Switching Combinatorial Objects

Patric R. J. Östergård
Department of Communications and Networking
Aalto University
P.O. Box 13000, 00076 Aalto, Finland
E-mail: patric.ostergard@tkk.fi
(Currently visiting Universität Bayreuth, Germany.)

Joint work with Petteri Kaski, Veli Mäkinen, and Olli Pottonen.

The research was supported in part by the Academy of Finland.

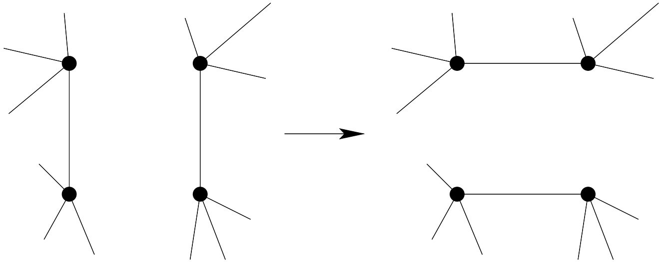
Switching

trade
: A transformation that leaves the main (basic as well as regularity) parameters of a combinatorial object unchanged.

switch
: A *local* transformation that leaves the main (basic as well as regularity) parameters of a combinatorial object unchanged.

Example: Switching

2-switch of a graph.



History of Switching

Norton (1939) and Fisher (1940) Latin squares and Steiner triple systems [F,N].

Vasil'ev (1962) (Perfect) codes [V].

Van Lint and Seidel (1966) Graphs (*Seidel switching*) [LS].

[F] R. A. Fisher, An examination of the different possible solutions of a problem in incomplete blocks, *Ann. Eugenics* **10** (1940), 52–75.

[N] H. W. Norton, The 7 x 7 squares, *Ann. Eugenics* **9** (1939), 269–307.

[V] Ju. L. Vasil'ev, On nongroup close-packed codes, (in Russian), *Problemy Kibernet.* **8** (1962), 337–339.

[LS] J. H. van Lint and J. J. Seidel, Equilateral point sets in elliptic geometry, *Indag. Math.* **28** (1966), 335–348.

Why Switch?

There are many reasons for switching, including the following:

1. As a part of a mathematical proof.
2. To define neighbors in a local search algorithm.
3. To try to find new combinatorial objects from old ones.
4. In order to gain understanding in why there are so many equivalence/isomorphism classes of objects with certain parameters.

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

000000000111111111
011101001000110 1
000111100011100 1
001100111001001 1
001011010101010 1
011010100100101 1
010110010010011 1

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

```
000000001111111
011101001000110 1
```

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

000001001111111
011101001000110
000111100011100
001100111001001
001011010101010
011010100100101
010110010010011

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

```
000001001111111
011101001000110 1
```

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

000001001110001 1
011101001000110 1
000111100011100 1
001100111001001 1
001011010101010 1
011010100100101 1
010110010010011 1

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

000001001110011
011101001000110 1
000111100011100 1
001100111001001 1
001011010101010 1
011010100100101 1
010110010010011 1

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

010000111111100011
011101001000110 1
000111100011100 1
001100111001001 1
001011010101010 1
011010100100101 1
010110010010011 1

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

010001111110001 1
011101001000110 1
000111100011100 1
001100111001001 1
001011010101010 1
011010100100101 1
010110010010011 1

Switching Binary Codes

All codes in the sequel are *binary*.

Example. Code with minimum distance 3.

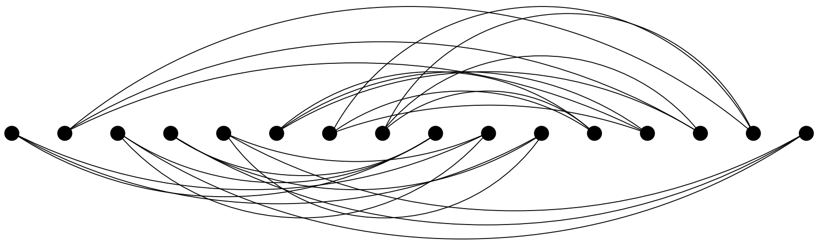
```
010001111110000 1
011101001000110 1
000111100011100 1
001100111001001 1
001011010101010 1
011010100100101 1
010110010010011 1
```

Switching via an Auxiliary Graph

1. Consider a particular coordinate i.
2. Construct a graph G with one vertex for each codeword and an edge between two vertices that differ in the ith coordinate and whose mutual distance equals the minimum distance of the code.
3. Complement the ith coordinate in a connected component of the graph G.

Example: Auxiliary Graph

For the previous example we get the following auxiliary graph with respect to the first coordinate:



Switching Graph and Switching Classes

switching graph A graph with one vertex for each equivalence class of codes and with an edge if there is a switch taking a code from one class to the other.

switching class A connected component of the switching graph, in other words, a complete set of (equivalence classes of) codes connected via a sequence of switches.

Example: Switching Optimal 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	7398	7013, 385
15	3	2048	5983	5819, 153, 3, 2, 2, 1, 1, 1, 1

Switching Constant Weight Codes

The aforementioned switch changes the Hamming weight of codewords.

$\Rightarrow$

If we consider codes with constant Hamming weight, then we need to apply a switch in a different way.

How?

Switching Constant Weight Codes

The aforementioned switch changes the Hamming weight of codewords.

$\Rightarrow$

If we consider codes with constant Hamming weight, then we need to apply a switch in a different way.

How?

