Left MQQs whose left parastrophe is also quadratic

Simona Samardjiska, Danilo Gligoroski

Abstract. A left quasigroup (Q, q) of order 2^w that can be represented as a vector of Boolean functions of degree 2 is called a left multivariate quadratic quasigroup (LMQQ). For a given LMQQ there exists a left parastrophe operation $q_\backslash$ defined by: $q_\backslash(u, v) = w \Leftrightarrow q(u, w) = v$ that also defines a left multivariate quasigroup. However, in general, $(Q, q_\backslash)$ is not quadratic. Even more, representing it in a symbolic form may require exponential time and space. In this work we investigate the problem of finding a subclass of LMQQs whose left parastrophe is also quadratic (i.e. is also an LMQQ), and in the same time can be easily constructed. These LMQQs are affine in the second argument, and their left parastrophe can be easily expressed from the quasigroup operation. We give necessary and sufficient conditions for an LMQQ of this type to have a left parastrophe that is also an LMQQ. Based on this, we distinguish a special class that satisfies our requirements and whose construction is deterministic and straightforward.

Keywords: left multivariate quadratic quasigroup, left parastrophe, algebraic degree, matrix of Boolean polynomials

Classification: 20N05, 11T55, 11T71

1. Introduction

The potential and usefulness of quasigroups (or equivalently Latin squares) in the design of different types of cryptographic primitives and codes has been addressed in numerous works, beginning with the seminal work of Shannon [21] more than half a century ago. Since then, quasigroups were incorporated in the design of many different cryptographic schemes as well as codes. We can mention some of them:

- *Secret sharing schemes:* Cooper et al. [7] designed a secret sharing scheme arising from Latin squares,
- *Block Ciphers:* A version of the block cipher DES that uses Latin squares was proposed by Carter et al. [4],
- *Hash functions:* The need of using quasigroups in the design of cryptographic hash functions was discussed by Schnorr and Vaudenay in [20], and later, in the SHA-3 hash competition, at least three functions had quasigroups or left quasigroups in their design (Edon-R [11], NaSHA [14] and Blue Midnight Wish [12]),
- *Stream Ciphers:* The fast software stream cipher CryptMT by Matsumoto et al. [15] actually uses quasigroups that belong to the class of polynomial quasigroups analyzed by Rivest in [17],

- *Hardware stream cipher:* A hardware stream cipher Edon80 using four different quasigroups of order 4 was proposed in [8],
- *Coding theory:* Latin squares were used for designing LDPC codes in [23], [16].

Recently, in [9], a new class of quasigroups called *Multivariate Quadratic Quasigroups* (MQQs) was introduced. The distinctive property of these quasigroups is that when represented as Boolean functions in their algebraic normal form, they are multivariate quadratic. MQQs have found an application [9], [10] in the field of Multivariate cryptography, or MQ (multivariate quadratic) cryptography. MQ schemes have performance advantages over the classical public key schemes based on integer factorization (RSA) and on the discrete logarithm problem in the additive group of points defined by elliptic curves over finite fields (ECC). Additionally, they are considered as one of the post-quantum alternatives to the most popular RSA and ECC schemes, since there are no known quantum algorithms that would break MQ schemes. However, they have one disadvantage — the size of the public/private key pair is much bigger than in the currently used cryptosystems.

The authors of [9] constructed only MQQs of lower orders (up to 2^5). In [2], a randomized algorithm was proposed to generate MQQs of higher orders, but just up to 2^{14}. In [5], a method for construction of bilinear MQQs was proposed. A detailed survey on the properties and construction of multivariate quadratic loops and quasigroups was given in [6]. In [19], an approach was taken to construct quasigroups based on T-functions defined by Klimov and Shamir [13]. These quasigroups were called T-multivariate quasigroups, and can be (but are not exclusively limited to be) quadratic. An extension of the algorithms from [5] and [19] to arbitrary Galois fields $\mathbb{F}_{p^k}$ was recently given in [18].

In this paper we continue the analysis of MQQs, by investigating the wider class of Left MQQs (LMQQs), and distinguishing subclasses that are of special interest for cryptographic use in multivariate public key schemes. More concretely, since in general a parastrophe of an LMQQ is not quadratic and representing it in a symbolic form may require exponential time and space, it is a challenging problem to find a subclass of LMQQs whose parastrophes are also quadratic, and in the same time can be easily constructed.

1.1 Contribution and organization of the paper. We first introduce and give a general construction of left multivariate quasigroups (LMQs) of any order 2^w and any degree, and afterwards focus on the properties of a subclass of the class of all LMQs of order 2^w that consists of left quasigroups affine in the second argument, whose left parastrophe can be easily expressed.

We then distinguish a special family of LMQQs and give the necessary and sufficient conditions for these LMQQs to have a left parastrophe that has degree 2, i.e., is also an LMQQ. As this characterization does not provide an algorithmic construction of this type of LMQQs, we further refine the requirements at several stages, to finally reach very simple sufficient conditions for an LMQQ to have a quadratic left parastrophe and provide an especially easy construction procedure.

The paper is organized as follows: The preliminaries are given in Section 2; in Section 3 we investigate LMQs of several different types, give effective constructions, and determine the relations between them; Section 4 is devoted to finding and analyzing different sufficient conditions for an LMQQ to have a quadratic left parastrophe in order to find suitable ones that give a simple and easy algorithmic procedure for their construction. The conclusions are given in Section 5.

2. Preliminaries

2.1 Quasigroups. The following definitions and basic properties can be found in classic textbooks on quasigroup theory, such as Belousov's [3], or Smith's [22].

Let (Q,q) be a groupoid and let a be a fixed element of Q. The mappings $L_{q,a}, R_{q,a} : Q \to Q$, called left and right translations (translation mappings), are defined by:

$$L_{q,a}(x) = q(a,x), \quad R_{q,a}(x) = q(x,a),$$

for every $x \in Q$.

Definition 1. The groupoid (Q,q) is called a left (right) quasigroup if the mapping $L_{q,a}$ $(R_{q,a})$ is a permutation of Q for every $a \in Q$.

If (Q,q) is both left and right quasigroup, then it is simply called a quasigroup.

If a quasigroup (Q,q) has a unit element e, then (Q,q) is called a loop.

A finite (left/right) quasigroup of n elements is said to be a (left/right) quasigroup of order n.

Definition 2. Given a (left) quasigroup (Q,q) a new (left) quasigroup operation $q_\backslash$ can be defined on the set Q by

$$q_\backslash(u,v) = w \Leftrightarrow q(u,w) = v,$$

called a left parastrophe operation. The two operations satisfy the identities

$$q(u, q_\backslash(u,v)) = v, \quad q_\backslash(u, q(u,v)) = v, \tag{1}$$

for all $u, v \in Q$.

Let $\mathcal{Q}_n$ be a set of all left quasigroup operations over the set Q of n elements, and let $\mathcal{S}_Q$ be the symmetric group upon Q. Since a left quasigroup from $\mathcal{Q}_n$ can be considered as a collection of n permutations from $\mathcal{S}_Q$, the definition of composition of permutations from $\mathcal{S}_Q$ can be naturally extended to $\mathcal{Q}_n$.

Let $q_1, q_2 \in \mathcal{Q}_n$. A composition of q_1 and q_2 is defined by:

$$(q_1 \circ q_2)(u,v) = q_1(u, q_2(u,v)), \quad \text{for all} \;\; u,v \in Q.$$

Moreover, it is not hard to see that the following holds.

Proposition 1. *$(\mathcal{Q}_n, \circ)$ is a group isomorphic to $(\mathcal{S}_Q)^n$.* □

Definition 3. Two (left) quasigroups (Q, q_1) and (Q, q_2) are said to be isotopic, if there exist permutations $\alpha, \beta, \gamma \in \mathcal{S}_Q$ such that

$$\gamma(q_1(u,v)) = q_2(\alpha(u), \beta(v)), \quad \text{for all} \quad u, v \in Q.$$

We denote the isotopy by (α, β, γ). If $\alpha = \beta = \gamma$ we say that the (left) quasigroups are isomorphic.

Using the definition, we can efficiently construct new (left) quasigroups isotopic to a known one.

Proposition 2 ([1]). *Given a binary (left) quasigroup (Q, q), and permutations $\alpha, \beta, \gamma \in \mathcal{S}_Q$, the operation q' defined by*

$$q'(u,v) = \gamma^{-1}(q(\alpha(u), \beta(v))), \quad \textit{for all} \quad u, v \in Q,$$

defines a (left) quasigroup (Q, q') isotopic to (Q, q). (α, β, γ) is an isotopy from (Q, q) to (Q, q').

In the rest of the paper we will be mainly interested in properties of finite left quasigroups.

2.2 Left Multivariate Quasigroups. We will use the following notations.

Let $\mathbb{F}_2$ denote the Galois field of order 2, and $\mathbb{F}_2[x_1, x_2, \ldots, x_\nu]$ the ring of polynomials in the variables $x_1, x_2, \ldots, x_\nu$ over the field $\mathbb{F}_2$.

We will call the elements of the quotient ring $\mathbb{F}_2[x_1, x_2, \ldots, x_\nu]/(x_1^2 - x_1, x_2^2 - x_2, \ldots, x_\nu^2 - x_\nu)$ Boolean polynomials.

We will consider the elements $(u_1, u_2, \ldots, u_w) \in \mathbb{F}_2^w$ as column vectors and use the notation $\mathbf{u} = (u_1, u_2, \ldots, u_w)$. Furthermore, for $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) \in (\mathbb{F}_2^w)^m$ we denote by $u_{s,j}$ the s-th bit of the j-th component $\mathbf{u}_j$.

Let $f : (\mathbb{F}_2^w)^m \to \mathbb{F}_2^w$ be a mapping, and let $f(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)_s$ denote the s-th bit of $f(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)$. The function f can be represented as a w-tuple of Boolean functions as $f = (f^{(1)}, f^{(2)}, \ldots, f^{(w)})$, where $f^{(s)} : (\mathbb{F}_2^w)^m \to \mathbb{F}_2$ for every $s = 1, \ldots, w$, and $f^{(s)}(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) = f(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)_s$ for every $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m) \in (\mathbb{F}_2^w)^m$.

It is a well known fact that every Boolean function $g : (\mathbb{F}_2^w)^m \to \mathbb{F}_2$ can be represented uniquely by its Algebraic Normal Form (ANF) as a Boolean polynomial in mw variables $\hat{g} \in \mathbb{F}_2[x_{1,1}, x_{2,1}, \ldots, x_{w,1}, x_{1,2}, \ldots, x_{1,m}, \ldots, x_{w,m}]$. Hence, $f^{(s)}$ can be represented by a polynomial of the form

$$\hat{f}^{(s)}(x_{1,1}, \ldots, x_{w,m}) = \sum_{i=(i_{1,1}, \ldots, i_{w,m}) \in \mathbb{F}_2^{mw}} a_i \prod_{\substack{1 \le j \le w \\ 1 \le k \le m}} x_{j,k}^{i_{j,k}},$$

where $a_i \in \mathbb{Z}_2$, $x_{j,k}^0 = 1$ and $x_{j,k}^1 = x_{j,k}$. The algebraic degree of a Boolean function g is the number of variables in the longest term of $\hat{g}$.

Here, we will be interested in the case when $m \le 2$. For simplicity, we will use the variables $x_1, x_2, \ldots, x_w$ for the case of $m = 1$, and $x_1, x_2, \ldots, x_w, y_1, y_2, \ldots, y_w$

for the case of $m = 2$. We will denote by $\mathbf{x}$ and $\mathbf{y}$ the $w \times 1$ matrices $[x_i]_{w\times 1}$ and $[y_i]_{w\times 1}$ over $\mathbb{F}_2[x_1,\ldots,x_w,y_1,\ldots,y_w]$, respectively.

