight will later bring us to designs via Steiner systems (which are optimal constant weight error-correcting codes).

2.1. Error-correcting codes

Unless otherwise mentioned, the binary case will be assumed in the sequel. A *word* is a vector $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{Z}_2^n$, and a *code* is a nonempty set of (code) words $C \subseteq \mathbb{Z}_2^n$. The (Hamming) *distance* between two words, $\mathbf{x}$ and $\mathbf{y}$, is the number of coordinates in which they differ and is denoted by $d_H(\mathbf{x}, \mathbf{y})$. The *minimum distance* of the code C is

$$d(C) := \min\{d_H(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in C,\ \mathbf{x} \neq \mathbf{y}\}.$$

A code $C \subseteq \mathbb{Z}_2^n$ with M codewords and minimum distance d is called an (n, M, d) *error-correcting code*. A classic reference for error-correcting codes is [62].

One of the major problems in coding theory is that of determining the largest possible value of M for which an (n, M, d) code exists, given n and d; this value is denoted by $A(n, d)$ and the corresponding codes are said to be *optimal*. A code with odd minimum distance can be extended by adding a parity check bit to increase the minimum distance by 1; in this way it is easy to see that

$$A(n, 2d - 1) = A(n + 1, 2d). \tag{1}$$

Two codes are *equivalent* if one of the codes can be mapped onto the other by a permutation of the coordinates and permutations of the coordinate values (the permutations of the coordinate values can in the binary case be viewed via addition of a vector to all codewords). All such mappings from a code C onto itself form the *automorphism group* of the code, $\mathrm{Aut}(C)$.

We shall now introduce the concept of switching for error-correcting codes. Locality of transformations is ensured by restricting the number of coordinates affected. In fact, we require the number to be as small as possible, that is, one.

Definition 1. A *switch* of a binary code is a transformation that concerns exactly one coordinate and keeps the studied parameter of the code (for example, minimum distance) unchanged.

Note that Definition 1 indeed applies to other types of codes than error-correcting codes.

For error-correcting codes, all possible switches of a given code can be calculated efficiently. First of all, there are n (the length of the code) possible choices for the particularized coordinate s. Now, if some codeword $\mathbf{c}$ is altered, by changing c_s to $1 - c_s$, then for any codeword $\mathbf{c}'$ such that

$$d_H(\mathbf{c}, \mathbf{c}') = d, \qquad c_s \neq c'_s \tag{2}$$

we must also change c'_s to $1 - c'_s$. Any changes that are carried out propagate in this direct way. Let us now have a formal look at this situation in the context of graphs.

Given a code C, a particularized coordinate s, and the minimum distance $d(C)$, we construct a bipartite graph $G = (V, E)$ with $V = V_0 \cup V_1$ and one vertex in V_j for each codeword $\mathbf{c} \in C$ with $c_s = j$. Moreover, there is a edge between two vertices in V exactly when the corresponding codewords fulfill (2).

The vertex set V is now partitioned by the connected components of G. Any switch with respect to the coordinate s corresponds to a union of such connected components. A switch that corresponds to exactly one connected component is called *minimal* or *irreducible*. As any switch for a given coordinate can be obtained via a sequence of minimal switches, a computational study may be restricted to minimal switches.

Extracting the connected components is one of the most basic computational problems for graphs; this problem can be solved in linear time with respect to $|V| + |E|$ by breadth-first search or depth-first search [19]. Consequently, the time needed is at most quadratic with respect to the number of vertices.

The graph G might have only one connected component, whereby a switch just complements a coordinate and, by definition, produces an equivalent code. Such a switch is called *trivial*. However, under a certain common circumstance

Table 1
Switching classes of error-correcting codes.

n	d	$A(n, d)$	N	Sizes of switching classes
6	3	8	1	1
7	3	16	1	1
8	3	20	5	3, 2
9	3	40	1	1
10	3	72	562	165, 134, 110, 89, 26, 15, 14, 9
11	3	144	7 398	7013, 385
12	3	256	237 610	234 749, 2509, 331, 3, 3, 3, 3, 3, 3
13	3	512	117 823	115 973, 1240, 561, 6, 6, 4, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 1, 1, 1
15	3	2048	5 983	5819, 153, 3, 2, 2, 1, 1, 1, 1
9	5	6	1	1
10	5	12	1	1
11	5	24	1	1
12	5	32	2	2
13	5	64	1	1
14	5	128	1	1
15	5	256	1	1
12	7	4	9	3, 2, 2, 1, 1
13	7	8	6	2, 2, 1, 1
14	7	16	10	3, 2, 2, 2, 1
15	7	32	5	1, 1, 1, 1, 1
15	9	4	9	3, 2, 2, 1, 1

there will inevitable be nontrivial switches, see [79,91]: If the minimum distance d is odd, then two codewords, $\mathbf{c}$ and $\mathbf{c}'$, with $c_s = c'_s$ and $d_H(\mathbf{c}, \mathbf{c}')$ odd or $c_s \neq c'_s$ and $d_H(\mathbf{c}, \mathbf{c}')$ even cannot belong to the same connected component. Alternatively, the condition can be phrased in terms of the distance between $\mathbf{c}$ and $\mathbf{c}'$ for the subset of coordinates excluding the particularized one: a necessary condition for the codewords to belong to the same connected component is that this distance is even.

Also nontrivial switches might obviously lead to codes that are equivalent to the original code.

The *switching graph* for given code parameters consists of one vertex for each equivalence class of codes and with edges between two vertices whenever one can get from one equivalence class to the other via a switch. We may consider undirected graphs as switches are reversible. The (equivalence classes corresponding to the vertices of the) connected components of the switching graph are called *switching classes*. To emphasize that such switching classes consist of equivalence classes of codes, the longer term *switching equivalence classes* can be used; however, this is generally not necessary unless switching of labeled structures are considered simultaneously. As we are only interested in the codes of the switching classes, it does not matter whether any switches or only minimal ones are considered for the switching graph.

Almost all studies in the literature on switching error-correcting codes have considered a particular type of optimal error-correcting codes known as perfect codes (in particular, for minimum distance 3). In fact, the roots of switching in coding theory can be traced back to the early 1960s and work by Vasil'ev [98], who presented a doubling construction of perfect codes, producing (many equivalence classes of) codes of length $2n + 1$ from codes of length n which can be explained in terms of switching.

In the 1980s, computational resources became available to researchers and the theory and practice for code switching started to emerge into its current form. Solov'eva [89] studied this issue in the context of disjunctive normal forms; other early results include [26]. In 1999, Phelps and LeVan [82] coined the term *switching* for these transformations. Some restricted [63,64] as well as more general [2] variants of switching have also been considered in the literature.

The sizes of switching classes of optimal binary error-correcting codes with odd minimum distance and length at most 15 are listed in Table 1, where N denotes the total number of equivalence classes. The switching classes of $(15, 2048, 3)$ (perfect) codes were determined in [80] from the codes classified and made available in [78] (it was previously known [82] that these have at least two switching classes). The switching classes of $(12, 256, 3)$ and $(13, 512, 3)$ codes were determined in [59]; see also [79], where two new codes were found in a calculation of switching classes. All other results in Table 1 were obtained in the current work, utilizing the electronic supplement to [53]. See also [53, Table 7.2].

