

# Quadratization of Symmetric Pseudo-Boolean Functions

Martin Anthony<sup>a</sup>, Endre Boros<sup>b</sup>, Yves Crama<sup>c</sup>, Aritanan Gruber<sup>d,\*</sup>

<sup>a</sup>*Department of Mathematics, London School of Economics and Political Science, UK.*

*m.anthony@lse.ac.uk*

<sup>b</sup>*MSIS and RUTCOR, Rutgers University, NJ, USA. endre.boros@rutgers.edu*

<sup>c</sup>*QuantOM, HEC Management School, University of Liege, Belgium.*

*yves.crama@ulg.ac.be*

<sup>d</sup>*RUTCOR, Rutgers University, NJ, USA. aritanan.gruber@rutgers.edu*

---

## Abstract

A *pseudo-Boolean function* is a real-valued function  $f(x) = f(x_1, x_2, \dots, x_n)$  of  $n$  binary variables, that is, a mapping from  $\{0, 1\}^n$  to  $\mathbb{R}$ . For a pseudo-Boolean function  $f(x)$  on  $\{0, 1\}^n$ , we say that  $g(x, y)$  is a *quadratization* of  $f$  if  $g(x, y)$  is a quadratic polynomial depending on  $x$  and on  $m$  *auxiliary* binary variables  $y_1, y_2, \dots, y_m$  such that  $f(x) = \min\{g(x, y) : y \in \{0, 1\}^m\}$  for all  $x \in \{0, 1\}^n$ . By means of quadratizations, minimization of  $f$  is reduced to minimization (over its extended set of variables) of the quadratic function  $g(x, y)$ . This is of practical interest because minimization of quadratic functions has been thoroughly studied for the last few decades, and much progress has been made in solving such problems exactly or heuristically. A related paper [1] initiated a systematic study of the minimum number of auxiliary  $y$ -variables required in a quadratization of an arbitrary function  $f$  (a natural question, since the complexity of minimizing the quadratic function  $g(x, y)$  depends, among other factors, on the number of binary variables). In this paper, we determine more precisely the number of auxiliary variables required by quadratizations of *symmetric* pseudo-Boolean functions  $f(x)$ , those functions whose value depends only on the Hamming weight of the input  $x$  (the number of variables equal to 1).

*Keywords:* Symmetric (pseudo-)Boolean functions, Nonlinear and quadratic binary optimization, Reformulation methods for polynomials.

*2000 MSC:* 06E30, 90C09, 90C20

---

\*Corresponding author.

---

## 1. Introduction

### 1.1. Quadraticizations of pseudo-Boolean functions

A *pseudo-Boolean function* is a real-valued function  $f(x) = f(x_1, x_2, \dots, x_n)$  of  $n$  binary variables, that is, a mapping from  $\{0, 1\}^n$  to  $\mathbb{R}$ . It is well-known that every pseudo-Boolean function can be uniquely represented as a multilinear polynomial in its variables. *Nonlinear binary optimization problems*, or *pseudo-Boolean optimization (PBO) problems*, of the form

$$\min\{f(x) : x \in \{0, 1\}^n\},$$

where  $f(x)$  is a pseudo-Boolean function, have attracted the attention of numerous researchers, and they are notoriously difficult, as they naturally encompass a broad variety of models such as maximum satisfiability, maximum cut, graph coloring, simple plant location, and so on. Many approaches have been proposed for the solution of PBO problems; the reader may refer to [3, 5, 17] for general overviews. In recent years, several authors have revisited an approach initially proposed by Rosenberg [20]. This involves reducing PBO to its quadratic case (QPBO) by relying on the following concept.