For better readability, we will also use the notations $\mathbf{M}(\mathbf{x})$ and $\mathbf{M}(\mathbf{x},\mathbf{y})$ for matrices over $\mathbb{F}_2[x_1,\ldots,x_w,y_1,\ldots,y_w]$ whose elements are polynomials in the variables $x_1,\ldots,x_w$ and $x_1,\ldots,x_w,y_1,\ldots,y_w$, respectively.

Recall that an $n \times n$ matrix $\mathbf{M}$ over a commutative ring is called nonsingular or invertible if there exists an $n \times n$ matrix $\mathbf{T}$ such that $\mathbf{MT} = \mathbf{TM} = \mathbf{I}_n$. Furthermore $\mathbf{M}$ is nonsingular if and only if its determinant is invertible. In the case of a square matrix $\mathbf{M}(\mathbf{x},\mathbf{y})$ over the ring $\mathbb{F}_2[x_1,\ldots,x_w,y_1,\ldots,y_w]$, this means that $\mathbf{M}(\mathbf{x},\mathbf{y})$ is nonsingular if and only if $\det(\mathbf{M}(\mathbf{x},\mathbf{y})) = 1$, if and only if $\det(\mathbf{M}(\mathbf{a},\mathbf{b})) = 1$ over $\mathbb{F}_2$ for every $\mathbf{a},\mathbf{b} \in \mathbb{F}_2^w$.

In the rest of this text, we will not distinguish between a Boolean function g and its polynomial ANF form $\hat{g}$, i.e., we will consider them equivalent, and use simply the notation g.

Let (Q,q) be a left quasigroup of order 2^w. We fix a bijection $\phi : Q \to \mathbb{F}_2^w$ and identify $u \in Q$ by the Boolean vector $\phi(u) = \mathbf{u}$. Now, the binary operation q on Q can be viewed as a mapping $q_\phi : \mathbb{F}_2^{2w} \to \mathbb{F}_2^w$ defined by:

$$q_\phi(\mathbf{u},\mathbf{v}) = \mathbf{z} \Longleftrightarrow q(u,v) = z.$$

Hence, without loss of generality, all left quasigroups of order 2^w can be viewed as mappings $q = (q^{(1)}, q^{(2)}, \ldots, q^{(w)}) : \mathbb{F}_2^{2w} \to \mathbb{F}_2^w$ represented in their ANF form over $\mathbb{F}_2[x_1,x_2,\ldots,x_w,y_1,y_2,\ldots,y_w]$.

We will call these quasigroups Left Multivariate Quasigroups (LMQ). If the algebraic degree of an LMQ is 2, we will call it Left Multivariate Quadratic Quasigroup (LMQQ). Note that this is in accordance with the naming convention from [9] where the notion of Multivariate Quadratic Quasigroups (MQQ) was introduced.

3. Construction of left multivariate quasigroups

In [19], the authors give necessary and sufficient conditions for a T-function (defined in [13]) to define a permutation or a quasigroup. This characterization provides a deterministic construction of multivariate quasigroups.

For left multivariate quasigroups it is possible to give a simpler form than the one in [19]. We will need the following straightforward result.

Theorem 1 ([19]). *A mapping $p = (p^{(1)}, p^{(2)}, \ldots, p^{(w)}) : \mathbb{F}_2^w \to \mathbb{F}_2^w$ such that for every $s = 1,\ldots,w$, the component $p^{(s)}$ is a Boolean polynomial of the form*

$$p^{(s)}(x_1,\ldots,x_w) = x_s + \sum_{j=(j_{s+1},\ldots,j_w)\in\mathbb{F}_2^{w-s}} \alpha_j^{(s)} x_{s+1}^{j_{s+1}} x_{s+2}^{j_{s+2}} \ldots x_w^{j_w},$$

defines a permutation on the set $\mathbb{F}_2^w$. □

It is an easy consequence that the following holds.

Theorem 2. *A mapping $q = (q^{(1)}, q^{(2)}, \dots, q^{(w)}) : \mathbb{F}_2^{2w} \to \mathbb{F}_2^w$ such that for every $s = 1, \dots, w$, the component $q^{(s)}$ is a Boolean polynomial of the form*

$$(2)\quad q^{(s)}(x_1, \dots, x_w, y_1, \dots, y_w) = y_s + \sum_{\substack{k=(k_1,..,k_w)\in\mathbb{F}_2^w \\ j=(j_{s+1},\dots,j_w)\in\mathbb{F}_2^{w-s}}} \alpha_{k,j}^{(s)} x_1^{k_1} x_2^{k_2} \dots x_w^{k_w} y_{s+1}^{j_{s+1}} y_{s+2}^{j_{s+2}} \dots y_w^{j_w},$$

defines an LMQ of order 2^w.

Proof: Clearly, for any $(a_1, \dots, a_w) \in \mathbb{F}_2^w$, $q^{(s)}(a_1, \dots, a_w, y_1, \dots, y_w)$ is a permutation by Theorem 1, hence (2) defines an LMQ. $\square$

The form given in Theorem 2 can be rewritten in an equivalent matrix form.

Theorem 3. *Let $\mathbf{A}(\mathbf{x}) = [a_i(\mathbf{x})]_{w\times 1}$ and $\mathbf{B}(\mathbf{x}, \mathbf{y}) = [b_{ij}(\mathbf{x}, \mathbf{y})]_{w\times w}$ be matrices of Boolean polynomials in the variables $x_1, \dots, x_w, y_1, \dots, y_w$, such that $a_i(\mathbf{x})$ depends only on the variables $x_1, \dots, x_w$, for all i, $1 \le i \le w$, and $\mathbf{B}(\mathbf{x}, \mathbf{y})$ is an upper triangular matrix with 1s on the diagonal, and $b_{ij}(\mathbf{x}, \mathbf{y})$ depends only on the variables $x_1, \dots, x_w$, $y_{j+1}, \dots, y_w$, for all i, j, $1 \le i < j \le w$.*

Then the mapping

$$(3)\qquad q(\mathbf{x}, \mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x}, \mathbf{y}) \cdot \mathbf{y}$$

defines a left multivariate quasigroup of order 2^w.

Proof: We show that the forms (2) and (3) are equivalent.

Let an LMQ q be given by the form (2). Then the component $q^{(s)}$ is of the form

$$q^{(s)}(x_1, \dots, x_w, y_1, \dots, y_w) = \left(\sum_{k=(k_1,..,k_w)\in\mathbb{F}_2^w} \alpha_{k,0}^{(s)} x_1^{k_1} x_2^{k_2} \dots x_w^{k_w} \right) + y_s +$$

$$+ \left(\sum_{\substack{k=(k_1,..,k_w)\in\mathbb{F}_2^w \\ j=(1,j_{s+2},\dots,j_w)\in\mathbb{F}_2^{w-s}}} \alpha_{k,j}^{(s)} x_1^{k_1} x_2^{k_2} \dots x_w^{k_w} y_{s+2}^{j_{s+2}} \dots y_w^{j_w} \right) \cdot y_{s+1} + \dots +$$

$$+ \left(\sum_{\substack{k=(k_1,..,k_w)\in\mathbb{F}_2^w \\ j=(0,\dots,0,1,j_w)\in\mathbb{F}_2^{w-s}}} \alpha_{k,j}^{(s)} x_1^{k_1} \dots x_w^{k_w} y_w^{j_w} \right) \cdot y_{w-1} +$$

$$+ \left(\sum_{\substack{k=(k_1,..,k_w)\in\mathbb{F}_2^w \\ j=(0,\dots,0,1)\in\mathbb{F}_2^{w-s}}} \alpha_{k,j}^{(s)} x_1^{k_1} \dots x_w^{k_w} \right) \cdot y_w.$$

As this is true for every component $q^{(s)}$ of q, q can be rewritten in the matrix form (3). □

Form (3) allows creation of left quasigroups of any order and degree. If we take the Boolean polynomials in $\mathbf{A}(\mathbf{x})$ to be of degree d, and the Boolean polynomials in $\mathbf{B}(\mathbf{x},\mathbf{y})$ to be of degree $d-1$, then the left quasigroup q will have degree d. Using isotopy we can create new left quasigroups, and if the isotopy is (α,β,γ), such that $\alpha(\mathbf{x}) = \mathbf{D_1}\mathbf{x} + \mathbf{c_1}, \beta(\mathbf{x}) = \mathbf{D_2}\mathbf{x} + \mathbf{c_2}, \gamma^{-1}(\mathbf{x}) = \mathbf{D_3}\mathbf{x} + \mathbf{c_3}$, where $\mathbf{D_1},\mathbf{D_2},\mathbf{D_3}$ are nonsingular $w \times w$ matrices over $\mathbb{F}_2$ and $\mathbf{c_1},\mathbf{c_2},\mathbf{c_3} \in \mathbb{F}_2^w$, the degree is preserved, i.e. the newly obtained left quasigroups are again of degree d. Throughout the rest of the text we will call such isotopies linear.

Finding the parastrophe $q_\backslash$ of q for a given LMQ can in general be a task of great space and time complexity. That is also true for the special class of left quasigroups defined in Theorem 3. However, if (3) is of the form

$$q(\mathbf{x},\mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \cdot \mathbf{y}, \tag{4}$$

i.e., $\mathbf{B}(\mathbf{x})$ depends only on the variables $x_1,\dots,x_w$, then the left parastrophe $q_\backslash$, can be easily found, using one of the identities (1), to be:

$$q_\backslash(\mathbf{x},\mathbf{y}) = \mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x}) + \mathbf{B}^{-1}(\mathbf{x}) \cdot \mathbf{y}. \tag{5}$$

Even more, we have the following.

Proposition 3. *Let $\mathcal{TLQ}_{2^w}$ be the set of all left quasigroups of order 2^w of the form (4). Then $(\mathcal{TLQ}_{2^w},\circ)$ is a subgroup of $(\mathcal{Q}_{2^w},\circ)$.*

Proof: Let $q_1,q_2 \in \mathcal{TLQ}_{2^w}$. Then $q_1(\mathbf{x},\mathbf{y}) = \mathbf{A}_1(\mathbf{x}) + \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{y}$ and $q_2(\mathbf{x},\mathbf{y}) = \mathbf{A}_2(\mathbf{x}) + \mathbf{B}_2(\mathbf{x}) \cdot \mathbf{y}$, for some matrices $\mathbf{A}_1(\mathbf{x}),\mathbf{A}_2(\mathbf{x})$ of Boolean polynomials, and some upper triangular matrices $\mathbf{B}_1(\mathbf{x}),\mathbf{B}_2(\mathbf{x})$ of Boolean polynomials, with 1s on the diagonal. Then,

$$\begin{aligned} q_1 \circ q_2(\mathbf{x},\mathbf{y}) &= q_1(\mathbf{x},q_2(\mathbf{x},\mathbf{y})) = \mathbf{A}_1(\mathbf{x}) + \mathbf{B}_1(\mathbf{x}) \cdot (\mathbf{A}_2(\mathbf{x}) + \mathbf{B}_2(\mathbf{x}) \cdot \mathbf{y}) \\ &= (\mathbf{A}_1(\mathbf{x}) + \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{A}_2(\mathbf{x})) + \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2(\mathbf{x}) \cdot \mathbf{y}. \end{aligned}$$

Since $\mathbf{B}_1(\mathbf{x})$ and $\mathbf{B}_2(\mathbf{x})$ are upper triangular, their product $\mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2(\mathbf{x})$ is again upper triangular, and has 1s on the diagonal. So $q_1 \circ q_2 \in \mathcal{TLQ}_{2^w}$. The identity element of $(\mathcal{Q}_{2^w},\circ)$ is $e(\mathbf{x},\mathbf{y}) = \mathbf{y}$, and it is clearly in $\mathcal{TLQ}_{2^w}$ as well.