There is a particular reason why only switching classes for optimal codes with *odd* minimum distance have been tabulated here. Namely, the codes obtained by switching an $(n, M, 2d)$ code can be viewed as different extensions of the $(n - 1, M, 2d - 1)$ code with the particularized coordinate removed. However, it is not a common property among short optimal error-correcting codes to have several inequivalent extensions, see [53, p. 230] and [75].

Consequently, it would make sense to define a switch of a code with even minimum distance via complementation of *two* coordinates rather than one (we shall see such switches later for other types of structures). However, as any coordinate of an optimal code with even minimum distance is in many cases (up to equivalence) uniquely determined by the rest of the code, such a switch is essentially a switch of a punctured code with odd minimum distance. So there is actually no need for a separate treatment of the even distance case. This observation is also related to the comment in [80] that the process of first extending and then puncturing a code can essentially be seen as a switch of the original code.

A code class that is not conserved by the switch in Definition 1 is that of constant weight error-correcting codes.

2.2. Constant weight codes

The (Hamming) *weight* of a word is the number of nonzero coordinates. If all codewords have the same weight, then the code is said to be a *constant weight code*. The largest size of a code with length n, minimum distance d, and constant weight w is denoted by $A(n, d, w)$. In the definition of equivalence of constant weight codes, only coordinate permutations are considered.

Definition 1 decreases or increases the weight of a codeword by 1. To be able to use switching in the sense of complementing coordinate values, one should therefore consider altering at least two coordinates.

Definition 2. A *switch* of a constant weight code is a transformation that concerns exactly two coordinates and keeps the studied parameter of the code (for example, minimum distance) unchanged.

The four possible combinations of values in two coordinates are 00, 01, 10, and 11. The combinations 00 and 11 must not be altered in switching, so only the combinations 01 and 10 are altered (that is, complemented) when switching constant weight codes. Note that such a complementation does not affect the distance from words with 01 or 10 to words with 00 or 11 in these coordinates.

Having set up the framework for switching constant weight codes in this way, the procedures and terminology related to switching can be directly carried over from Section 2.1, with obvious modifications. For example, now two distinct coordinates are particularized, s and t, and we start by constructing a bipartite graph $G = (V, E)$ with $V = V_{01} \cup V_{10}$ and a vertex in V_{ij} for each codeword $\mathbf{c} \in C$ with $c_s = i$ and $c_t = j$ and with an edge between two vertices exactly when the corresponding codewords fulfill

$$d_H(\mathbf{c}, \mathbf{c}') = d, \qquad (c_s, c_t) \neq (c'_s, c'_t). \tag{3}$$

The author is not aware of any earlier studies of switching classes of constant weight codes, except in cases where the codes correspond to combinatorial designs (which will be considered in Section 2.3). Constant weight codes have been classified in [74]. The switching classes for the codes tabulated in [74] were computed in the current work in all cases where $d \leq 10$ and the number of equivalence classes is greater than 1. The results are presented in Table 2.

The tabulation of the results in Table 2 is stopped at $d = 10$, for which most of the listed switching classes have size 1. One might be tended to draw a conclusion from this observation, but it should be emphasized that there is the following explanation for many sets of parameters with only switching classes of size 1: If a code has the property that the distance between any two codewords is d (such a code is called *equidistant*), then the bipartite graph with vertex set $V = V_{01} \cup V_{10}$ will be complete and thereby connected. Constant weight codes that come from a design in a way that will be discussed at the end of Section 2.3 are examples of equidistant codes; switching of such codes will be considered in Section 2.4.

We shall now proceed to the case of combinatorial designs, which are linked to constant weight codes in several ways.

2.3. Designs

A t-(v, k, λ) design is a pair $(X, \mathcal{B})$, where X is a finite set of v *points* and $\mathcal{B}$ is a multiset of k-subsets of X, called *blocks*, such that every t-subset of X occurs in exactly λ blocks. The number of points v is the *order* of the design. It is not difficult the determine the number of blocks, b, as well as the number of blocks in which a point occurs, r, in a t-(v, k, λ) design:

$$b = \lambda \binom{v}{t} \Big/ \binom{k}{t}, \qquad r = bk/v.$$

A design with $v = b$ (and therefore $k = r$) is said to be *symmetric*. The handbook [15] is an invaluable source for further information about designs and related structures.

Two designs are *isomorphic* if one of the designs can be mapped onto the other by a permutation of the points. All such mappings from a design $\mathcal{D}$ onto itself form the *automorphism group* of the design, $\mathrm{Aut}(\mathcal{D})$.

A t-$(v, k, 1)$ design corresponds to an optimal code with length v, constant weight k, and minimum distance $2(k-t+1)$; the points of a block give the 1s in a codeword. So the theory of switching developed in Section 2.2 is directly applicable for $\lambda = 1$; such designs are known as *Steiner systems*.

The smallest possible nontrivial parameters for Steiner systems are $k = 3$, $v = 2$—such *Steiner triple systems* exist exactly for $v \equiv 1, 3 \pmod 6$. A switch of a Steiner triple system must involve at least 4 blocks, and such a switch is essentially unique (the first two points are particularized):

$$\begin{matrix} 1100 & & 0011 \\ 0011 & & 1100 \\ 1010 & & 1010 \\ 1001 & \rightarrow & 1001 \\ 0110 & & 0110 \\ 0101 & & 0101. \end{matrix} \tag{4}$$

The term *Pasch switch* has been used in the literature for this switch as it involves the so-called *Pasch configuration*. Already Fisher, in a paper [28] published in 1940, found out that there is a switching class of Steiner triple systems of order

Table 2
Switching classes of constant weight codes.

n	d	w	$A(n, d, w)$	N	Sizes of switching classes
10	4	3	13	2	2
11	4	3	17	2	2
12	4	3	20	5	5
11	4	4	35	11	8, 1, 1, 1
12	4	4	51	17	13, 2, 1, 1
12	4	5	80	2	2
12	6	5	12	11	4, 2, 1, 1, 1, 1, 1
14	6	7	42	8	6, 2
13	8	5	3	3	1, 1, 1
16	8	5	6	3	1, 1, 1
17	8	5	7	5	1, 1, 1, 1, 1
13	8	6	4	3	1, 1, 1
14	8	6	7	4	1, 1, 1, 1
15	8	6	10	3	1, 1, 1
16	8	6	16	3	1, 1, 1
17	8	6	17	2	1, 1
14	8	7	8	4	1, 1, 1, 1
15	8	7	15	5	1, 1, 1, 1, 1
16	8	7	16	54	4, 3, 3, 2, 1 (42 times)
17	8	7	24	7	1, 1, 1, 1, 1, 1, 1
16	8	8	30	5	1, 1, 1, 1, 1
16	10	6	3	3	1, 1, 1
17	10	6	3	4	1, 1, 1, 1
19	10	6	4	3	1, 1, 1
17	10	7	5	2	1, 1
18	10	7	6	4	1, 1, 1, 1
19	10	7	8	13	1 (13 times)
20	10	7	10	45	1 (45 times)
16	10	8	4	3	1, 1, 1
17	10	8	6	4	1, 1, 1, 1
18	10	8	9	11	1 (11 times)
19	10	8	12	56	3, 2, 2, 2, 2, 2, 2, 1 (41 times)
20	10	8	17	53	1 (53 times)
18	10	9	10	21	1 (21 times)
19	10	9	19	6	1, 1, 1, 1, 1, 1
20	10	10	38	3	1, 1, 1

15 that contains 79 isomorphism classes. This result was later confirmed in [31]. Since the total number of isomorphism classes of Steiner triple systems of order 15 is 80 (a result obtained almost 100 years ago [18]), there is obviously one isomorphism class that is unaffected by Pasch switches (in fact, as the designs of that isomorphism class do not contain any Pasch configurations, no Pasch switches can be carried out). See also [36].