**Definition 1.1.** *For a pseudo-Boolean function  $f(x)$  on  $\{0, 1\}^n$ , we say that  $g(x, y)$  is a quadraticization of  $f$  if  $g(x, y)$  is a quadratic polynomial depending on  $x$  and on  $m$  auxiliary binary variables  $y_1, y_2, \dots, y_m$ , such that*

$$f(x) = \min\{g(x, y) : y \in \{0, 1\}^m\} \quad \text{for all } x \in \{0, 1\}^n.$$

Clearly, if  $g(x, y)$  is a quadraticization of  $f$ , then

$$\min\{f(x) : x \in \{0, 1\}^n\} = \min\{g(x, y) : x \in \{0, 1\}^n, y \in \{0, 1\}^m\},$$

so that the minimization of  $f$  is reduced through this transformation to the QPBO problem of minimizing  $g(x, y)$  (much like linearization techniques reduce QPBO to a linear optimization problem in 0–1 variables; see [3, 5]).

We are also interested (see [1]) in special types of quadraticizations, which we call  *$y$ -linear quadraticizations*, which contain no products of auxiliary variables. If  $g(x, y)$  is a  $y$ -linear quadraticization, then  $g$  can be written as

$$g(x, y) = q(x) + \sum_{i=1}^m a_i(x)y_i,$$

where  $q(x)$  is quadratic in  $x$  and each  $a_i(x)$  is a linear function of  $x$ . When minimizing  $g$  over  $y$ , each product  $a_i(x)y_i$  takes the value  $\min\{0, a_i(x)\}$ .

Rosenberg [20] has proved that every pseudo-Boolean function  $f(x)$  has a quadratization, and that a quadratization can be efficiently computed from the polynomial expression of  $f$ . This also easily follows from our observations in Section 1.2 that every monomial has a quadratization. (It is also the case — see [1] — that any pseudo-Boolean function has a  $y$ -linear quadratization.) Of course, quadratic PBO problems remain difficult in general, but this special class of problems has been thoroughly studied for the last few decades, and much progress has been made in solving large instances of QPBO, either exactly or heuristically. Quadratization has emerged in recent years as one of the most successful approaches to the solution of very large-scale PBO problems arising in computer vision applications. (See, for instance, Boykov, Veksler and Zabih [4], Kolmogorov and Rother [15], Kolmogorov and Zabik [16], Rother, Kolmogorov, Lempitsky and Szummer [22], Kohli, Ladický and Torr [13], Kohli, Pawan Kumar and Torr [14], Boros and Gruber [2], Fix, Gruber, Boros and Zabih [7], Freedman and Drineas [8], Ishikawa [10, 11], Ramalingam, Russell, Ladický and Torr [19], Rother, Kohli, Feng and Jia [21], Kappes et al. [12].)

In a related paper, the present authors [1] initiated a systematic study of quadratizations of pseudo-Boolean functions. We investigated the minimum number of auxiliary  $y$ -variables required in a quadratization. We showed, in particular, that there are pseudo-Boolean functions of  $n$  variables for which every quadratization must involve at least  $\Omega(2^{n/2})$  auxiliary variables and, conversely, that  $O(2^{n/2})$  auxiliary variables always suffice for every function. Other authors have established more precise upper bounds for special subclasses of pseudo-Boolean functions:  $(n - 1)$  auxiliary variables for symmetric functions (Fix [6]),  $\lfloor \frac{n-1}{2} \rfloor$  auxiliary variables for positive monomials (Ishikawa [11]), two auxiliary variables for certain types of submodular functions (Kohli et al. [13], Ramalingam et al. [19]), etc. In this paper, our focus is on *symmetric* pseudo-Boolean functions. We introduce this class in the next section.

### 1.2. Symmetric functions

A symmetric pseudo-Boolean function is one in which the value of the function depends only on the weight of the input. More precisely, a pseudo-Boolean function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  is *symmetric* if there is a discrete function  $k : \{0, 1, \dots, n\} \rightarrow \mathbb{R}$  such that  $f(x) = k(l)$  where  $l = |x| = \sum_{j=1}^n x_j$  is the

Hamming weight (number of ones) of  $x$ . In another way,  $f$  is symmetric if it is invariant under any permutation of the coordinates  $\{1, 2, \dots, n\}$  of its variables. Here, we investigate the number of auxiliary variables required in quadratizations of symmetric functions.