The inverse of a quasigroup q is its left parastrophe $q_\backslash$, and from (5) it is clear that $q_\backslash \in \mathcal{TLQ}_{2^w}$. Hence, the claim follows. □

Example 1. We give an example of a construction of an LMQQ of order 2^4 obtained by applying isotopic transformation to an LMQQ from $\mathcal{TLQ}_{2^4}$.

We first construct q in the form (4).

Let $\mathbf{A}(\mathbf{x})$ be a 4×1 matrix of quadratic Boolean polynomials in the variables x_1,x_2,x_3,x_4, given by:

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} x_1 + x_3 + x_1x_3 + x_2x_3 + x_1x_4 + x_2x_4 + x_3x_4 \\ 1 + x_1 + x_2 + x_3 + x_1x_3 + x_2x_3 + x_1x_4 \\ x_2 + x_1x_2 + x_3 + x_1x_3 + x_1x_4 + x_3x_4 \\ x_1x_2 + x_1x_3 + x_2x_3 + x_4 + x_1x_4 + x_2x_4 + x_3x_4 \end{bmatrix}.$$

Let $\mathbf{B}(\mathbf{x})$ be a 4×4 upper triangular matrix of linear Boolean polynomials in the variables x_1, x_2, x_3, x_4, with 1s on the diagonal given by:

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} 1 & x_2 + x_3 & 1 + x_2 & 1 + x_4 \\ 0 & 1 & 1 + x_4 & 1 + x_1 + x_2 + x_3 \\ 0 & 0 & 1 & x_2 + x_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then $q(\mathbf{x}, \mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \cdot \mathbf{y}$ is

$$q(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} x_1 + x_3 + x_1x_3 + x_2x_3 + x_1x_4 + x_2x_4 + x_3x_4 + y_1 + \\ + x_2y_2 + x_3y_2 + y_3 + x_2y_3 + y_4 + x_4y_4 \\ 1 + x_1 + x_2 + x_3 + x_1x_3 + x_2x_3 + x_1x_4 + y_2 + y_3 + \\ + x_4y_3 + y_4 + x_1y_4 + x_2y_4 + x_3y_4 \\ x_2 + x_1x_2 + x_3 + x_1x_3 + x_1x_4 + x_3x_4 + y_3 + x_2y_4 + x_3y_4 \\ x_1x_2 + x_1x_3 + x_2x_3 + x_4 + x_1x_4 + x_2x_4 + x_3x_4 + y_4 \end{bmatrix}.$$

The matrix $\mathbf{B}^{-1}(\mathbf{x})$ is given by

$$\mathbf{B}^{-1}(\mathbf{x}) = \begin{bmatrix} 1 & x_2 + x_3 & 1 + x_3 + x_2x_4 + x_3x_4 & \begin{matrix} 1 + x_2 + x_1x_2 + x_1x_3 + \\ + x_2x_3 + x_4 + x_2x_4 + x_3x_4 \end{matrix} \\ 0 & 1 & 1 + x_4 & 1 + x_1 + x_2x_4 + x_3x_4 \\ 0 & 0 & 1 & x_2 + x_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and the parastrophe $q_{\backslash}(\mathbf{x}, \mathbf{y}) = \mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x}) + \mathbf{B}^{-1}(\mathbf{x}) \cdot \mathbf{y}$ by:

$$q_{\backslash}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} x_1 + x_2 + x_1x_2 + x_3 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_3x_4 + x_1x_2x_3x_4 + \\ + y_1 + x_2y_2 + x_3y_2 + y_3 + x_3y_3 + x_2x_4y_3 + x_3x_4y_3 + y_4 + x_2y_4 + \\ + x_1x_2y_4 + x_1x_3y_4 + x_2x_3y_4 + x_4y_4 + x_2x_4y_4 + x_3x_4y_4 \\ 1 + x_1 + x_1x_2 + x_1x_2x_3 + x_4 + y_2 + y_3 + x_4y_3 + \\ + y_4 + x_1y_4 + x_2x_4y_4 + x_3x_4y_4 \\ x_2 + x_3 + x_1x_4 + x_1x_2x_4 + x_3x_4 + x_1x_3x_4 + y_3 + x_2y_4 + x_3y_4 \\ x_1x_2 + x_1x_3 + x_2x_3 + x_4 + x_1x_4 + x_2x_4 + x_3x_4 + y_4 \end{bmatrix}.$$

Next we apply to q a linear isotopy defined by the nonsingular matrices:

$$\mathbf{D_1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{D_2} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \mathbf{D_3} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

and by the vectors: $\mathbf{c_1} = (1, 1, 0, 1)$, $\mathbf{c_2} = (0, 1, 1, 0)$, $\mathbf{c_3} = (0, 0, 1, 1)$.

We obtain the left quasigroup
$q'(\mathbf{x},\mathbf{y}) = \mathbf{D_3}(\mathbf{A}(\mathbf{D_1}\mathbf{x}+\mathbf{c_1}) + \mathbf{B}(\mathbf{D_1}\mathbf{x}+\mathbf{c_1})\cdot(\mathbf{D_2}\mathbf{y}+\mathbf{c_2})) + \mathbf{c_3}$ given by

$$q'(\mathbf{x},\mathbf{y}) = \begin{bmatrix} 1+x_1+x_1x_2+x_3+x_1x_3+x_2x_4+x_2y_1+x_3y_1+x_4y_1+y_2+ \\ +x_3y_2+y_3+y_4+x_2y_4+x_3y_4 \\ 1+x_1+x_1x_2+y_2+x_2y_2+x_3y_2+x_4y_2+x_2y_4+x_3y_4+x_4y_4 \\ x_2+x_1x_2+x_1x_4+y_1+x_2y_1+x_3y_1+x_4y_1+y_2+x_1y_2+ \\ +x_2y_2+y_3+x_3y_4+x_4y_4 \\ 1+x_1x_2+x_3+x_1x_3+x_2x_3+x_4+x_2x_4+y_1+y_2+x_1y_2+ \\ +x_2y_2+x_3y_2+x_2y_4+x_4y_4 \end{bmatrix}.$$

Since the left quasigroups in the class $\mathcal{TLQ}_{2^w}$ are quite easy to construct, they will be our main focus in the next section. In the rest of this section we will point out the relationship with the bigger class of left quasigroups affine in the second argument $\mathbf{y}$.

Definition 4. A left multivariate quasigroup $(\mathbb{F}_2^w, q)$ is said to be left affine, if for every $\mathbf{a}\in\mathbb{F}_2^w$, $L_{q,\mathbf{a}}(\mathbf{y})$ is an affine mapping.

We denote the set of all LMQs of order 2^w that are left affine by $\mathcal{LLQ}_{2^w}$.

Proposition 4. *A mapping $q : \mathbb{F}_2^{2w} \to \mathbb{F}_2^w$ is in $\mathcal{LLQ}_{2^w}$ if and only if it has the form*

$$q(\mathbf{x},\mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{B}_0(\mathbf{x})\cdot\mathbf{y}, \tag{6}$$

where $\mathbf{A}(\mathbf{x}) = [a_i(\mathbf{x})]_{w\times 1}$ is a $w\times 1$ matrix of Boolean polynomials in the variables $x_1,\dots,x_w$, and $\mathbf{B}_0(\mathbf{x}) = [b_{ij}(\mathbf{x})]_{w\times w}$ is a $w\times w$ nonsingular matrix of Boolean polynomials in the variables $x_1,\dots,x_w$.

Proof: If q has the form (6) then clearly it is a left quasigroup, and it is in $\mathcal{LLQ}_{2^w}$. Conversely, let $q\in\mathcal{LLQ}_{2^w}$. Then, using a vector notation, it can be represented in the general form: $q(\mathbf{x},\mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{A}_1(\mathbf{x},\mathbf{y}) + \mathbf{A}_2(\mathbf{y})$ where the s-th component is

$$q^{(s)}(\mathbf{x},\mathbf{y}) = \mathbf{A}^{(s)}(\mathbf{x}) + \mathbf{A}_1^{(s)}(\mathbf{x},\mathbf{y}) + \mathbf{A}_2^{(s)}(\mathbf{y}) = \sum_{k=(k_1,..,k_w)\in\mathbb{F}_2^w} \alpha_k^{(s)} x_1^{k_1}x_2^{k_2}\dots x_w^{k_w} +$$

$$+ \sum_{\substack{k=(k_1,..,k_w)\in\mathbb{F}_2^w \\ j=(j_1,\dots,j_w)\in\mathbb{F}_2^w \\ k,j\neq\mathbf{0}}} \beta_{k,j}^{(s)} x_1^{k_1}\dots x_w^{k_w} y_1^{j_1}\dots y_w^{j_w} + \sum_{\substack{k=(k_1,..,k_w)\in\mathbb{F}_2^w \\ k\neq\mathbf{0}}} \gamma_k^{(s)} y_1^{k_1}y_2^{k_2}\dots y_w^{k_w} \tag{7}$$

Now, for every $\mathbf{a}\in\mathbb{F}_2^w$,

$$L_{q,\mathbf{a}}(\mathbf{y}) = \mathbf{A}(\mathbf{a}) + \mathbf{A}_1(\mathbf{a},\mathbf{y}) + \mathbf{A}_2(\mathbf{y})$$

is an affine mapping, so

$$\mathbf{A}_1(\mathbf{a},\mathbf{y}) + \mathbf{A}_2(\mathbf{y}) = \mathbf{B}_0(\mathbf{a})\cdot\mathbf{y}$$

has to be a linear mapping, i.e. $\mathbf{B}_0(\mathbf{a})$ is a nonsingular $w \times w$ matrix for every $\mathbf{a} \in \mathbb{F}_2^w$. Hence, q has the form (6). □

Similarly as for Proposition 3 it is straightforward that the following is true.

Proposition 5. *$(\mathcal{LLQ}_{2^w}, \circ)$ is a subgroup of $(\mathcal{Q}_{2^w}, \circ)$, and also $(\mathcal{TLQ}_{2^w}, \circ)$ is a subgroup of $(\mathcal{LLQ}_{2^w}, \circ)$.* □

Proposition 6. *Let $q \in \mathcal{LLQ}_{2^w}$ be given in the form (6). If $\mathbf{B}_0(\mathbf{x})$ can be decomposed as $\mathbf{B}_0(\mathbf{x}) = \mathbf{D_1} \cdot \mathbf{B}(\mathbf{x}) \cdot \mathbf{D_2}$, where $\mathbf{D_1}, \mathbf{D_2}$ are $w \times w$ nonsingular Boolean matrices, and $\mathbf{B}(\mathbf{x})$ is an upper triangular matrix of Boolean polynomials in $x_1, \dots, x_w$, with 1s on the diagonal, then q can be constructed using a linear isotopy from a quasigroup in $\mathcal{TLQ}_{2^w}$ with the same degree.*

Proof: Let q be as defined. Then

$$\begin{aligned} q(\mathbf{x},\mathbf{y}) &= \mathbf{A}(\mathbf{x}) + \mathbf{B}_0(\mathbf{x}) \cdot \mathbf{y} = \mathbf{A}(\mathbf{x}) + \mathbf{D_1} \cdot \mathbf{B}(\mathbf{x}) \cdot \mathbf{D_2} \cdot \mathbf{y} \\ &= \mathbf{D_1} \cdot (\mathbf{D_1}^{-1} \cdot \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \cdot (\mathbf{D_2} \cdot \mathbf{y})) \\ &= \mathbf{D_1} \cdot (\mathbf{A}'(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \cdot (\mathbf{D_2} \cdot \mathbf{y})). \end{aligned}$$

Let $q'(\mathbf{x},\mathbf{y}) = \mathbf{A}'(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \cdot \mathbf{y}$. Clearly, $q' \in \mathcal{TLQ}_{2^w}$. Now, $q(\mathbf{x},\mathbf{y}) = \mathbf{D_1} \cdot q'(\mathbf{x}, \mathbf{D_2} \cdot \mathbf{y})$, i.e., q can be obtained from q' using the isotopy $(I, \mathbf{D_2}, \mathbf{D_1}^{-1})$. □

Proposition 7. *Let q be a quadratic loop of order 2^w. Then $q \in \mathcal{LLQ}_{2^w}$.*

Proof: First, let q be a quadratic loop of order 2^w with a unit element $\mathbf{0} = (0, 0, \dots, 0) \in \mathbb{F}_2^w$. Then from [6],

$$q(\mathbf{x},\mathbf{y}) = \mathbf{x} + \beta(\mathbf{x},\mathbf{y}) + \mathbf{y}$$

where β is a bilinear Boolean map. Clearly, q is left affine i.e., $q \in \mathcal{LLQ}_{2^w}$.

