International Workshop On Combinatorial Algorithms
IWOCA

http://www.iwoca.org—Problems section.

Orthogonal labelings in de Bruijn graphs

Luca Mariot
Delft University of Technology, The Netherlands
L.Mariot@tudelft.nl

presented at IWOCA 2020
10-6-2020

Problem Statement

Let $G_{m,n} = (V, E)$ be the de Bruijn graph of order n over a set S of m symbols. An edge labeling function $l : E \to S$ for $G_{m,n}$ is called *bipermutative* if, for any vertex $v \in V$, both the labels on the ingoing and outgoing edges of v form a permutation of S.

Given two labelings $l_1, l_2 : E \to S$, define the superposed labeling $l_1.l_2 : E \to S \times S$ as $l_1.l_2(u, v) = (l_1(u, v), l_2(u, v))$ for all $(u, v) \in E$. Two bipermutative labelings l_1, l_2 are *orthogonal* if, for each pair of strings $(x, y) \in S^n \times S^n$, there exists *exactly one* path in $G_{m,n}$ of length n which is labeled by (x, y) under the superposed labeling $l_1.l_2$.

As an example, let $m = n = 2$, and consider the de Bruijn graph $G_{2,2}$ in Figure 1. It can be checked that the two labelings l_1, l_2 defined in the table are indeed bipermutative and orthogonal, i.e. for each of the 16 pairs of 2-bit strings $p = (x_1x_2, y_1y_2)$ there is exactly one path of length 2 in $G_{2,2}$ which is labeled by p. We now state the problems concerning orthogonal labelings that we are interested in.

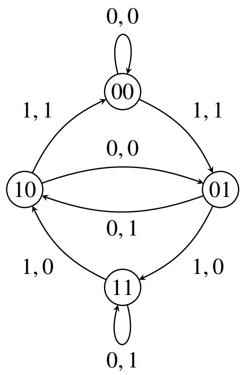
Problem 1 (Counting)**.** Find a formula for counting the number $N(m, n)$ of orthogonal pairs of bipermutative labelings for $G_{m,n}$.

Problem 2 (Enumeration)**.** Find an algorithm that, given m and n in input, enumerates all $N(m, n)$ orthogonal pairs of bipermutative labelings for $G_{m,n}$.

Context

Orthogonal labelings on de Bruijn graphs are closely connected with *orthogonal Latin squares* defined by *cellular automata* (CA). A CA of length t and diameter $d \leq t$ over an alphabet S is a vectorial function $F : S^t \to S^{t-d+1}$ defined for all $x \in S^t$ as:

$$F(x_1, \cdots, x_t) = (f(x_1, \cdots, x_d), f(x_2, \cdots, x_{d+1}), \cdots, f(x_{t-d+1}, \cdots, x_t)) \quad . \tag{1}$$



$(v_1, v_2) \to (u_1, u_2)$	l_1	l_2
$00 \to 00$	0	0
$10 \to 00$	1	1
$01 \to 10$	0	1
$11 \to 10$	1	0
$00 \to 01$	1	1
$10 \to 01$	0	0
$01 \to 11$	1	0
$11 \to 11$	0	1

Figure 1: Example of orthogonal bipermutative labelings for the de Bruijn graph $G_{2,2}$.

where $f : S^d \to S$ is called the *local rule* of the CA F. As shown for example in [4], a CA $F : S^t \to S^{t-d+1}$ over the alphabet S can be described by a de Bruijn graph $G_{m,n}$ where $n = d - 1$ and $m = |S|$, and where the edge labeling function corresponds to the local rule f. The CA input is a path of length $t - d + 1$ *on the vertices* of $G_{m,n}$, obtained by overlapping all vertices in the path to get a string $x \in S^t$, while $F(x)$ is the corresponding path *on the edges*, obtained by concatenating the labels.

Recall that a Latin square of order N is a square $N \times N$ matrix such that, in each row and in each column, each number from 1 to N occurs exactly once. It has been shown in [2] that if the labeling induced by the local rule f on the de Bruijn graph is bipermutative, the CA $F : S^{2(d-1)} \to S^{d-1}$ defines a Latin square of order m^{d-1}. Moreover, two orthogonal bipermutative labelings on $G_{m,n}$ induce two CA such that their Latin squares are *orthogonal*, i.e. each pair of numbers from 1 to m^{d-1} occurs exactly once in the superposition of the two squares. Therefore, solving Problem 1 is equivalent to counting the number of orthogonal Latin squares of order m^{d-1} that are generated by CA. This problem has been solved in [2] for the particular case where the alphabet S is the finite field $\mathbb{F}_q$ and the local rules are linear maps on the vector space $\mathbb{F}_q^d$. However, there is no known general formula for $N(n, m)$ up to now. Similarly, Problem 2 is equivalent to designing an algorithm that enumerates all pairs of orthogonal Latin squares of order m^{d-1} defined by CA. There exists an enumeration algorithm described in [1], which however explores a superset of CA-based orthogonal Latin squares, and then filters only the orthogonal pairs. Additionally, evolutionary algorithms have been used in [3] to construct single orthogonal pairs.

References

[1] L. Mariot, E. Formenti, and A. Leporati. Enumerating orthogonal latin squares generated by bipermutive cellular automata. In A. Dennunzio, E. Formenti, L. Manzoni, and A. E. Porreca, editors, *AUTOMATA 2017*, volume 10248 of *LNCS*, pages 151–164. Springer, 2017.

[2] L. Mariot, M. Gadouleau, E. Formenti, and A. Leporati. Mutually orthogonal latin squares based on cellular automata. *Des. Codes Cryptogr.*, 88(2):391–411, 2020.

[3] L. Mariot, S. Picek, D. Jakobovic, and A. Leporati. Evolutionary algorithms for the design of orthogonal latin squares based on cellular automata. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017*, pages 306–313, 2017.

[4] K. Sutner. De bruijn graphs and linear cellular automata. *Complex Syst.*, 5(1), 1991.