**Example 1.2.** Consider the *negative monomial*

$$-\prod_{i=1}^n x_i = -x_1 x_2 \cdots x_n.$$

This elementary symmetric pseudo-Boolean function has the following *standard quadratization* which requires only one auxiliary variable (Freedman and Drineas [8]):

$$s_n(x_1, x_2, \dots, x_n, y) = y \left( n - 1 - \sum_{i=1}^n x_i \right).$$

The reason is as follows: unless all the  $x_i$  are 1, then the quantity in parentheses in the expression for  $s_n$  is non-negative and the minimum value of  $s_n$  is therefore 0, obtained when  $y = 0$ ; and, if all  $x_i$  are 1, the expression equals  $-y$ , minimized when  $y = 1$ , giving value  $-1$ . In both cases, the minimum value of  $s_n$  is the same as the value of the negative monomial.  $\square$

**Example 1.3.** The *positive monomial* is the symmetric function

$$\prod_{i=1}^n x_i = x_1 x_2 \cdots x_n.$$

Ishikawa [11] showed that it can be quadratized using  $\lfloor \frac{n-1}{2} \rfloor$  auxiliary variables. This is currently the best available bound for positive monomials; see also [7].  $\square$

Of course, contrary to the general case of PBO, minimizing a symmetric pseudo-Boolean function is a trivial task since the function takes at most  $n + 1$  distinct values. Our motivation for analyzing ‘best possible’ quadratizations of such functions, therefore, is to be found elsewhere. For one thing, many quadratization procedures proceed by *independently quadratizing every monomial* of the multilinear expression of  $f$  (as explained in the examples above), and by adding the resulting quadratic expressions. Then, the number of auxiliary variables appearing in the final quadratization is the sum of

the number of auxiliary variables required for the monomials. Therefore, it is of interest to understand the ‘best possible’ quadratizations of monomials.

Other symmetric functions similarly arise as ‘building blocks’ in the definition of more complex objective functions considered, e.g., in computer vision models based on so-called ‘clique potentials’; see Kohli et al. [13, 14]. Here again, the objective function can be quadratized by *independently handling its symmetric constituents*.

The analysis presented in Sections 2-4 applies to arbitrary symmetric functions, and it provides a *unifying framework* for the derivation of a series of previously scattered results, including the best-known upper bound for positive monomials and a slightly improved upper bound for arbitrary symmetric functions.

In a broader setting, and beyond the potential algorithmic implications of our results, we are mostly interested in the theoretical question of understanding the *mathematical properties*, and in particular the size of quadratizations of pseudo-Boolean functions. The bounds established in [1] are asymptotically tight for generic functions. However, it is probably quite hard to get *sharp bounds for specific functions* or classes of functions. Symmetric functions are a natural class of ‘elementary’ functions, much studied in other contexts (in particular, in Boolean circuit and neural network theory; see, e.g., [9, 18, 23, 25]), whose *analysis remains very challenging*, but for which we can hope to obtain *more precise bounds*. Section 4 and Section 5 contain results in this direction.

### 1.3. Outline

In Section 2, we prove a representation theorem for discrete functions and several corollaries which provide useful tools for expressing symmetric pseudo-Boolean functions. In Section 3 we explain how we can use such representations to construct quadratizations, and we present the implications for upper bounds on the number of auxiliary variables required for several classes of symmetric functions in Section 4. Section 5 presents two types of lower bounds on the number of auxiliary variables for quadratizations of symmetric functions: an existence result, establishing that there are symmetric functions needing a rather large number of auxiliary variables; and a concrete lower bound on the number of auxiliary variables in any  $y$ -linear quadratization of the parity function.