Steiner triple systems with exactly one Pasch configuration before and after the (only possible) switch is carried out have been studied in [37] and are termed *twin systems*. Twin systems reside in switching classes (with respect to Pasch switches) of size 1 or 2, depending on whether the two designs are isomorphic or not.

Switches that can be carried out for Steiner triple systems have a rather specific structure. For a Steiner triple system $(X, \mathcal{B})$, let $\{s, t, u\} \in \mathcal{B}$ be the unique block that contains the pair $\{s, t\}$; the switch will be carried out with respect to these two points. Moreover, let $X' = X \setminus \{s, t, u\}$. Both the blocks that contain s and not t and those that contain t and not s partition X' into pairs. Viewing the elements of X' as a set of vertices of a graph, the aforementioned partitions correspond to perfect matchings in this graph. The union of two perfect matchings is a set of even cycles with one cycle for each minimal switch. Switches of Steiner triple systems are therefore commonly called *cycle switches*. See, for example, [38] for more details.

The (cycle) switching classes for Steiner triple systems of order at most 19 have been determined [38,50]: there is exactly one switching class for each admissible set of parameters. Determination of the switching class(es) of the 11 084 874 829 isomorphism classes of Steiner triple systems of order 19, classified in [51], required a major computational effort [50].

The 3-$(v, 4, 1)$ designs are called *Steiner quadruple systems* and exist exactly for $v \equiv 2, 4 (\text{mod } 6)$. The Steiner quadruple systems have been classified for all admissible orders up to 16. For these small orders there are more than one isomorphism class only for orders 14 and 16, for which there are 4 (classified in [68]) and 1054 163 (classified in [54]) isomorphism classes, respectively.

(Certain types of) switching of Steiner quadruple systems was studied in the 1970s [93] and more recently [105,106]. The smallest number of blocks that can be involved in switching a Steiner quadruple system is 8; such a configuration is unique up to isomorphism and was the main object of interest when switching in the earlier studies.

Switching of the Steiner quadruple systems of orders 14 and 16 was studied in the current work. It turns out that all 4 Steiner quadruple systems of order 14 belong to different switching classes, whereas the Steiner quadruple systems of order 16 are partitioned into 691 switching classes, containing 1043 486, 1853, 951, 920, 676, 584, 495, 427, 398, 384, 382, 381, 338, 307, 180, 166, 103, 94, 77, 59, 54, 53, 43, 41, 36, 32, 25, 22, 18, 17, 16 (5 times), 15, 14 (4 times), 12 (7 times), 11 (2 times), 10 (5 times), 9 (3 times), 8 (11 times), 7 (5 times), 6 (21 times), 5 (14 times), 4 (33 times), 3 (54 times), 2 (118 times), and 1 (378 times) isomorphism classes.

Little has been done on switching Steiner systems with block size greater than 4. In [35], however, switching of a certain 32-block configuration, termed hyperquadrilateral, was considered for 5-$(v, 6, 1)$ designs. The smallest parameter for which new designs could be obtained in [35] was $v = 72$. In fact, designs with no nontrivial automorphisms could be obtained for parameters for which only designs with a rather large automorphism group were previously known. In the current work, switching was applied exhaustively to the 5-$(24, 6, 1)$, 5-$(28, 7, 1)$, 5-$(36, 6, 1)$, and 5-$(48, 6, 1)$ designs with large automorphism groups of the types discussed in [17, Section 5.6], but this did not lead to any new designs.

As a last example, we apply switching to 2-$(v, 4, 1)$ designs with small parameters. Enumeration results of such Steiner systems are surveyed in [84]. It turns out that for $v \leq 25$, all switching classes are singletons. For the next two admissible parameters, $v = 28$ and $v = 37$, Krčadinac [57,58] has shown that the number of isomorphism classes of designs with nontrivial automorphisms is 4466 and at least 51 402, respectively. Applying switching to all these designs leads to a total of 4606 and more than 10^6 designs, respectively, the latter being an improvement of the corresponding bound in [84]. In [4], 187 isomorphism classes of 2-$(28, 4, 1)$ designs with only trivial automorphisms were found; those could be checked against the designs found here and switching could be further applied in an attempt to find even more designs.

The results regarding switching of designs are in most earlier studies explained in terms of trades. Generally speaking, a *trade* for a design with given parameters consists of two disjoint sets of blocks with the property that if the design contains the blocks of one of the sets, then these blocks can be replaced by the blocks of the other set. For more results on trades, see [6,42,56] and their references.

Switches, as defined here, correspond to certain trades, but the whole class of trades is much larger. For example, if a design has a subdesign, then the subdesign can be replaced by a disjoint subdesign with the same parameters, and such a transformation cannot be performed by a switch.

The main practical difference between switches and trades for Steiner systems (the only designs we have considered so far) is that all possible switches with respect to two given points can be computed in a straightforward manner in a time that is at worst quadratic with respect to the number of blocks, as we have seen earlier, whereas for trades it is not even known in general what possible trades there are. Trades have been classified only for certain families of designs and restricted sets of parameters [29]. And even if one would know the possible trades, then determining whether a trade is possible, for example, whether a design contains certain subdesigns or other structures, may require a nontrivial effort [13]. In any case, trades have several other important applications.

The connection between constant weight codes and t-designs have so far concerned the case of $\lambda = 1$ (Steiner systems), where blocks of the design correspond to codewords of the code. If a design is expressed as an $v \times b$ incidence matrix, then the aforementioned connection comes via the 1s in the columns of the incidence matrix.

For 2-designs, it is known [87] that also the 1s in the rows of an incidence matrix correspond to an optimal constant weight code; such a code has length b, weight r, minimum distance $2(r - \lambda)$, and size v (relating to the parameters of the 2-design). However, this code is equidistant and our switching approach is not useful for such codes (see comment in Section 2.2), so new ideas are needed to connect isomorphism classes via switching. Such ideas will be considered in Section 2.4.

Various ways of switching are indeed suggested here for different types of t-designs. Let us have a look at one more class of designs, 1-designs. A 1-design simply corresponds to 0–1 matrices with constant row and column sums. Ryser [85] showed that any two such matrices with given parameters (considered as labeled structures) are connected to each other via a sequence of switches of the following kind:

$$\begin{matrix} 10 \\ 01 \end{matrix} \rightarrow \begin{matrix} 01 \\ 10. \end{matrix} \tag{5}$$

Such switches fit nicely into the framework of switching developed here as two points and blocks with exactly one of these points are considered, and the switch is a complementation.

2.4. Equidistant codes

Although the following discussion will be in the framework of equidistant codes, the main application is obviously 2-designs, so we start by considering constant weight equidistant codes.