Now, let q be an arbitrary quadratic loop of order 2^w. Then, q is linearly isomorphic to a loop with unit element $\mathbf{0}$. The linear isomorphism does not change the degree of $\mathbf{y}$, hence, again $q \in \mathcal{LLQ}_{2^w}$. □

4. LMQQs whose left parastrophe is also quadratic

In this section we will focus on the left quasigroups from $\mathcal{TLQ}_{2^w}$ that have an algebraic degree 2, i.e. on LMQQs that can be represented in the form (4). Then the left parastrophe of the LMQQ q is given by (5).

The possibility of expressing $q_\backslash$ using a short formula is a neat property of these LMQQs. But this does not imply that it is always efficient to use $q_\backslash$ in such a form. In general, although q is quadratic, $q_\backslash$ can be of any degree d, $2 \le d \le 2w$ (see Example 1). Hence for a random q, the average number of terms in $q_\backslash$ is exponential in the number of variables.

Here, we will focus on finding a class of such LMQQs in the group $\mathcal{TLQ}_{2^w}$, with the additional property of efficient algorithmic construction.

From (5) it is straightforward that:

Proposition 8. *An LMQQ that can be written in the form* (4) *has a left parastrophe that is also an LMQQ (i.e., it is also of degree* 2) *if and only if* $\mathbf{B}^{-1}(\mathbf{x})$ *is a* $w \times w$ *upper triangular matrix of linear Boolean polynomials, and* $\mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x})$ *is a* $w \times 1$ *matrix of Boolean polynomials of degree* 2. □

Next we want to find under what conditions the elements of $\mathbf{B}^{-1}(\mathbf{x})$ are linear polynomials.

We introduce the following notations.

Let $\mathbf{B}'(\mathbf{x})$ be an upper triangular matrix of linear Boolean polynomials in the variables $x_1, x_2, \dots, x_w$, with 1s on the diagonal. We denote the elements of the matrices $\mathbf{B}(\mathbf{x})$ and $\mathbf{B}'(\mathbf{x})$ by $b_{ij}(\mathbf{x})$ and $b'_{ij}(\mathbf{x})$, respectively, and represent them in the following form:

$$b_{ij}(\mathbf{x}) = \mathbf{x}^\top \cdot \mathbf{b}_{ij} + \mathrm{b}_{ij} \quad \text{and} \quad b'_{ij}(\mathbf{x}) = \mathbf{x}^\top \cdot \mathbf{b}'_{ij} + \mathrm{b}'_{ij}, \tag{8}$$

where $\mathbf{b}_{ij}, \mathbf{b}'_{ij} \in \mathbb{F}_2^w$, and $\mathrm{b}_{ij}, \mathrm{b}'_{ij} \in \mathbb{F}_2$. (Note that $b_{ii}(\mathbf{x}) = b_{ii} = 1$.)

In other words, we represent the matrices $\mathbf{B}(\mathbf{x})$ and $\mathbf{B}'(\mathbf{x})$ as sums of upper triangular matrices

$$\mathbf{B}(\mathbf{x}) = \mathbf{B}_1(\mathbf{x}) + \mathbf{B}_2, \quad \text{and} \quad \mathbf{B}'(\mathbf{x}) = \mathbf{B}'_1(\mathbf{x}) + \mathbf{B}'_2, \tag{9}$$

where the Boolean polynomials $\mathbf{x}^\top \cdot \mathbf{b}_{ij}$ and $\mathbf{x}^\top \cdot \mathbf{b}'_{ij}$ are the elements of $\mathbf{B}_1(\mathbf{x})$ and $\mathbf{B}'_1(\mathbf{x})$, respectively, and $\mathrm{b}_{ij} \in \mathbb{F}_2$ and $\mathrm{b}'_{ij} \in \mathbb{F}_2$ are the elements of $\mathbf{B}_2$ and $\mathbf{B}'_2$, respectively.

It is straightforward to verify the following:

Proposition 9. *Let the matrices* $\mathbf{B}(\mathbf{x})$ *and* $\mathbf{B}'(\mathbf{x})$, *given in the form* (9), *satisfy the conditions:*

1. $\mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}'_1(\mathbf{x}) = \mathbf{0}$,
2. $\mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}'_2 + \mathbf{B}_2 \cdot \mathbf{B}'_1(\mathbf{x}) = \mathbf{0}$,
3. $\mathbf{B}'_2 = \mathbf{B}_2^{-1}$.

Then, $\mathbf{B}'(\mathbf{x}) = \mathbf{B}^{-1}(\mathbf{x})$. □

The conditions 1., 2., and 3. from Proposition 9 can be rewritten in a simpler equivalent form given in the next proposition.

Proposition 10. *The matrix* $\mathbf{B}(\mathbf{x})$ *given in the form* (9), *satisfies the condition:*

$$\mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) = \mathbf{0} \tag{10}$$

if and only if there exists a matrix $\mathbf{B}'(\mathbf{x}) = \mathbf{B}'_1(\mathbf{x}) + \mathbf{B}'_2$ *of the form* (9) *such that the conditions* 1., 2., *and* 3., *are satisfied for* $\mathbf{B}(\mathbf{x})$ *and* $\mathbf{B}'(\mathbf{x})$.

Furthermore, if $\mathbf{B}(\mathbf{x})$ *satisfies* (10), *then* $\mathbf{B}^{-1}(\mathbf{x}) = \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2^{-1} + \mathbf{B}_2^{-1}$.

Proof: Let the matrix $\mathbf{B}(\mathbf{x})$ satisfy (10). Let $\mathbf{B}'_1(\mathbf{x}) = \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2^{-1}$ and $\mathbf{B}'_2 = \mathbf{B}_2^{-1}$. It is easy to verify that the conditions 1., 2., and 3., from Proposition 9 hold for the matrices $\mathbf{B}(\mathbf{x})$ and $\mathbf{B}'(\mathbf{x}) = \mathbf{B}'_1(\mathbf{x}) + \mathbf{B}'_2$.

Conversely, let there exist a matrix $\mathbf{B}'(\mathbf{x})$ such that 1., 2., and 3. hold. Then from Proposition 9, $\mathbf{B}'(\mathbf{x}) = \mathbf{B}^{-1}(\mathbf{x})$ and thus $\mathbf{B}_1'(\mathbf{x}) \cdot \mathbf{B}_1(\mathbf{x}) + \mathbf{B}_1'(\mathbf{x}) \cdot \mathbf{B}_2 + \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) = \mathbf{0}$. Now

$$\mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) = \mathbf{B}_1(\mathbf{x})(\mathbf{B}_1'(\mathbf{x}) \cdot \mathbf{B}_1(\mathbf{x}) + \mathbf{B}_1'(\mathbf{x}) \cdot \mathbf{B}_2) = \mathbf{0},$$

i.e., (10) holds.

Now it is clear that if $\mathbf{B}(\mathbf{x})$ satisfies (10), then $\mathbf{B}^{-1}(\mathbf{x}) = \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2^{-1} + \mathbf{B}_2^{-1}$. □

The next proposition provides an equivalent explicit form of (10).

Proposition 11. *The matrix $\mathbf{B}(\mathbf{x})$ given in the form* (9)*, satisfies condition* (10) *if and only if for every i, j, $j - i \geq 2$,*

$$\sum_{i=r_0<\cdots<r_m=j} \mathbf{b}_{r_0 r_1} \mathrm{b}_{r_1 r_2} \cdots \mathrm{b}_{r_{m-2} r_{m-1}} \mathbf{b}^{\top}_{r_{m-1} r_m} = \mathbf{0}. \tag{11}$$

Furthermore, if (11) *holds, then the elements $b'_{ij}(\mathbf{x})$ of $\mathbf{B}^{-1}(\mathbf{x})$ are linear Boolean polynomials and $b'_{ij}(\mathbf{x}) = \mathbf{x}^{\top} \cdot \mathbf{b}'_{ij} + \mathrm{b}'_{ij}$, where:*

$$\mathbf{b}'_{ij} = \sum_{\substack{i = r_0 < \cdots < r_m = j \\ t \in \{0, \ldots, m-1\}}} \mathrm{b}_{r_0, r_1} \cdots \mathbf{b}_{r_t r_{t+1}} \cdots \mathrm{b}_{r_{m-1}, r_m}, \tag{12}$$

$$\mathrm{b}'_{ij} = \sum_{i=r_0<\cdots<r_m=j} \mathrm{b}_{r_0 r_1} \mathrm{b}_{r_1 r_2} \cdots \mathrm{b}_{r_{m-1}, r_m}. \tag{13}$$

Proof: We will expand the condition (10). First we need an explicit form for $\mathbf{B}_2^{-1}$, i.e. a formula for the elements of the inverse of an upper triangular Boolean matrix with 1s on the diagonal.

From elementary linear algebra, $\mathbf{B}_2^{-1} = [\det(B_{ij})]_{w \times w}$, where $B_{ij} = [\beta^{ij}_{sr}]_{(w-1)\times(w-1)}$ is obtained from $\mathbf{B}_2$ by removing its j-th row and i-th column. Clearly, for $i > j$ $\det(B_{ij}) = 0$, and $\det(B_{ii}) = 1$.

For $i < j$,

$$\beta^{ij}_{sr} = \begin{cases} \mathrm{b}_{s,r}, & \text{for } s < j, r < i, \\ \mathrm{b}_{s+1,r}, & \text{for } s \geq j, r < i, \\ \mathrm{b}_{s,r+1}, & \text{for } s < j, r \geq i, \\ \mathrm{b}_{s+1,r+1}, & \text{for } s \geq j, r \geq i. \end{cases} \tag{14}$$

In general, $\det(B_{ij}) = \sum_{\sigma \in \mathcal{S}_{w-1}} \beta^{ij}_{1,\sigma(1)} \beta^{ij}_{2,\sigma(2)} \cdots \beta^{ij}_{w-1,\sigma(w-1)}$. From (14), the terms in the sum are 0, except for permutations $\sigma \in \mathcal{S}_{w-1}$ such that $s + 1 \leq \sigma(s)$, for every s, $1 \leq s \leq w-1$. The permutations that satisfy this condition are permutations with cyclic decomposition to cycles of the form $(s, s-1, \ldots, s-t)$.