Apply switching to a *pair* of coordinates.

Switching Steiner Systems

Steiner systems can be viewed as optimal constant weight codes.

0000111
1100100
0110001
0011100
0101010
1010010
1001001

Switching Steiner Systems

Steiner systems can be viewed as optimal constant weight codes.

0000111
1100100
0110001
0011100
0101010
1010010
1001001

Switching Steiner Systems

Steiner systems can be viewed as optimal constant weight codes.

1000111
0100100
0110001
0011100
0101010
1010010
1001001

Switching Steiner Systems

Steiner systems can be viewed as optimal constant weight codes.

1000111
0100100
0110001
0011100
0101010
1010010
1001001

Switching Steiner Systems

Steiner systems can be viewed as optimal constant weight codes.

1000100
0100111
0110001
0011100
0101010
1010010
1001001

Switching Steiner Systems

Steiner systems can be viewed as optimal constant weight codes.

1000100
0100111
0110001
0011100
0101010
1010010
1001001

Switching Steiner Systems

Steiner systems can be viewed as optimal constant weight codes.

1100100
0000111
0110001
0011100
0101010
1010010
1001001

We have seen an example of the well-known **Pasch switch**!

Switching Covering Codes

A covering code has the property that all words in the ambient space are within Hamming distance R from some codeword.

How to switch a covering code with codewords $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ in some coordinate s ?

Criterion for edges in auxiliary graph:

$$d_H(\mathbf{c}, \mathbf{c}') \leq 2R + 1, \quad d_H(\mathbf{c}, \mathbf{c}') \text{ odd}, \quad c_s \neq c'_s, \tag{1}$$

Switching Covering Codes: Outline of Proof

It suffices to consider a word **b** that is at distance R from a codeword **a** that is altered and the case when $a_s = b_s$.

Consider the word **c**, which coincides with **b**, except that $a_s = b_s \neq c_s$; and also consider the word **e** that covers **c**. We get three cases:

1) $e_s = b_s$: $\Rightarrow d_H(\mathbf{b}, \mathbf{e}) \leq R - 1$.
2) $e_s \neq b_s$ and $d_H(\mathbf{c}, \mathbf{e}) \leq R - 1$:. . . .
3) $e_s \neq b_s$ and $d_H(\mathbf{c}, \mathbf{e}) = R$: $\Rightarrow d_H(\mathbf{a}, \mathbf{e})$ is odd and smaller than or equal to $R + 1 + R = 2R + 1 \Rightarrow$ the conditions of (1) are fulfilled.

Example: Switching 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

The two known codes attaining $K(9,1) = 62$ belong to one switching class.

Switching Steiner Triple Systems

Result 1. The switching graph of the 11 084 874 829 isomorphism classes of Steiner triple systems of order 19 is connected.

Corollary. The switching graph of the labeled 1 348 410 350 618 155 344 199 680 000 designs is connected.

[KMO] P. Kaski, V. Mäkinen, and P. R. J. Ö., The cycle switching graph of the Steiner triple systems of order 19 is connected, submitted for publication.

Switching Steiner Triple Systems: The Search

(BFS and DFS require too much memory.)

1. Random walk of 10 000 000 000 steps. This spans 6 438 182 977 isomorphism classes in 8 CPU days.
2. BFS from the representatives of the isomorphism classes not spanned. 13 CPU days.

Total CPU: $\approx$ 1 month
Max memory: $\approx$ 93 GB

Switching Steiner Triple Systems: Some Challenges

- The graph is implicit and the vertices are isomorphism classes: a switch from a representative of an isomorphism class need not result in a representative.
- Representatives of the 11 billion odd isomorphism classes can be compressed into 39 GB, but that database is not searchable.

Solution:

- Design of *injective* hash function that takes the 11 billion odd canonical representatives to unique 72-bit values.
- After (radix) sorting the hash values, to obtain a (binary-)searchable database, the data is prefix-compressed. $\rightarrow$ 63 GB

Switching Designs: Some Results

Result 2. The 1 054 163 isomorphism classes of Steiner quadruple systems of order 16 belong to switching classes of size 1 043 486, 1 853, 951, 920, 676, 584, 495, 427,. . . ,1.

Result 3. Krčadinac [K,RR] has shown that the number of isomorphism classes of $S(2,4,37)$ designs is at least 51 402. Switching shows that this number is $> 1\,000\,000$.

[K] V. Krčadinac, Some new Steiner 2-designs $S(2,4,37)$, *Ars Combin.* **78** (2006), 127–135.

[RR] C. Reid and A. Rosa, Steiner systems $S(2,4,v)$—A survey, *Electron. J. Combin.*, Dynamic Survey DS18.

Switching Codes: A Result

Theorem A. (Best & Brouwer, 1977) When 1-perfect codes are shortened once, twice, or three times, one gets optimal one-error-correcting codes.

Theorem B. (Blackmore, 1999) The inverse of Theorem A holds for codes with the parameters of 1-perfect codes shortened once.

Result 4. (Ö. & Pottonen [OP]) The inverse of Theorem A does not always hold for codes with the parameters of 1-perfect codes shortened twice.

Proof. Switching the codes obtained by shortening the 1-perfect codes of length 15 twice gives two new codes.

[OP] P. R. J. Ö. and O. Pottonen, Two optimal one-error-correcting codes of length 13 that are not doubly shortened perfect codes, *Des. Codes Cryptogr.*, to appear.

The End

THANK YOU!!!