As complementation of equidistant constant weight codes in two particularized coordinates only leads to switching classes of size 1, it is clear that we will have to consider more than two coordinates. A consideration of more than two (perhaps even an arbitrary number of) coordinates will not lead to as elegant an approach as in the cases treated earlier, and there might be different ways for carrying out switching, but the presentation here will anyway cover most related earlier results in the literature.

Since the weight of a codeword that is complemented in the particularized coordinates should not change, the number of particularized coordinates, which we denote by m, must be even.

The first task in switching is now to find a set of m coordinates, where each codeword has m, $m/2$, or 0 1s. (The presence of m in the list makes this requirement slightly more general than a direct generalization of the ideas of [22].) Once again we construct a graph $G = (V, E)$, which now has one vertex for each codeword with $m/2$ 1s in the particularized coordinates and edges between vertices whenever the distance within the particularized coordinates between the corresponding codewords differs from $m/2$. If m is not divisible by 4, then the graph G is complete; note that switching in such a situation indeed can produce an inequivalent code whenever $m \geq 4$, whereas switching of all codewords with weight 1 in 2 coordinates cannot, as this is the same as permuting the coordinates.

With $m = 4$, one can see that the Pasch switch in (4) fits into this framework after a transposition of the matrices. A vector with four 1s in the particularized coordinates with $m = 4$ has inner product 0 (in $\mathbb{Z}_2$) with all codewords exactly when the codewords have even weights in these coordinates, that is, when the above mentioned criterion is fulfilled. This means that for designs, the sets of blocks where switching can be carried out for $m = 4$ correspond to the codewords of weight 4 in the dual of the point code.

Denniston [22] applied switching to symmetric 2-(25, 9, 3) designs using $m = 4$. Jungnickel and Tonchev [48,49] considered switching of symmetric designs as well as certain other closely related designs; for example, they remark [49] that the 3 equivalence classes of 2-(16, 6, 2) symmetric designs belong to 1 switching class (symmetric designs with $\lambda = 2$ are known as *biplanes*).

An exhaustive study of switching of 2-designs is not possible within the scope of the current work. There could also be effective variants of switching that still have not been discovered, so the general case with its applications is an intriguing research problem.

The study of equidistant codes need not be restricted to constant weight codes, but general, unrestricted codes can be considered as well. The technique described earlier applies as such to unrestricted codes. Moreover, it can be substantially relaxed, since (a) the codewords may have arbitrary weight in the particularized coordinates and (b) when constructing the graph $G = (V, E)$, there is one vertex for any codeword and edges between vertices whenever the distance within the particularized coordinates between the corresponding codewords differs from $m/2$.

An arbitrary choice of particularized coordinates commonly leads to a situation where the graph G is connected (whereby switching necessarily produces an equivalent code), so it is worthwhile to study conditions that are necessary to make G disconnected.

Hadamard matrices can be viewed as a certain class of equidistant unrestricted codes. A *Hadamard matrix* of order n is an $n \times n$ matrix $\mathbf{H}$ over $\{-1, 1\}$ such that $\mathbf{H}\mathbf{H}^T = n\mathbf{I}$. A necessary condition for the existence of a Hadamard matrix is that $n = 1, 2$ or $n \equiv 0 \pmod 4$. Hadamard matrices are considered in-depth in [20,44].

Two Hadamard matrices are said to be *equivalent* if one can be obtained from the other by permutations of rows and columns followed by possible negations of rows and columns. If the entries -1 of a Hadamard matrix are replaced by 0s, then the columns (alternatively, rows) of the matrix form an equidistant $(n, n, n/2)$ code over $\mathbb{Z}_2$. (One can even get equidistant $(n-1, n, n/2)$ codes from Hadamard matrices of order n, cf. [53, Theorem 2.127].)

Orrick [73] applied switching to Hadamard matrices in the described manner with $m = 4$. Then switching is effective when there are four rows of the Hadamard matrix of the form (-1 is indicated by a minus sign)

$$\left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 1 & & 1 & 1 & 1 & & 1 & 1 & 1 & & 1 & 1 & 1 & & 1 \\ 1 & 1 & \ldots & 1 & 1 & 1 & \ldots & 1 & - & - & \ldots & - & - & - & \ldots & - \\ 1 & 1 & & 1 & - & - & & - & 1 & 1 & & 1 & - & - & & - \\ 1 & 1 & & 1 & - & - & & - & - & - & & - & 1 & 1 & & 1 \end{array}\right]. \tag{6}$$

Obviously, the graph G constructed with respect to four rows with this structure (up to equivalence) consists of four connected components, and switching can be carried out in any of these. A necessary condition for a Hadamard matrix of order n to have 4 rows of the form (6) is that n be divisible by 8, see, for example, [53, Theorem 8.9]. Orrick [73] presents an effective transformation also for orders n not divisible by 8, which – although the author has not been able to formulate this transformation within the switching framework considered here – seems to be *the* switch for such parameters.

The number of equivalence classes of Hadamard matrices is 1 for all admissible orders up to 12; there are 5, 3, 60, and 487 equivalence classes of Hadamard matrices of orders 16, 20, 24, and 28, respectively. Orrick [73] showed that the number of switching classes of Hadamard matrices is 1 for order 16 and 9 for order 24; if transposes of matrices are also considered, then the number for order 24 is reduced to 2 (note that sets of four *columns* of a matrix can also be applied to switching). The transformation carried out in [73] for Hadamard matrices of order 20 and 28 gives 1 and 2 switching classes, respectively, for these parameters.

Orrick also recognizes how the techniques used by Denniston [22] for 2-(25, 9, 3) designs can be explained in the same framework and discusses a possible generalization in [73, Section 3.3]. In [73, Section 3.4] there is a discussion of the relationship between switching and a well-known doubling construction that produces Hadamard matrices of order $2n$ from matrices of order n; recall (Section 2.1) that the origin of switching of codes is also related to a doubling construction!

Switching of D-optimal matrices, which are certain structures closely related to Hadamard matrices, has been studied in [72].

2.5. Group divisible designs

Several types of designs are conveniently considered in the framework of *group divisible designs* (GDDs). A (k, λ)-GDD of type $g_1^{a_1} g_2^{a_2} \cdots g_p^{a_p}$ is a triple $(X, \mathcal{G}, \mathcal{B})$, where X is a set of $\sum_{i=1}^{p} a_i g_i$ points, $\mathcal{G}$ is a partition of X into a_i subsets of size g_i

for $1 \le i \le p$ (called *groups*), and $\mathcal{B}$ is a collection of k-subsets (blocks) of points, such that every 2-subset of points occurs in exactly λ blocks or one group, but not both. If $\lambda = 1$, we write k-GDD instead of $(k, 1)$-GDD. Basic properties of group divisible designs can be found in [70].

The definition of isomorphism for group divisible designs coincides with that for t-designs, with the additional requirement that groups should be mapped to groups.

A Steiner triple system of order v is a 3-GDD of type 1^v. Also Latin squares can be viewed as 3-GDDs. A *Latin square* of order n is an $n \times n$ array with entries from an n-set of symbols such that every symbol occurs exactly once in each row and column. A Latin square of order n is a 3-GDD of type n^3. Two Latin squares whose 3-GDDs are isomorphic are said to belong to the same *main class*. For more information about Latin squares, see [16].