Again, from (14),

$$\beta^{ij}_{r+1,r} = \begin{cases} \mathrm{b}_{r+1,r+1}, & \text{for } \ i \leq r < j-1, \\ 0, & \text{otherwise.} \end{cases}$$

Hence, $\det(B_{ij}) = \sum_{i=r_0<\cdots<r_m=j} \mathrm{b}_{r_0 r_1} \mathrm{b}_{r_1 r_2} \cdots \mathrm{b}_{r_{m-1},r_m}$.

Now (10) is equivalent to

$$\sum_{t=i+1}^{j} \Big(\sum_{k=i+1}^{t} \mathbf{x}^\top \mathbf{b}_{ik} \det(B_{kt}) \Big) \mathbf{b}_{tj} \mathbf{x} = \mathbf{0}, \quad \text{for every} \quad i < j,$$

which in turn is equivalent to

$$\sum_{t=i+1}^{j} \sum_{k=i+1}^{t} \mathbf{b}_{ik} \det(B_{kt}) \mathbf{b}_{tj} = \mathbf{0}, \quad \text{for every} \quad i < j.$$

If we expand the last expression using the above formula for $\det(B_{kt})$, we obtain (11).

The rest of the proposition follows directly from Proposition 9 if we apply the formula for $\mathbf{B}_2^{-1}$. □

Having found sufficient conditions for the matrix $\mathbf{B}(\mathbf{x})$ to have an inverse that is a matrix of linear polynomials over $\mathbb{F}_2$, we can state the following.

Proposition 12. *Let the matrix $\mathbf{B}(\mathbf{x})$ satisfy condition (10). Then an LMQQ $q(\mathbf{x},\mathbf{y})$ of the form (4) has a left parastrophe of degree 2 if and only if there exists a $w \times 1$ matrix of homogeneous quadratic Boolean polynomials $\mathbf{A}_2'(\mathbf{x})$ such that $\mathbf{B}_1(\mathbf{x})\mathbf{A}_2'(\mathbf{x})$ is a $w \times 1$ matrix of homogeneous quadratic Boolean polynomials.*

Proof: Let $\mathbf{A}_2'(\mathbf{x})$ satisfy the given conditions. Let $\mathbf{A}_1'(\mathbf{x})$ be a $w \times 1$ matrix of linear Boolean polynomials. Put $\mathbf{A}(\mathbf{x}) = \mathbf{B}_2(\mathbf{A}_1'(\mathbf{x}) + \mathbf{A}_2'(\mathbf{x}))$. From Proposition 10,

$$\begin{aligned} \mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x}) &= (\mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{B}_2^{-1} + \mathbf{B}_2^{-1}) \cdot \mathbf{B}_2 \cdot (\mathbf{A}_1'(\mathbf{x}) + \mathbf{A}_2'(\mathbf{x})) \\ &= \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{A}_1'(\mathbf{x}) + \mathbf{A}_1'(\mathbf{x}) + \mathbf{A}_2'(\mathbf{x}) + \mathbf{B}_2^{-1} \cdot \mathbf{B}_1(\mathbf{x}) \cdot \mathbf{A}_2'(\mathbf{x}), \end{aligned}$$

which is a $w \times 1$ matrix of homogeneous quadratic Boolean polynomials. From Proposition 8, the left parastrophe of q is of degree 2.

Conversely, let $q(\mathbf{x},\mathbf{y})$ have a quadratic left parastrophe. Then from Proposition 8, $\mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x})$ is quadratic. Represent $\mathbf{A}(\mathbf{x})$ as $\mathbf{A}(\mathbf{x}) = \mathbf{A}_1(\mathbf{x}) + \mathbf{A}_2(\mathbf{x})$ where $\mathbf{A}_1(\mathbf{x})$ is $w \times 1$ matrix of linear Boolean polynomials and $\mathbf{A}_2(\mathbf{x})$ is $w \times 1$ matrix of homogeneous quadratic Boolean polynomials. Then, it is not hard to see that the matrix $\mathbf{A}_2'(\mathbf{x}) = \mathbf{B}_2^{-1}\mathbf{A}_2(\mathbf{x})$ satisfies the conditions. □

4.1 Algorithms for construction of the matrices $\mathbf{B}(\mathbf{x})$ and $\mathbf{A}(\mathbf{x})$. Next, we give a procedure for construction of a matrix $\mathbf{B}(\mathbf{x})$ that satisfies the given requirements.

In fact, we will first describe an algorithm that finds all the possible matrices $\mathbf{B}(\mathbf{x})$ that satisfy the constraints (11), for every i, j, where $j - i \geq 2$. The algorithm is in essence a search algorithm in a tree that uses depth-first search and backtracking techniques and finds all the possible solutions of the system of equations (11) for all $j - i \geq 2$, with unknowns $\mathbf{b}_{s,t} \in \mathbb{F}_2^w$ and $\mathrm{b}_{s,t} \in \mathbb{F}_2$.

Then we modify this algorithm by randomizing the value (successor) selection heuristics, to obtain a new but equivalent algorithm. However, the introduction of the heuristics enables the algorithm to be adapted to find a single matrix $\mathbf{B}(\mathbf{x})$ that satisfies the conditions, and is randomly drawn from the set of all possible matrices that satisfy the conditions.

First, we create the tree using the next procedure.

TreeSetup:

(1) We define an ordering "$\prec$"on the set of indices $\mathcal{I} = \{(i,j)|\ 2 \leq i+1 < j \leq w\}$ that correspond to the appropriate indices of the matrix $\mathbf{B}(\mathbf{x})$ by:

$$(i,j) \prec (i',j') \text{ if } j - i < j' - i', \ \text{ or if } \ j - i = j' - i' \ \text{ and } \ i < i'.$$

It is not hard to see that "$\prec$" is a total strict ordering.

With every index $(i,j) \in \mathcal{I}$ we associate the equation (11).

(2) We define a rooted tree of depth $|\mathcal{I}| = \frac{(w-2)(w-1)}{2}$ by associating the indices $(i,j) \in \mathcal{I}$ in ascending order to each level of the tree, starting from the level at depth 0, i.e. starting from the root node. Note that we do not associate indices to the last level, i.e. to the leafs. We label each level with the associated index.

The successors of each node are determined by the new unknowns appearing in (11) associated to the current level. At this point, we are not interested in the solutions of the associated equations, but rather in the new unknowns appearing in the equations. All the possible assignments for the new unknowns define a successor for the node. In more details, the successors are defined in the following way:

(i) The associated equation to level $(1,3)$ (the root of the tree) is $\mathbf{b}_{1,2}\mathbf{b}_{2,3}^\top = \mathbf{0}$. We assign each possible value of $(\mathbf{b}_{1,2}, \mathbf{b}_{2,3}) \in \mathbb{F}_2^{2w}$ to a different successor of the root node. Thus, the root node has 2^{2w} successors. We order the successors lexicographically.

(ii) The associated equation to level $(2,4)$ is $\mathbf{b}_{2,3}\mathbf{b}_{3,4}^\top = \mathbf{0}$. The new unknown appearing in the equation is $\mathbf{b}_{3,4}$. Hence, every possible value of $\mathbf{b}_{3,4} \in \mathbb{F}_2^w$ is assigned to 2^w different, lexicographically ordered successors of each of the nodes in the current level.

(iii) In a similar manner, for each of the nodes in the levels $(3,5), \ldots, (w-2, w)$ we define 2^w different, lexicographically ordered successors.

(iv) For level $(1,4)$, the associated equation is

$$\mathbf{b}_{1,2}\mathbf{b}_{2,4}^{\top}+\mathbf{b}_{1,3}\mathbf{b}_{3,4}^{\top}+\mathbf{b}_{1,2}\mathrm{b}_{2,3}\mathbf{b}_{3,4}^{\top}=\mathbf{0}.$$

The new unknowns appearing in the equation are $\mathbf{b}_{1,3},\mathbf{b}_{2,4}$ and $\mathrm{b}_{2,3}$. For each of the nodes in the level, we define 2^{2w+1} successors (for each possible value of $(\mathbf{b}_{1,3},\mathbf{b}_{2,4},\mathrm{b}_{2,3})\in\mathbb{F}_2^{2w+1}$) to which we assign the elements of $\mathbb{F}_2^{2w+1}$ in a lexicographic order.

(v) For each of the levels $(i,i+3), i\in\{2,\dots,w-3\}$, the associated equation is

$$\mathbf{b}_{i,i+1}\mathbf{b}_{i+1,i+3}^{\top}+\mathbf{b}_{i,i+2}\mathbf{b}_{i+2,i+3}^{\top}+\mathbf{b}_{i,i+1}\mathrm{b}_{i+1,i+2}\mathbf{b}_{i+2,i+3}^{\top}=\mathbf{0}. \tag{15}$$

The new unknowns are $\mathbf{b}_{i+1,i+3}$ and $\mathrm{b}_{i+1,i+2}$. Hence, for each of the nodes in the level, we define 2^{w+1} successors (for every $(\mathbf{b}_{i+1,i+3},\mathrm{b}_{i+1,i+2})\in\mathbb{F}_2^{w+1}$) to which we assign the elements of $\mathbb{F}_2^{w+1}$ in a lexicographic order.

(vi) We continue in the same manner, and for the level $(1,1+k)$, $3<k<w$ the associated equation is

$$\sum_{1=r_0<\cdots<r_m=1+k}\mathbf{b}_{1,r_1}\mathrm{b}_{r_1r_2}\cdots\mathrm{b}_{r_{m-2}r_{m-1}}\mathbf{b}_{r_{m-1},1+k}^{\top}=\mathbf{0}. \tag{16}$$

The only new unknowns appearing in the equation are $\mathbf{b}_{1,k}$, $\mathbf{b}_{2,k+1}$ and $\mathrm{b}_{2,k}$. Every possible value of $(\mathbf{b}_{1,k},\mathbf{b}_{2,k+1},\mathrm{b}_{2,k})\in\mathbb{F}_2^{2w+1}$ is assigned to 2^{2w+1} different lexicographically ordered successors of each of the nodes in the current level.

(vii) The associated equation for each of the levels $(i,i+k)$, $i\in\{2,\dots,w-k\}$, $3<k<w-1$ is

$$\sum_{i=r_0<\cdots<r_m=i+k}\mathbf{b}_{i,r_1}\mathrm{b}_{r_1r_2}\cdots\mathrm{b}_{r_{m-2}r_{m-1}}\mathbf{b}_{r_{m-1},i+k}^{\top}=\mathbf{0}. \tag{17}$$

Since the new unknowns appearing are $\mathbf{b}_{i+1,i+k}$ and $\mathrm{b}_{i+1,i+k-1}$, for each of the nodes of level $(i,i+k)$, we define 2^{w+1} successors (for each $(\mathbf{b}_{i+1,i+k},\mathrm{b}_{i+1,i+k-1})\in\mathbb{F}_2^{w+1}$) to which we assign the elements of $\mathbb{F}_2^{w+1}$ in a lexicographic order.

We should point out several properties of the tree we have just constructed.

- There is a one-to-one correspondence between the paths from the root to the leaves and all the possible assignments to the unknowns appearing in the system of equations (11) for all $j-i\ge 2$.
- An exhaustive search for solutions of the system (11), $j-i\ge 2$, corresponds to an exhaustive search through the tree. Thus all the solutions to the system are present in the tree, in the form of paths.

Trying out all the possible assignments of the unknowns $\mathbf{b}_{s,t}\in\mathbb{F}_2^w$ and $\mathrm{b}_{s,t}\in\mathbb{F}_2$ appearing in the system of equations (11), $j-i\ge 2$, and checking whether the system is satisfied, clearly will lead to finding all the solutions. However, the procedure can be made more efficient using the constructed tree and introducing a pruning technique. The pruning can be done based on a test for consistency of a partial assignment of the unknowns. We define the following depth-first search algorithm.