Switching can be applied to k-GDDs exactly as for Steiner systems in Section 2.3, since switching respects the definition of group divisible designs as long as the two particularized points belong to the same group.

As in the case of Steiner triple systems, the auxiliary graph obtained in determining possible switches for (3-GDDs related to) Latin squares consists of cycles only; depending on the origin of the group where the two particularized points reside, the terms *row cycle*, *column cycle*, and *symbol cycle* can be used. (Note, however, that in a general study of main classes, the origins of the groups is irrelevant.) Cycle structures of random Latin squares have been studied in [10].

Norton [71] applied switching to the study of Latin squares (or order 7) already in the 1930s. Parker [81] used switching in a study of Latin squares of order 10. The main focus of these early studies were on Pasch switches; the term *intercalate switches* is used for Latin squares. More recently, Wanless [99] studied the switching classes of Latin squares up to order 8 extensively. As pointed out in [50], a determination of the number of cycle switching classes of the 19 270 853 541 main classes of Latin squares of order 9 seems within reach.

In Section 2.3, transformations that are more general than switches, called trades, were briefly mentioned. Such general transformations have been considered also for Latin squares; these are then called *Latin trades* [6,55,100].

Clearly, switching can be considered for any k-GDDs. One example (also mentioned in [50]) for which no specific switching results seem to be known are the 3-GDDs of type $(2n-1)^1 1^{2n}$, which correspond to 1-factorizations of the complete graph K_{2n}.

2.6. Covering codes

The *covering radius* of a code $C \subseteq \mathbb{Z}_2^n$ is the smallest integer R such that every word in $\mathbb{Z}_2^n$ is at distance at most R from some codeword of C. Alternatively, one may say that the balls of radius R around the codewords of C cover the entire ambient space $\mathbb{Z}_2^n$. A code $C \subseteq \mathbb{Z}_2^n$ with M codewords and covering radius R is called an $(n, M)R$ *covering code*. The smallest possible value of M for which $(n, M)R$ codes exist is denoted by $K(n, R)$, and the corresponding codes are said to be *optimal* (covering codes). See [11] for an in-depth treatment of covering codes.

Perfect codes were mentioned earlier in the discussion of error-correcting codes. Perfect codes are precisely those codes that have covering radius R and minimum distance $2R+1$ for some integer R. This means that the balls of radius R tile the ambient space. See [43,92] for recent surveys of perfect codes.

Switching of $(n, M)R$ codes will be carried out according to Definition 1. Now, however, it is not as straightforward to derive conditions for switching as for error-correcting codes; we shall here propose a certain way of carrying out switches, and then prove that it works. The proposed switch is as follows.

If we alter some codeword $\mathbf{c}$, by changing c_s to $1 - c_s$, then we also change c'_s to $1 - c'_s$ for any codeword $\mathbf{c}'$ such that

$$d_H(\mathbf{c}, \mathbf{c}') \le 2R+1, \qquad d_H(\mathbf{c}, \mathbf{c}') \text{ odd}, \qquad c_s \ne c'_s. \tag{7}$$

Theorem 1. *A switch of an $(n, M)R$ code carried out according to (7) produces another $(n, M)R$ code.*

Proof. Consider an $(n, M)R$ code. Since only one coordinate is affected, any words at distance at most $R-1$ from some codeword will be covered after the switch. To prove the theorem, it therefore suffices to consider words that are at distance R from a codeword that is altered and that has the same value in the altered coordinate before the change. We consider altering the codeword $\mathbf{a}$ in coordinate s and verify that a word $\mathbf{b}$ with $d_H(\mathbf{a}, \mathbf{b}) = R$ and $a_s = b_s$ is covered after the switch.

We now introduce the unique word $\mathbf{c}$ with $d_H(\mathbf{a}, \mathbf{c}) = R+1$, $d_H(\mathbf{b}, \mathbf{c}) = 1$, and $a_s = b_s \ne c_s$. The word $\mathbf{c}$ is covered (before the switch) by some codeword $\mathbf{e}$. Assume that the switch turns $\mathbf{e}$ into $\mathbf{e}'$ (with the possibility that $\mathbf{e} = \mathbf{e}'$ if $\mathbf{e}$ is unaffected by the switch). The core of the proof is split into three subcases.

If $e_s = a_s$ then $d_H(\mathbf{b}, \mathbf{e}) < d_H(\mathbf{c}, \mathbf{e}) \le R$, so $d_H(\mathbf{b}, \mathbf{e}) \le R-1$ and thereby $d_H(\mathbf{b}, \mathbf{e}') \le d_H(\mathbf{b}, \mathbf{e}) + 1 \le (R-1)+1 = R$.

If $e_s \ne a_s$ then $d_H(\mathbf{b}, \mathbf{e}') \le d_H(\mathbf{b}, \mathbf{e})$, so if we also have that $d_H(\mathbf{c}, \mathbf{e}) \le R-1$, then $d_H(\mathbf{b}, \mathbf{e}') \le d_H(\mathbf{b}, \mathbf{e}) = d_H(\mathbf{c}, \mathbf{e}) + 1 \le (R-1)+1 = R$.

We are now left with the situation when $d_H(\mathbf{c}, \mathbf{e}) = R$ and $a_s \ne e_s$ (so $d_H(\mathbf{b}, \mathbf{e}) = R+1$); since $d_H(\mathbf{a}, \mathbf{e})$ is odd and smaller than or equal to $R+1+R = 2R+1$, the conditions of (7) are fulfilled and e_s will be altered in the switch. Now $d_H(\mathbf{b}, \mathbf{e}') = d_H(\mathbf{b}, \mathbf{e}) - 1 = (R+1) - 1 = R$. This completes the proof. □

Table 3
Switching classes of covering codes.

n	R	$K(n, R)$	N	Sizes of switching classes
5	1	7	1	1
6	1	12	2	2
7	1	16	1	1
8	1	32	10	5, 3, 2
7	2	7	3	3
8	2	12	277	9, 8 (8 times), 7 (13 times), 6 (8 times), 5 (5 times), 4 (7 times), 3 (2 times), 2 (3 times)
9	2	16	4	2, 2
9	3	7	8	3, 3, 2

It should be noticed that in the case of perfect codes, the only possible (odd or even) distance between codewords in the range between 1 and $2R + 1$ is $2R + 1$; then the switches defined in Section 2.1 and here coincide.

Although there are many more distances to consider when constructing the graphs for covering codes than for error-correcting codes, the following comment for error-correcting codes applies here as well: two codewords at an odd mutual distance considering all but the particularized coordinate cannot be in the same connected component. So there will generally be nontrivial switches for each coordinate.

The sizes of switching classes of optimal binary covering codes with length at most 9 are listed in Table 3, where once again N denotes the total number of equivalence classes. All switching classes were computed in this work, utilizing the electronic catalog of [53]; see also [53, Table 7.4].

It is known [76] that $K(9, 1) = 62$, and two inequivalent optimal codes have been found; the codes, listed in [77,102], form 1 switching class.

3. Searching for codes and designs

Switching can be used as a part of an algorithm searching for particular codes or designs, in making local changes to structures. A concrete example of the general applicability of this idea comes from the celebrated traveling salesman problem and its k-opt heuristic [46]. There are obviously numerous possibilities for utilizing switching in this context. Anyway, it turns out that a classic method by Stinson can in fact be put into this framework.

Starting from the empty set of blocks, Stinson [95] builds up a Steiner triple system in the following way. As long as the design is not complete, there exists a point that occurs in fewer than b blocks; the algorithm takes one such point x. Then there are necessarily pairs of points, xy and xz, that do not occur in a block. If yz does not occur either, then xyz is added, and the process is iterated. Otherwise, there is a block yzw:

$$\begin{array}{ll} x & 0 \\ y & 1 \\ z & 1 \\ w & 1 \end{array}$$

and we carry out a switch with respect to the points x and w. As there is no word (in the terms of codes) with a 1 in x, a 0 in w, and a 1 in y or z (by the arguments above), we have a connected component a size 1, so we can indeed replace yzw by xyz in a switch.

A modification of this algorithm with the main focus on an improved distribution of random Steiner triple systems can be found in [41]. Stinson's algorithm is more or less directly applicable to any 3-GDDs; such algorithms have indeed been proposed and used for various types of 3-GDDs and related structures [12,23,24,32].

In general, there are two major ways of utilizing switches when searching for a combinatorial structure. Either one can build up the structure by adding new blocks/codewords/etc. whenever possible and otherwise carry out switching. This is quite precisely the approach in Stinson's algorithm described above.

Another possibility is to start from a structure of the right size that violates some main parameter and search for a proper structure via a sequence of switches, utilizing an optimization function to guide the search. In such an approach, certain switches could define the neighborhood for a local search algorithm [33]. Elliott and Gibbons [25] develop an algorithm of this kind for the search of Latin squares without 2×2 subsquares.

4. Conclusions

Many types of structures have been treated in this work, but there are still many other related structures that can be considered in a similar manner. For example, only binary codes are considered here, as only sporadic results [60,61] are known for the general case of q-ary codes. Constant weight codes were here discussed in the packing case and for designs, but these can also be considered in the covering case; such structures are known as *covering designs* (or just *coverings*) [69]. It is not difficult to extend this list further.

The main challenge in extending the list of studied structure is to find a proper definition of a switch. Sometimes there are several natural definitions, and it might even be worth considering different definitions for different subclasses of structures (as we have seen for designs).

The main focus of the current study has been on the definition and unification of switching and on the computation of switching classes. Several other issues related to switching have barely been touched: proving properties of switching, using switching as a means for obtaining theoretical results, and finding structures with particular properties by using switching. Such studies include [59,90–92,96] (codes), [3,52,103,104] (Steiner systems), [83] (Hadamard matrices), and [10,21,66] (Latin squares); this list is certainly far from complete.

The main motivations for the current work were mentioned in the Introduction. In addition to those, switching can be useful in several other ways, some of which are mentioned in the previous paragraph. We shall conclude the paper by briefly mentioning a few more applications.

It was observed in [99] that switching can be used to obtain efficient compression of collections of combinatorial structures. A scheme can be developed where a structure is expressed via the switch required to produce it from another structure. The amount of memory required for specifying the switch can be very small—for example, for a switch applied to one coordinate of a code, only the coordinate and the specific connected component need to be pinpointed. Obviously, at least one structure has to be given explicitly. However, as mentioned in [50], it could be a major challenge to construct the required data structure in cases where the number of structures is huge (which is the case when compression is usually needed).

Finally, switching can be used (as a part) in a scheme for producing random combinatorial structures [1,8,9,41,45,67,95]; note that some of these schemes lead to sequences whose intermediate structures are not necessarily proper. Various types of estimations are also made possible by switching [27,40].

Acknowledgments

The author is grateful to Olli Pottonen and Alfred Wassermann for providing some of the designs that were studied here, and to Doug Weakley for useful comments.

The author was supported by the Academy of Finland under Grants Nos 110196, 130142, and 132122.

References

[1] M. Aryapoor, E.S. Mahmoodian, On uniformly generating Latin squares, Bull. Inst. Combin. Appl. (in press).
[2] S.V. Avgustinovich, F.I. Solov'eva, Construction of perfect binary codes by sequential shifts of $\tilde{\alpha}$-components, Problemy Peredachi Informatsii 33 (3) (1997) 15–21 (in Russian) English translation in Probl. Inf. Transm. 33 (1997) 202–207.
[3] G.K. Bennett, M.J. Grannell, T.S. Griggs, On the bi-embeddability of certain Steiner triple systems of order 15, European J. Combin. 23 (2002) 499–505.
[4] A. Betten, D. Betten, V.D. Tonchev, Unitals and codes, Discrete Math. 267 (2003) 23–33.
[5] Bhagwandas, S.S. Shrikhande, Seidel-equivalence of strongly regular graphs, Sankhyā Ser. A 30 (1968) 359–368.
[6] E.J. Billington, Combinatorial trades: a survey of recent results, in: W.D. Wallis (Ed.), Designs 2002: Further Computational and Constructive Design Theory, Kluwer, Boston, 2003, pp. 47–67.
[7] R.H. Bruck, R.C. Bose, The construction of translation planes from projective spaces, J. Algebra 1 (1964) 85–102.
[8] P.J. Cameron, A Markov chain for Steiner triple systems, Working Document, 2002.
[9] P.J. Cameron, Random strongly regular graphs? Discrete Math. 273 (2003) 103–114.
[10] N.J. Cavenagh, C. Greenhill, I.M. Wanless, The cycle structure of two rows in a random Latin square, Random Structures Algorithms 33 (2008) 286–309.
[11] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, Covering Codes, North-Holland, Amsterdam, 1997.
[12] C.J. Colbourn, Small group divisible designs with block size three, J. Combin. Math. Combin. Comput. 14 (1993) 153–171.
[13] C.J. Colbourn, M.J. Colbourn, D.R. Stinson, The computational complexity of finding subdesigns in combinatorial designs, Ann. Discrete Math. 26 (1985) 59–65.
[14] C.J. Colbourn, D.G. Corneil, On deciding switching equivalence of graphs, Discrete Appl. Math. 2 (1980) 181–184.
[15] C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007.
[16] C.J. Colbourn, J.H. Dinitz, I.M. Wanless, Latin squares, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007, pp. 135–152.
[17] C.J. Colbourn, R. Mathon, Steiner systems, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007, pp. 102–110.
[18] F.N. Cole, L.D. Cummings, H.S. White, The complete enumeration of triad systems in 15 elements, Proc. Acad. Sci. USA 3 (1917) 197–199.
[19] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 2nd ed., MIT Press, Cambridge, MA, 2001.
[20] R. Craigen, H. Kharaghani, Hadamard matrices and Hadamard designs, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007, pp. 273–280.
[21] P. Danziger, I.M. Wanless, B.S. Webb, Monogamous Latin squares, J. Combin. Theory Ser. A 118 (2011) 796–807.
[22] R.H.F. Denniston, Enumeration of symmetric designs (25, 9, 3), Ann. Discrete Math. 15 (1982) 111–127.
[23] J.H. Dinitz, P. Dukes, A.C.H. Ling, Sets of three pairwise orthogonal Steiner triple systems, J. Combin. Theory Ser. A 101 (2003) 90–116.
[24] J.H. Dinitz, D.R. Stinson, A hill-climbing algorithm for the construction of one-factorizations and Room squares, SIAM J. Algebraic Discrete Methods 8 (1987) 430–438.
[25] J.R. Elliott, P.B. Gibbons, The construction of subsquare free Latin squares by simulated annealing, Australas. J. Combin. 5 (1992) 209–228.
[26] T. Etzion, A. Vardy, Perfect binary codes: constructions, properties, and enumeration, IEEE Trans. Inform. Theory 40 (1994) 754–763.
[27] V. Fack, B.D. McKay, A generalized switching method for combinatorial estimation, Australas. J. Combin. 39 (2007) 141–154.
[28] R.A. Fisher, An examination of the different possible solutions of a problem in incomplete blocks, Ann. Eugenics 10 (1940) 52–75.
[29] A.D. Forbes, M.J. Grannell, T.S. Griggs, Configurations and trades in Steiner triple systems, Australas. J. Combin. 29 (2004) 75–84.
[30] D.R. Fulkerson, A.J. Hoffman, M.H. McAndrew, Some properties of graphs with multiple edges, Canad. J. Math. 17 (1965) 166–177.
[31] P.B. Gibbons, Computing techniques for the construction and analysis of block designs, Ph.D. Thesis, University of Toronto, 1976.
[32] P.B. Gibbons, R. Mathon, The use of hill-climbing to construct orthogonal Steiner triple systems, J. Combin. Des. 1 (1993) 27–50.