FindAllSolutions:

(1) Initiate an empty "history list" to keep track of the visited nodes. We assume that the list is being maintained throughout the algorithm. In essence it contains all the predecessors of the current node.

(2) At the root node solve the associated equation $\mathbf{b}_{1,2}\mathbf{b}_{2,3}^{\top} = \mathbf{0}$. Create a lexicographically ordered list of all the solutions $(\mathbf{b}_{1,2}, \mathbf{b}_{2,3}) \in \mathbb{F}_2^{2w}$ of the equation. Prune all the successors that are not in the list of solutions. Move to the leftmost successor, i.e. the one that corresponds to the first solution in the list of solutions. Update the history list by adding the chosen solution of the current equation.

(3) Move depth-first throughout the tree. At each node that is not a leaf
   - If the node is being visited for the first time (i.e. there is no list of solutions associated to it), use the history list to assign the values of $\mathbf{b}_{s,t} \in \mathbb{F}_2^{w}$ and $\mathrm{b}_{s,t} \in \mathbb{F}_2$ chosen in the previous steps in the equation that is associated to the current level. Solve the current equation, and put the solutions in a lexicographically ordered list. Prune all the successors that are not in the list of solutions.
     - If the list of solutions is not empty, move to the successor that corresponds to the first solution in the list of solutions, i.e. to the leftmost successor. Update the history list by adding the chosen solution to the current equation.
     - If the list of solutions is empty, go up to the predecessor.
   - If the node has been visited before, the current visit is due to moving up the tree. Read the last entry from the history list, and locate it in the solution list.
     - If it is not the last in the solution list, move to the successor node that is next in the solution list. Update the history list by deleting the last entry, and adding the solution that corresponds to the chosen successor node.
     - If the read entry is last in the solution list, go up to the predecessor. Update the history list by deleting the last entry.

(4) When the algorithm reaches a leaf, save the history list as one solution of the system (11), $j - i \geq 2$ in a list **Sol**. Go up to the predecessor.

(5) The algorithm ends when the root is reached again and the current solution list has been exhausted. In fact, at this point there are no more possible moves.

(6) Output the list **Sol**.

It is clear that since the algorithm **FindAllSolutions** traverses all the nodes that satisfy the equations associated to them, the list **Sol** contains all the solutions of the system (11), $j - i \geq 2$.

Note that, not all $\mathbf{b}_{s,t}$ and $\mathrm{b}_{s,t}$ that define the elements of the matrix $\mathbf{B}(\mathbf{x})$ appear in the equations (11). In particular, $\mathrm{b}_{1,i}$, $\mathrm{b}_{i,w}$, $1 < i < w$, as well as $\mathbf{b}_{1,w}$ do not appear in (11). This means that there are no special constraints for them and can take any value.

The algorithm **FindAllSolutions** can be modified to an equivalent one by changing the successor selection heuristics. In **FindAllSolutions**, at each first visit of a node a solution list is created with solutions to the associated equations. The solutions are ordered lexicographically, and this list is used for choosing the successors in this visit and all other subsequent visits of the node.

Let **FindAllSolutionsRand** be an algorithm that is the same as **FindAllSolutions**, except the solutions in the solution list at each node are being permuted using a random permutation once at the time of creation of the solution list. After that, this list is used in the same manner as in **FindAllSolutions**, and is not being permuted again. The output of the algorithm is a list **SolRand** that contains all the solutions of the system (11), $j - i \geq 2$.

It is not hard to see that the two algorithms are equivalent and in the same number of steps find all the solutions of the system (11), $j - i \geq 2$. The only difference is that **SolRand** is a permutation of the entries in **Sol**.

We introduce the algorithm **FindAllSolutionsRand** because it can be naturally modified for the purpose of finding a single random solution of the system (11), $j - i \geq 2$. Let **FindOneSolutionRand** be the subalgorithm of **FindAllSolutionsRand** that contains all the steps of **FindAllSolutionsRand** from the beginning until the first entry is written down in **SolRand**. In other words, we run **FindAllSolutionsRand** until one solution is found, and then we terminate the algorithm.

We note that a similar modification to **FindAllSolutions** is not useful in this setting, since the first solution that this algorithm finds is always the same. Instead, if we want to use **FindAllSolutions** to find a random solution, we would have to find all solutions first, i.e. run the complete algorithm, and then pick one based on some probability distribution, for example the uniform distribution. This is, however, a highly inefficient method of finding a random solution.

We should point out that the random solution the algorithm **FindOneSolutionRand** finds is not uniformly distributed in the set of all solutions. Indeed, at each node the solution list is permuted using a random permutation, thus all solutions have an equal probability to be first after the permutation is applied. However, if the pruned subtree of the node is not balanced, then some of the partial solutions in the solution list will yield more global solutions than others. As a consequence, the random permutation actually creates bias in the process.

This can be overcome if the permutation used at each node is not drawn from the uniform distribution, but rather from the distribution of the partial solutions of the successors with regards to the global solutions. However, without the knowledge of the nature of the pruned tree, or equivalently the set of global solutions **SolRand**, this can not be done. Characterizing completely the solutions **SolRand** is an interesting but nontrivial open problem.

Next, we present a procedure for constructing the $w \times 1$ matrix $\mathbf{A}(\mathbf{x})$ once the matrix $\mathbf{B}(\mathbf{x})$ is known.

We will use Proposition 12, and first construct a $w \times 1$ matrix of homogeneous quadratic Boolean polynomials $\mathbf{A}'_2(\mathbf{x})$ such that $\mathbf{B}_1(\mathbf{x})\mathbf{A}'_2(\mathbf{x})$ is a $w \times 1$ matrix of homogeneous quadratic Boolean polynomials.

Let the elements of $\mathbf{A}'_2(\mathbf{x})$ be denoted by $a'_k(\mathbf{x}) = \sum_{1\le i,j\le w} a_{ij}^{(k)} x_i x_j$, where $1 \le k \le w$.

ConstructA(x):

(1) For a given $\mathbf{B}(\mathbf{x}) = \mathbf{B}_1(\mathbf{x}) + \mathbf{B}_2$, calculate $\mathbf{T}(\mathbf{x}) = \mathbf{B}_1(\mathbf{x})\mathbf{A}'_2(\mathbf{x})$.
(2) Represent $\mathbf{T}(\mathbf{x})$ as $\mathbf{T}(\mathbf{x}) = \mathbf{T}_2(\mathbf{x}) + \mathbf{T}_3(\mathbf{x})$ where $\mathbf{T}_2(\mathbf{x})$ consists of homogeneous quadratic polynomials, and $\mathbf{T}_3(\mathbf{x})$ consists of homogeneous cubic polynomials.
(3) Solve $\mathbf{T}_3(\mathbf{x}) = \mathbf{0}$ in the unknowns $a_{ij}^{(k)}, 1 \le i, j, k \le w$. Let **SolA** be the set of solutions.
(4) For $s \in$ **SolA** construct $\mathbf{A}'_2(\mathbf{x})$.
(5) Let $\mathbf{A}'_1(\mathbf{x})$ be a $w \times 1$ matrix of linear Boolean polynomials.
(6) Construct $\mathbf{A}(\mathbf{x}) = \mathbf{B}_2(\mathbf{A}'_1(\mathbf{x}) + \mathbf{A}'_2(\mathbf{x}))$.

4.2 Efficient construction of LMQQs whose parastrophe is also quadratic. Although the algorithms **FindAllSolutions** and **FindAllSolutions-Rand** find all matrices $\mathbf{B}(\mathbf{x})$ with the desired properties, they are extremely inefficient. Even the algorithm **FindOneSolutionRand** requires solving equations of the form (11) at least $\frac{(w-2)(w-1)}{2}$ times (at least once for every level), that are in essence systems of equations over $\mathbb{F}_2$, and possibly many backtracking steps.

Next we present a very simple sufficient condition for a matrix $\mathbf{B}(\mathbf{x})$ to satisfy the conditions (11). This result provides a very simple, straightforward algorithmic construction of the matrix $\mathbf{B}(\mathbf{x})$, that does not require solving systems of equations, nor backtracking strategy, nor any kind of tests during the construction. Thus, it is very suitable for implementation.

Proposition 13. *Let for $i < j$, the elements $b_{ij}(\mathbf{x}) = \mathbf{x}^\top \cdot \mathbf{b}_{ij} + \mathrm{b}_{ij}$ of the $w \times w$ upper triangular matrix $\mathbf{B}(\mathbf{x})$ of linear Boolean polynomials satisfy the conditions:*

$$\mathbf{b}_{2k_1+1,2k_2+1} = \mathbf{0}, \mathbf{b}_{2k_1+2,2k_2+1} = \mathbf{0}, \mathbf{b}_{2k_1+2,2k_2+2} = \mathbf{0}, \quad \textit{and} \tag{18}$$

$$\mathrm{b}_{2k_1+2,2k_2+1} = 0, \tag{19}$$

where $k_1, k_2 \in \{0, \dots, \lfloor \frac{w}{2} \rfloor - 1\}$, and $b_{ii}(\mathbf{x}) = 1$.

Then, the elements of $\mathbf{B}^{-1}(\mathbf{x})$ are linear Boolean polynomials.

The vectors $\mathbf{b}_{2k_1+1,2k_2+2}$, *and the constants* $\mathrm{b}_{2k_1+1,2k_2+1}$, $\mathrm{b}_{2k_1+1,2k_2+2}$, $\mathrm{b}_{2k_1+2,2k_2+2}$ *can be chosen at random.*

Proof: We prove the lemma formally, i.e. we show that the condition from Proposition 11 holds.

First, let i be even, i.e., let $i = 2k_1 + 2$, for some $k_1 \in \{0, \ldots, \lfloor \frac{w}{2} \rfloor - 1\}$. Then $\mathbf{b}_{i,j} = \mathbf{0}$, for any j, and (11) is clearly satisfied.

Similarly, for j odd, $\mathbf{b}_{i,j} = \mathbf{0}$ for any i, and again (11) holds.

What is left is to analyze the case when $i = 2k_1 + 1$ and $j = 2k_2 + 2$, for some $k_1, k_2 \in \{0, \ldots, \lfloor \frac{w}{2} \rfloor - 1\}$. Using the same argument as for the previous two cases, (11) turns into:

$$\begin{aligned} &\sum_{i=r_0<\cdots<r_m=j} \mathbf{b}_{r_0 r_1} \mathrm{b}_{r_1 r_2} \cdots \mathrm{b}_{r_{m-2} r_{m-1}} \mathbf{b}^{\top}_{r_{m-1} r_m} \\ (20) \qquad = &\sum_{\substack{i = r_0 < r_1 < \cdots < r_{m-1} < r_m = j \\ r_0, r_{m-1} - odd, r_1, r_m - even}} \mathbf{b}_{r_0 r_1} \mathrm{b}_{r_1 r_2} \cdots \mathrm{b}_{r_{m-2} r_{m-1}} \mathbf{b}^{\top}_{r_{m-1} r_m}. \end{aligned}$$

Now, in any of the terms in the sum (20), the product $\mathrm{b}_{r_1 r_2} \cdots \mathrm{b}_{r_{m-2} r_{m-1}}$ is such that r_1 is even and r_{m-1} is odd. No matter the parity of $r_2, \ldots, r_{m-2}$, there exists $\mathrm{b}_{r_s r_t}$, $r_1 \le s < t \le r_{m-1}$, such that r_s is even and r_t is odd. But then, $\mathrm{b}_{r_s r_t} = 0$, and the term in question is equal to 0. Since this holds for every term, the sum (20) is equal to $\mathbf{0}$.