[33] P.B. Gibbons, P.R.J. Östergård, Computational methods in design theory, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007, pp. 755–783.
[34] C.D. Godsil, B.D. McKay, Constructing cospectral graphs, Aequationes Math. 25 (1982) 257–268.
[35] M.J. Grannell, Some rigid Steiner 5-designs, J. Combin. Des. 18 (2010) 392–400.
[36] M.J. Grannell, T.S. Griggs, J.P. Murphy, Equivalence classes of Steiner triple systems, Congr. Numer. 86 (1992) 19–25.
[37] M.J. Grannell, T.S. Griggs, J.P. Murphy, Twin Steiner triple systems, Discrete Math. 167–168 (1997) 341–352.
[38] M.J. Grannell, T.S. Griggs, J.P. Murphy, Switching cycles in Steiner triple systems, Util. Math. 56 (1999) 3–21.
[39] J. Hage, T. Harju, Towards a characterization of bipartite switching classes by means of forbidden subgraphs, Discuss. Math. Graph Theory 27 (2007) 471–483.
[40] M. Hasheminezhad, B.D. McKay, Combinatorial estimates by the switching method, in: R.A. Brualdi, S. Hedayat, H. Kharaghani, G.B. Khosrovshahi, S. Shahriari (Eds.), Combinatorics and Graphs, Amer. Math. Soc., Providence, 2010, pp. 209–221.
[41] D. Heap, P. Danziger, E. Mendelsohn, Hill-climbing to Pasch valleys, J. Combin. Des. 15 (2007) 405–419.
[42] A.S. Hedayat, G.B. Khosrovshahi, Trades, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007, pp. 644–648.
[43] O. Heden, A survey of perfect codes, Adv. Math. Commun. 2 (2008) 223–247.
[44] K.J. Horadam, Hadamard Matrices and Their Applications, Princeton University Press, Princeton, 2007.
[45] M.T. Jacobson, P. Matthews, Generating uniformly distributed random Latin squares, J. Combin. Des. 4 (1996) 405–437.
[46] D.S. Johnson, L.A. McGeoch, The traveling salesman problem: a case study in local optimization, in: E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, New York, 1997, pp. 215–310.
[47] L.K. Jørgensen, M. Klin, Switching of edges in strongly regular graphs. I. A family of partial difference sets on 100 vertices, Electron. J. Combin. 10 (2003) Research Paper 17.
[48] D. Jungnickel, V.D. Tonchev, Exponential number of quasi-symmetric SDP designs and codes meeting the Grey-Rankin bound, Des. Codes Cryptogr. 1 (1991) 247–253.
[49] D. Jungnickel, V.D. Tonchev, On symmetric and quasi-symmetric designs with the symmetric difference property and their codes, J. Combin. Theory Ser. A 59 (1992) 40–50.
[50] P. Kaski, V. Mäkinen, P.R.J. Östergård, The cycle switching graph of the Steiner triple systems of order 19 is connected, Graphs Combin. (in press).
[51] P. Kaski, P.R.J. Östergård, The Steiner triple systems of order 19, Math. Comp. 73 (2004) 2075–2092.
[52] P. Kaski, P.R.J. Östergård, There exist non-isomorphic STS(19) with equivalent point codes, J. Combin. Des. 12 (2004) 443–448.
[53] P. Kaski, P.R.J. Östergård, Classification Algorithms for Codes and Designs, Springer, Berlin, 2006.
[54] P. Kaski, P.R.J. Östergård, O. Pottonen, The Steiner quadruple systems of order 16, J. Combin. Theory Ser. A 113 (2006) 1764–1770.
[55] A.D. Keedwell, Critical sets in Latin squares and related matters: an update, Util. Math. 65 (2004) 97–131.
[56] G.B. Khosrovshahi, H.R. Maimani, R. Torabi, On trades: an update, Discrete Appl. Math. 95 (1999) 361–376.
[57] V. Krčadinac, Steiner 2-designs $S(2, 4, 28)$ with nontrivial automorphisms, Glas. Mat. Ser. III 37 (57) (2002) 259–268.
[58] V. Krčadinac, Some new Steiner 2-designs $S(2, 4, 37)$, Ars Combin. 78 (2006) 127–135.
[59] D.S. Krotov, P.R.J. Östergård, O. Pottonen, On optimal binary one-error-correcting codes of lengths $2^m - 4$ and $2^m - 3$, IEEE Trans. Inform. Theory (in press).
[60] A.V. Los', Construction of perfect q-ary codes by sequential shifts of the $\tilde{\alpha}$-components, Problemy Peredachi Informatsii 40 (1) (2004) 40–47; and 40 (2) (2004) 94 (in Russian) English translation in Probl. Inf. Transm. 40 (2004) 37–43; and 40 (2004) 186.
[61] A.V. Los', Construction of perfect q-ary codes by switchings of simple components, Problemy Peredachi Informatsii 42 (1) (2006) 34–42 (in Russian) English translation in Probl. Inf. Transm. 42 (2006) 30–37.
[62] F.J. MacWilliams, N.J.A. Sloane, The Theory of Error-Correcting Codes, North-Holland, Amsterdam, 1977.
[63] S.A. Malyugin, On enumeration of the perfect binary codes of length 15, Diskretn. Anal. Issled. Oper. Ser. 2 6 (2) (1999) 48–73 (in Russian). English translation in Discrete Appl. Math. 135 (2004) 161–181.
[64] S.A. Malyugin, On enumeration of nonequivalent perfect binary codes of length 15 and rank 15, Diskretn. Anal. Issled. Oper. Ser. 1 13 (1) (2006) 77–98 (in Russian) English translation in J. Appl. Ind. Math. 1 (2007) 77–89.
[65] R. Mathon, A.P. Street, Overlarge sets and partial geometries, J. Geom. 60 (1997) 85–104.
[66] B.D. McKay, J.C. McLeod, I.M. Wanless, The number of transversals in a Latin square, Des. Codes Cryptogr. 40 (2006) 269–284.
[67] B.D. McKay, N.C. Wormald, Uniform generation of random Latin rectangles, J. Combin. Math. Combin. Comput. 9 (1991) 179–186.
[68] N.S. Mendelsohn, S.H.Y. Hung, On the Steiner systems $S(3, 4, 14)$ and $S(4, 5, 15)$, Util. Math. 1 (1972) 5–95.
[69] W.H. Mills, R.C. Mullin, Coverings and packings, in: J.H. Dinitz, D.R. Stinson (Eds.), Contemporary Design Theory: A Collection of Surveys, Wiley, New York, 1992, pp. 371–399.
[70] R.C. Mullin, H.-D.O.F. Gronau, PBDs and GDDs: the basics, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007, pp. 231–236.
[71] H.W. Norton, The 7×7 squares, Ann. Eugenics 9 (1939) 269–307.
[72] W.P. Orrick, On the enumeration of some D-optimal designs, J. Statist. Plann. Inference 138 (2008) 286–293.
[73] W.P. Orrick, Switching operations for Hadamard matrices, SIAM J. Discrete Math. 22 (2008) 31–50.
[74] P.R.J. Östergård, Classification of binary constant weight codes, IEEE Trans. Inform. Theory 56 (2010) 3779–3785.
[75] P.R.J. Östergård, T. Baicheva, E. Kolev, Optimal binary one-error-correcting codes of length 10 have 72 codewords, IEEE Trans. Inform. Theory 45 (1999) 1229–1231.
[76] P.R.J. Östergård, U. Blass, On the size of optimal binary codes of length 9 and covering radius 1, IEEE Trans. Inform. Theory 47 (2001) 2556–2557.
[77] P.R.J. Östergård, H.O. Hämäläinen, New upper bounds for binary/ternary mixed covering codes, Res. Rep. A22, Digital Systems Laboratory, Helsinki University of Technology, 1993.
[78] P.R.J. Östergård, O. Pottonen, The perfect binary one-error-correcting codes of length 15: part I—classification, IEEE Trans. Inform. Theory 55 (2009) 4657–4660. Codes at: arXiv:0806.2513v3.
[79] P.R.J. Östergård, O. Pottonen, Two optimal one-error-correcting codes of length 13 that are not doubly shortened perfect codes, Des. Codes Cryptogr. 59 (2011) 281–285.
[80] P.R.J. Östergård, O. Pottonen, K.T. Phelps, The perfect binary one-error-correcting codes of length 15: part II—properties, IEEE Trans. Inform. Theory 56 (2010) 2571–2582.
[81] E.T. Parker, Computer investigation of orthogonal Latin squares of order ten, in: Proc. Sympos. Appl. Math., vol. XV, Amer. Math. Soc., Providence, 1963, pp. 73–81.
[82] K.T. Phelps, M. LeVan, Switching equivalence classes of perfect codes, Des. Codes Cryptogr. 16 (1999) 179–184.
[83] K.T. Phelps, J. Rifà, M. Villanueva, Rank and kernel of binary Hadamard codes, IEEE Trans. Inform. Theory 51 (2005) 3931–3937.
[84] C. Reid, A. Rosa, Steiner systems $S(2, 4, v)$—a survey, Electron. J. Combin. Dynamic Survey DS18.
[85] H.J. Ryser, Combinatorial properties of matrices of zeros and ones, Canad. J. Math. 9 (1957) 371–377.
[86] J.J. Seidel, Strongly regular graphs of L_2-type and of triangular type, Indag. Math. 29 (1967) 188–196.
[87] N.V. Semakov, V.A. Zinov'ev, Balanced codes and tactical configurations, Problemy Peredachi Informatsii 5 (3) (1969) 28–37 (in Russian) English translation in Probl. Inf. Transm. 5 (3) (1969) 22–28.
[88] S.S. Shrikhande, V.N. Bhat, Nonisomorphic solutions of pseudo-(3, 5, 2) and pseudo-(3, 6, 3) graphs, Ann. New York Acad. Sci. 175 (1970) 331–350.
[89] F.I. Solov'eva, Factorization of code-generating disjunctive normal forms, Metody Diskret. Anal. 47 (1988) 66–88 (in Russian).

[90] F.I. Solov'eva, Switchings and perfect codes, in: I. Althöfer, N. Cai, G. Dueck, L. Khachatrian, M.S. Pinsker, A. Sárközy, I. Wegener, Z. Zhang (Eds.), Numbers, Information and Complexity, Kluwer, Boston, 2000, pp. 311–324.
[91] F.I. Solov'eva, Structure of *i*-components of perfect binary codes, Discrete Appl. Math. 111 (2001) 189–197.
[92] F.I. Solov'eva, On perfect binary codes, Discrete Appl. Math. 156 (2008) 1488–1498.
[93] F.I. Solov'eva, M.F. Semenov, Steiner quadruple systems and the process of an exchange, presented at III All-Union Conference on Problems of Theoretical Cybernetics, Institute of Mathematics of the Siberian Division of the Academy of Sciences of the USSR, Novosibirsk, 1974 (in Russian).
[94] E. Spence, Two-graphs, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, 2nd ed., Chapman & Hall, CRC, Boca Raton, 2007, pp. 875–882.
[95] D.R. Stinson, Hill-climbing algorithms for the construction of combinatorial designs, Ann. Discrete Math. 26 (1985) 321–334.
[96] N.N. Tokareva, Components of preparata codes, Problemy Peredachi Informatsii 40 (2) (2004) 63–69 (in Russian) English translation in Probl. Inf. Transm. 40 (2004) 159–164.
[97] J.H. van Lint, J.J. Seidel, Equilateral point sets in elliptic geometry, Indag. Math. 28 (1966) 335–348.
[98] Ju.L. Vasil'ev, On nongroup close-packed codes, Problemy Kibernet. 8 (1962) 337–339 (in Russian).
[99] I.M. Wanless, Cycle switches in Latin squares, Graphs Combin. 20 (2004) 545–570.
[100] I.M. Wanless, A computer enumeration of small Latin trades, Australas. J. Combin. 39 (2007) 247–258.
[101] D.B. West, Introduction to Graph Theory, 2nd ed., Prentice Hall, Upper Saddle River, 2001.
[102] L.T. Wille, New binary covering codes obtained by simulated annealing, IEEE Trans. Inform. Theory 42 (1996) 300–302.
[103] H. Zeitler, Eine Klasse spezieller gemischter Ovale in STS(v), J. Geom. 33 (1988) 185–202.
[104] H. Zeitler, Modifications of the "central-method" to construct Steiner triple systems, Discrete Math. 77 (1989) 367–384.
[105] V.A. Zinov'ev, D.V. Zinov'ev, On the switching construction of Steiner quadruple systems, in: Proc. 11th Int'l Workshop on Algebraic and Combinatorial Coding Theory, Pamporovo, Bulgaria, June 16–22, 2008, pp. 318–324.
[106] V.A. Zinov'ev, D.V. Zinov'ev, On one transformation of Steiner quadruple systems $S(v, 4, 3)$, Problemy Peredachi Informatsii 45 (4) (2009) 26–42 (in Russian) English translation in Probl. Inf. Transm. 45 (2009) 317–332.