Again, we conclude that (11) holds.

Hence, from Proposition 11, the elements of $\mathbf{B}^{-1}(\mathbf{x})$ are linear Boolean polynomials. □

We now turn to finding a similar procedure for the construction of the vector $\mathbf{A}(\mathbf{x})$, such that $\mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x})$ is a vector of Boolean polynomials of degree 2.

We first need to find the form of $\mathbf{B}^{-1}(\mathbf{x})$.

Lemma 1. *Let the elements of the* $w \times w$ *upper triangular matrix* $\mathbf{B}(\mathbf{x})$ *of linear Boolean polynomials satisfy the conditions* (18) *and* (19)*. Then the matrix* $\mathbf{B}^{-1}(\mathbf{x})$ *has the same form as* $\mathbf{B}(\mathbf{x})$*, i.e. for* $i < j$*, the elements* $b'_{ij}(\mathbf{x}) = \mathbf{x}^{\top} \cdot \mathbf{b}'_{ij} + \mathrm{b}'_{ij}$ *of* $\mathbf{B}^{-1}(\mathbf{x})$ *satisfy:*

$$(21) \qquad \mathbf{b}'_{2k_1+1,2k_2+1} = \mathbf{0}, \mathbf{b}'_{2k_1+2,2k_2+1} = \mathbf{0}, \mathbf{b}'_{2k_1+2,2k_2+2} = \mathbf{0},$$

$$(22) \qquad \mathrm{b}'_{2k_1+2,2k_2+1} = 0$$

and $b'_{ii}(\mathbf{x}) = 1$.

Proof: From Proposition 11,

$$\mathbf{b}'_{ij} = \sum_{\substack{i = r_0 < \cdots < r_m = j \\ t \in \{0, \ldots, m-1\}}} \mathrm{b}_{r_0,r_1} \cdots \mathbf{b}_{r_t r_{t+1}} \cdots \mathrm{b}_{r_{m-1},r_m}, \tag{23}$$

$$\mathrm{b}'_{ij} = \sum_{i=r_0<\cdots<r_m=j} \mathrm{b}_{r_0 r_1} \mathrm{b}_{r_1 r_2} \cdots \mathrm{b}_{r_{m-1},r_m}. \tag{24}$$

First we prove (21).

Let i and j be both odd. We analyze one term $\mathrm{b}_{r_0,r_1} \cdots \mathbf{b}_{r_t r_{t+1}} \cdots \mathrm{b}_{r_{m-1},r_m}$ from (23).

If every r_s, $0 \le s \le m$ is odd, then r_t and r_{t+1} are odd as well, and from (18), $\mathbf{b}_{r_t r_{t+1}} = \mathbf{0}$. Hence, $\mathrm{b}_{r_0,r_1} \cdots \mathbf{b}_{r_t r_{t+1}} \cdots \mathrm{b}_{r_{m-1},r_m} = \mathbf{0}$. If there is at least one r_l that is even, $0 < l < m$, then either $l = t$, when from (18), $\mathbf{b}_{r_t r_{t+1}} = \mathbf{0}$, or $l \neq t$ and r_{l+1} is odd, when from (19), $\mathrm{b}_{r_l r_{l+1}} = 0$. In both cases, again, $\mathrm{b}_{r_0,r_1} \cdots \mathbf{b}_{r_t r_{t+1}} \cdots \mathrm{b}_{r_{m-1},r_m} = \mathbf{0}$. Hence $\mathbf{b}'_{ij} = \mathbf{0}$.

Let i and j be both even. Again we look at one term from (12). If every r_s, $0 \le s \le m$ is even, then r_t and r_{t+1} are even, and thus from (18), $\mathbf{b}_{r_t r_{t+1}} = \mathbf{0}$. If at least one r_l is odd, $0 < l < m$, then either $l = t + 1$, when from (18), $\mathbf{b}_{r_t r_{t+1}} = \mathbf{0}$, or $l \neq t + 1$ and r_{l-1} is even, when from (19), $\mathrm{b}_{r_{l-1} r_l} = 0$. Again, all the cases infer $\mathrm{b}_{r_0,r_1} \cdots \mathbf{b}_{r_t r_{t+1}} \cdots \mathrm{b}_{r_{m-1},r_m} = \mathbf{0}$. Hence $\mathbf{b}'_{ij} = \mathbf{0}$.

Let i be even and j be odd. Then there exists r_s, $0 \le s \le m$ such that r_s is even, and r_{s+1} is odd. If $s = t$, from (18), $\mathbf{b}_{r_t r_{t+1}} = \mathbf{0}$, and if $s \neq t$, then from (19), $\mathrm{b}_{r_s r_{s+1}} = 0$. Similarly as above, we conclude that $\mathbf{b}'_{ij} = \mathbf{0}$.

From the above it follows that (21) holds.

The last reasoning can also be directly applied to conclude that $\mathrm{b}'_{i,j} = 0$ when i is even and j is odd, i.e. that (22) holds. □

Let the elements of the vector $\mathbf{A}(\mathbf{x})$ of Boolean polynomials be denoted by $a_i(\mathbf{x})$, $1 \le i \le w$. For the vector $\mathbf{A}(\mathbf{x})$ we have the following lemma.

Lemma 2. *Let for all odd i, $a_i(\mathbf{x})$ be a quadratic Boolean polynomial, and for all even i let $a_i(\mathbf{x})$ be a linear Boolean expression. Let $\mathbf{B}(\mathbf{x})$ be an upper triangular matrix of linear Boolean polynomials with* 1*s on the diagonal, such that* (18) *and* (19) *are satisfied.*

Then $\mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x})$ is a vector of quadratic Boolean polynomials.

Proof: From Lemma 1, the only elements in the matrix $\mathbf{B}^{-1}(\mathbf{x})$ that can be linear Boolean polynomials are $b'_{2k_1+1,2k_2+2}(\mathbf{x})$, for some k_1, k_2. The others are all constants. The elements of the vector $\mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x})$ are of the form $\sum_{i=1}^{w} b'_{k,i}(\mathbf{x})$ $a_i(\mathbf{x})$ so for odd i, $b'_{k,i}(\mathbf{x})a_i(\mathbf{x})$ is quadratic since $b'_{k,i}(\mathbf{x})$ is a constant and $a_i(\mathbf{x})$ is quadratic, and also for even i, since $b'_{k,i}(\mathbf{x})$ is at most linear and $a_i(\mathbf{x})$ is linear, $b'_{k,i}(\mathbf{x})a_i(\mathbf{x})$ is again quadratic.

Thus, the elements of the vector $\mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x})$ are quadratic Boolean polynomials. □

Finally, we are ready to state the main theorem in this part, that gives sufficient conditions for a left quasigroup from $\mathcal{TLQ}_{2^w}$ of algebraic degree 2, to have a parastrophe that is again of degree 2.

Theorem 4. *Let $q(\mathbf{x},\mathbf{y})$ be a left quasigroups from $\mathcal{TLQ}_{2^w}$ of algebraic degree 2, i.e. let q be of the form*

$$q(\mathbf{x},\mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \cdot \mathbf{y}$$

where $\mathbf{A}(\mathbf{x})$ is a vector of Boolean polynomials $a_i(\mathbf{x})$ such that:

- *for all odd i, $a_i(\mathbf{x})$ is a quadratic Boolean polynomial, and*
- *for all even i, $a_i(\mathbf{x})$ is a linear Boolean polynomial,*

and $\mathbf{B}(\mathbf{x})$ is an upper triangular matrix of linear Boolean polynomials $b_{ij}(\mathbf{x})$ with 1s on the diagonal, such that:

- *for all odd i, and all odd j, $b_{ij}(\mathbf{x}) = \mathrm{b}_{ij}$, where $\mathrm{b}_{ij} \in \mathbb{F}_2$,*
- *for all even i, and all even j, $b_{ij}(\mathbf{x}) = \mathrm{b}_{ij}$, where $\mathrm{b}_{ij} \in \mathbb{F}_2$,*
- *for all odd i, and all even j, $b_{ij}(\mathbf{x})$ is a linear Boolean polynomial, and*
- *for all even i, and all odd j, $b_{ij}(\mathbf{x}) = 0$.*

Then q has a left parastrophe $q_\backslash$ that is again of degree 2.

Proof: The claim follows directly from Proposition 13, Lemma 2 and Proposition 8. □

If an LMQQ has a left parastrophe that is again of degree 2, then this property will be preserved under linear isotopy, i.e. the following holds.

Proposition 14. *Let q be a left quasigroups from $\mathcal{TLQ}_{2^w}$ of algebraic degree 2, that has a left parastrophe again of degree 2. Then every linearly isotopic quasigroup q' has also a left parastrophe of degree 2.*

Proof: Let q' be linearly isotopic to q, i.e., let

$$q'(\mathbf{x},\mathbf{y}) = \mathbf{D_3} \cdot q(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}, \mathbf{D_2}\mathbf{y} + \mathbf{c_2}) + \mathbf{c_3}.$$

Then from the identity $q'(\mathbf{x}, q'_\backslash(\mathbf{x},\mathbf{y})) = \mathbf{y}$ we have:

$$\begin{aligned}
& \mathbf{D_3} \cdot q(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}, \mathbf{D_2}q'_\backslash(\mathbf{x},\mathbf{y}) + \mathbf{c_2}) + \mathbf{c_3} = \mathbf{y} \\
\Leftrightarrow \quad & \mathbf{A}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}) + \mathbf{B}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}) \cdot (\mathbf{D_2}q'_\backslash(\mathbf{x},\mathbf{y}) + \mathbf{c_2}) = \mathbf{D_3}^{-1}\mathbf{y} + \mathbf{D_3}^{-1}\mathbf{c_3} \\
\Leftrightarrow \quad & q'_\backslash(\mathbf{x},\mathbf{y}) = \mathbf{D_2}^{-1}\mathbf{B}^{-1}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}) \cdot \mathbf{A}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}) \\
& + \mathbf{D_2}^{-1}\mathbf{B}^{-1}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}) \cdot \mathbf{D_3}^{-1}\mathbf{y} \\
& + \mathbf{D_2}^{-1}\mathbf{B}^{-1}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1}) \cdot \mathbf{D_3}^{-1}\mathbf{c_3} + \mathbf{D_2}^{-1}\mathbf{c_2}.
\end{aligned}$$

Since the form of $\mathbf{A}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1})$ is the same as the one of $\mathbf{A}(\mathbf{x})$, and the form of $\mathbf{B}^{-1}(\mathbf{D_1}\mathbf{x} + \mathbf{c_1})$ is the same as the one of $\mathbf{B}^{-1}(\mathbf{x})$, we can conclude that the algebraic degree of $q'_\backslash$ is 2. □

We conclude this part with an example of the described construction of an LMQQ that has a quadratic left parastrophe.

Example 2. We will construct an LMQQ of order 2^4 obtained by applying isotopic transformation to an LMQQ from $\mathcal{TLQ}_{2^4}$ that satisfies Theorem 4.

We first construct q.

Let $\mathbf{A}(\mathbf{x})$ be a vector of quadratic Boolean polynomials in $\mathbf{x} = (x_1, x_2, x_3, x_4)$, given by:

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} 1 + x_1 + x_2 + x_1x_3 + x_2x_3 + x_4 + x_1x_4 + x_3x_4 \\ 1 + x_1 + x_2 \\ x_1x_2 + x_3 + x_4 + x_1x_4 + x_3x_4 \\ 1 + x_3 + x_4 \end{bmatrix}.$$

Let $\mathbf{B}(\mathbf{x})$ be an upper triangular matrix of linear Boolean polynomials in $\mathbf{x} = (x_1, x_2, x_3, x_4)$, with 1s on the diagonal given by:

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} 1 & x_1 + x_2 + x_3 & 1 & x_2 + x_3 + x_4 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 + x_2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{B}^{-1}(\mathbf{x}) = \begin{bmatrix} 1 & x_1 + x_2 + x_3 & 1 & 1 + x_1 + x_2 + x_4 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 + x_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then $q(\mathbf{x}, \mathbf{y}) = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x}) \cdot \mathbf{y}$ is

$$q(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} 1 + x_1 + x_2 + x_1x_3 + x_2x_3 + x_4 + x_1x_4 + x_3x_4 + y_1 + x_1y_2 + \\ + x_2y_2 + x_3y_2 + y_3 + x_2y_4 + x_3y_4 + x_4y_4 \\ 1 + x_1 + x_2 + y_2 + y_4 \\ x_1x_2 + x_3 + x_4 + x_1x_4 + x_3x_4 + y_3 + y_4 + x_2y_4 \\ 1 + x_3 + x_4 + y_4 \end{bmatrix}.$$

The parastrophe $q_{\backslash}(\mathbf{x}, \mathbf{y}) = \mathbf{B}^{-1}(\mathbf{x})\mathbf{A}(\mathbf{x}) + \mathbf{B}^{-1}(\mathbf{x}) \cdot \mathbf{y}$ is

$$q_{\backslash}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} x_1x_2 + x_3 + x_1x_3 + x_2x_3 + x_4 + x_1x_4 + x_2x_4 + x_3x_4 + y_1 + x_1y_2 + \\ + x_2y_2 + x_3y_2 + y_3 + y_4 + x_1y_4 + x_2y_4 + x_4y_4 \\ x_1 + x_2 + x_3 + x_4 + y_2 + y_4 \\ 1 + x_2 + x_1x_2 + x_2x_3 + x_1x_4 + x_2x_4 + x_3x_4 + y_3 + y_4 + x_2y_4 \\ 1 + x_3 + x_4 + y_4 \end{bmatrix}.$$

We next apply linear isotopy to q defined by the nonsingular matrices:

$$\mathbf{D_1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \mathbf{D_2} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{D_3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

and the vectors: $\mathbf{c_1} = (0,1,0,0)$, $\mathbf{c_2} = (0,0,1,0)$, $\mathbf{c_3} = (1,1,0,1)$.

We obtain the quasigroup
$q'(\mathbf{x},\mathbf{y}) = \mathbf{D_3}(\mathbf{A}(\mathbf{D_1}\mathbf{x}+\mathbf{c_1}) + \mathbf{B}(\mathbf{D_1}\mathbf{x}+\mathbf{c_1})\cdot(\mathbf{D_2}\mathbf{y}+\mathbf{c_2})) + \mathbf{c_3}$:

$$q'(\mathbf{x},\mathbf{y}) = \begin{bmatrix} x_1x_2 + x_1x_3 + x_2x_4 + x_3y_1 + y_2 + x_4y_2 + x_3y_3 + y_4 + x_3y_4 \\ \begin{array}{c} x_2 + x_3 + x_2x_3 + x_1x_4 + x_2y_1 + x_3y_1 + \\ +y_3 + x_2y_3 + x_3y_3 + x_2y_4 + x_3y_4 \end{array} \\ \begin{array}{c} 1 + x_1 + x_1x_2 + x_1x_3 + x_4 + x_2x_4 + \\ +y_1 + x_3y_1 + x_4y_2 + y_3 + x_3y_3 + x_3y_4 \end{array} \\ \begin{array}{c} x_1 + x_3 + x_2x_3 + x_1x_4 + x_2y_1 + x_3y_1 + \\ +y_2 + y_3 + x_2y_3 + x_3y_3 + x_2y_4 + x_3y_4 \end{array} \end{bmatrix}.$$

The parastrophe $q'_{\backslash}(\mathbf{x},\mathbf{y})$ is:

$$q'_{\backslash}(\mathbf{x},\mathbf{y}) = \begin{bmatrix} \begin{array}{c} x_2 + x_3 + x_4 + x_1x_4 + x_2x_4 + x_3x_4 + \\ +y_1 + x_2y_3 + x_3y_3 + y_4 + x_2y_4 + x_3y_4 \end{array} \\ \begin{array}{c} x_1 + x_2 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_4 + x_3x_4 + \\ +y_1 + y_2 + y_3 + x_3y_3 + x_4y_3 + y_4 + x_3y_4 \end{array} \\ \begin{array}{c} 1 + x_1x_2 + x_3 + x_1x_3 + x_2x_3 + x_2x_4 + \\ +y_2 + y_3 + x_2y_3 + x_4y_3 + y_4 + x_2y_4 \end{array} \\ \begin{array}{c} x_3 + x_1x_4 + x_2x_4 + x_3x_4 + y_1 + y_3 + \\ +x_2y_3 + x_3y_3 + x_2y_4 + x_3y_4 \end{array} \end{bmatrix}.$$

5. Conclusions

In this paper, we investigated Left Multivariate Quasigroups, with a particular focus on Left Multivariate Quadratic Quasigroups (LMQQs). We provided an efficient construction of general LMQQs of any order and degree, as well as of more particular ones that are affine in the variable $\mathbf{y}$, and whose parastrophe is easy to express. The main goal was to distinguish a class of such LMQQs that have, even more, a left parastrophe that is again an LMQQ, and thus has a short symbolic form. First, we determined sufficient conditions for the parastrophe to be quadratic, and then presented a general backtracking algorithm that finds all LMQQs that satisfy these conditions. Since the backtracking nature of the algorithm makes it rather inefficient, we provide additional very simple sufficient conditions. These sufficient conditions provide an easy, efficient and straightforward construction of LMQQs whose left parastrophe is quadratic.

As the simplification of the sufficient conditions was driven by the main premise to find an efficient algorithmic construction of LMQQs whose left parastrophe is quadratic, and thus led to a narrow class of such LMQQs, the authors can set apart two open problems: Can a different strategy lead to efficient construction algorithm of a broader class of LMQQs whose left parastrophe is quadratic and that are left affine, and, can the same be accomplished for LMQQs that are quadratic in $\mathbf{y}$. In either case, a special attention must be paid to their suitability for use in multivariate cryptosystems.

Acknowledgment. The authors would like to thank the anonymous referee for the suggestions and comments that substantially helped improve the quality of the paper. The final form of the algorithms given in Section 4 was made in response to her/his inquiry.

References

[1] Albert A.A., *Quasigroups. I*, Trans. Amer. Math. Soc. **54** (1943), 507–519.

[2] Ahlawat R., Gupta K., Pal S.K., *Fast generation of multivariate quadratic quasigroups for cryptographic applications*, Proceeding of Mathematics in Defence, 2009.

[3] Belousov V.D., *Osnovi teorii kvazigrup i lup* (in Russian), Nauka, Moscow, 1967.

[4] Carter G., Dawson E., Nielsen L., *A latin square version of DES*, in Proc. Workshop of Selected Areas in Cryptography, Ottawa, Canada, 1995.

[5] Chen Y., Knapskog S.J., Gligoroski D., *Multivariate quadratic quasigroups (MQQs): Construction, bounds and complexity*, INSCRYPT, Proceedings of the 6th International Conference on Information Security and Cryptology, 2010.

[6] Christov A., *Kryptografie založená na teorii kvazigrup*, Diploma Thesis, Charles University, Prague, 2009, available at: `http://artax.karlin.mff.cuni.cz/ chria3am/thesis/`.

[7] Cooper J., Donovan D., Seberry J., *Secret sharing schemes arising from Latin Squares*, Bull. Inst. Combin. Appl. **4** (1994), 33–43.

[8] Gligoroski D., Markovski S., Kocarev L., Gusev M., *Edon80 Hardware Synchronous stream cipher*, SKEW 2005 - Symmetric Key Encryption Workshop, Aarhus Denmark, 2005.

[9] Gligoroski D., Markovski S., Knapskog S.J., *Multivariate quadratic trapdoor functions based on multivariate quadratic quasigroups*, MATH'08: Proceedings of the American Conference on Applied Mathematics, pp. 44–49, 2008. Extended version of the paper: *Public key block cipher based on multivariate quadratic quasigroups*, in Cryptology ePrint Archive, Report 2008/320, `http://eprint.iacr.org`.

[10] Gligoroski D., Ødegård R.S., Jensen R.E., Perret L., Faugère J.-C., Knapskog S.J., Markovski S., *MQQ-SIG, an ultra-fast and provably CMA resistant digital signature scheme*, in Proc. of INTRUST 2011, LNCS vol. 7222, 2012, pp. 184–203.

[11] Gligoroski D., Ødegård R.S., Mihova M., Knapskog S.J., Drápal A., Klima V., *Cryptographic Hash Function* EDON-$\mathcal{R}$, SHA-3 Algorithm Submission, 2008.

[12] Gligoroski D., Klima V., Knapskog S.J., El-Hadedy M., Amundsen J., Mjølsnes S.F., *Cryptographic Hash Function* BLUE MIDNIGHT WISH , SHA-3 Algorithm Submission, 2008.

[13] Klimov A., Shamir A., *A new class of invertible mappings*, 4th Workshop on Cryptographic Hardware and Embedded Systems CHES 2002, pp. 471-484, Springer, 2002.

[14] Markovski S., Mileva A., *Cryptographic Hash Function NaSHA*, SHA-3 Algorithm Submission, 2008.

[15] Matsumoto M., Saito M., Nishimura T., Hagita M., *CryptMT Stream Cipher Version 3*, in Workshop Record of SASC 2007: The State of the Art of Stream Ciphers, eSTREAM report 2007/028, 2007, available at: `http://www.ecrypt.eu.org/stream/papers.html`.

[16] Nguyen D.V., Chilappagari S.K., Marcellin M.W., Vasić B., *LDPC codes from latin squares free of small trapping sets*, arXiv:1008.4177, 2010, available at: `http://arxiv.org/abs/1008.4177`.

[17] Rivest R.L., *Permutation polynomials modulo* 2^w, Finite Fields Appl. **7** (2001), 287–292.

[18] Samardjiska S., Chen Y., Gligoroski D., *Algorithms for construction of multivariate quadratic quasigroups (MQQs) and their parastrophe operations in arbitrary Galois fields*, Journal of Information Assurance and Security **7** (2012), 164–172.

[19] Samardjiska S., Markovski S., Gligoroski D., *Multivariate quasigroups defined by T-functions*, Proceedings of the 2nd International Conference on Symbolic Computation and Cryptography, 2010, pp. 117–127.

[20] Schnorr C.P., Vaudenay S., *Black Box Cryptanalysis of hash networks based on multipermutations*, in Advances of Cryptology - EUROCRYPT'94, Springer, Berlin, 1995.

[21] Shannon C.E., *Communication theory of secrecy systems*, Bell Sys. Tech. J. **28** (1949), 657–715.

[22] Smith J.D.H., *An Introduction to Quasigroups and Their Representations*, Chapman and Hall/CRC, Boca Raton, FL, 2007.

[23] Zhang L., Huang Q., Lin S., Abdel-Ghaffar K., Blake I.F., *Quasicyclic LDPC codes on Latin squares and the ranks of their parity-check matrices*, in Inf. Theory and Appl. Workshop, 2010.

Department of Telematics, Norwegian University of Science and Technology, Trondheim, Norway

E-mail: simona.samardjiska@item.ntnu.no,
danilo.gligoroski@item.ntnu.no

(*Received* October 15, 2011, *revised* May 19, 2012)