## 2. A representation theorem

In this section, we present a result that will be key in our approach to obtaining quadratizations of symmetric pseudo-Boolean functions. The result is a ‘representation theorem’ that expresses an arbitrary discrete function defined on the non-negative integers  $\mathbb{N} = \{0, 1, \dots\}$  as a linear combination of terms of the form  $[i - \epsilon_i - l]^-$ , for arbitrary parameters  $\epsilon_i \in (0, 1]$  (for any real number  $a$ ,  $[a]^-$  denotes  $\min(a, 0)$ ). In Section 3, we will apply the representation theorem to the discrete function  $k(l)$  associated with a symmetric pseudo-Boolean function, for suitable choices of the parameters  $\epsilon_i$ .

In its most general form, our representation result is as follows.

**Theorem 2.1.** *Let  $0 < \epsilon_i \leq 1$ , for  $i \in \mathbb{N}$ . Then every discrete function  $k : \mathbb{N} \rightarrow \mathbb{R}$  can be represented uniquely in the form*

$$k(l) = \sum_{i=0}^{\infty} \alpha_i [i - \epsilon_i - l]^- \quad \text{for all } l \in \mathbb{N}. \quad (1)$$

*Proof.* In order to find a sequence of real coefficients  $\alpha_0, \alpha_1, \dots$  which establish the required representation, we need to solve the infinite system of linear equations (1). Note that the term  $[i - \epsilon_i - l]^-$  vanishes when  $i \geq l + 1$ , so that the right-hand side of (1) boils down to a finite sum and the system reads:

$$k(l) = \sum_{i=0}^l \alpha_i [i - \epsilon_i - l]^- = \sum_{i=0}^l \alpha_i (i - \epsilon_i - l) \quad \text{for all } l \in \mathbb{N}. \quad (2)$$

This system is lower-triangular with nonzero diagonal entries  $-\epsilon_i$  ( $i = 0, \dots, l$ ). So, it does indeed have a unique solution and therefore the representation exists and is unique.

In order to obtain an explicit expression of the coefficients  $\alpha_i$ , we can proceed by induction on  $l$ . When  $l = 0$ , the first equation in (2) immediately yields  $\alpha_0 = -\frac{1}{\epsilon_0} k(0)$ . Assume now that we have found values of  $\alpha_0, \alpha_1, \dots, \alpha_{j-1}$  such that (2) is satisfied for  $l = 0, 1, \dots, j - 1$ . Then, from the equation (2) written for  $l = j$ , we deduce that  $\alpha_j$  must satisfy

$$-\epsilon_j \alpha_j = k(j) + \sum_{i=0}^{j-1} \alpha_i (\epsilon_i + j - i) \quad (3)$$

and the value of  $\alpha_j$  is uniquely determined from this equation.  $\square$

When all the  $\epsilon_i$  are equal, we can obtain a closed-form expression of the coefficients in the representation (1).

**Theorem 2.2.** *Let  $0 < \epsilon \leq 1$ . Then every discrete function  $k : \mathbb{N} \rightarrow \mathbb{R}$  can be represented uniquely in the form*

$$k(l) = \sum_{i=0}^{\infty} \alpha_i [i - \epsilon - l]^- \quad \text{for all } l \in \mathbb{N}$$

where, for all  $i \in \mathbb{N}$ , the value of  $\alpha_i$  is

$$\alpha_i = - \sum_{j=0}^{i-2} \frac{(\epsilon - 1)^{i-j-2}}{\epsilon^{i-j+1}} k(j) + \left( \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \right) k(i-1) - \frac{1}{\epsilon} k(i) \quad (4)$$

and  $k(-1) = 0$  by convention.

(The first sum in (4) is, by usual convention, taken to be 0 if  $i < 2$ .)

*Proof.* We check that a solution (and hence *the* solution) of the system (2) with  $\epsilon_i = \epsilon$  for all  $i \in \mathbb{N}$ , is given by (4).

We proceed by induction on  $i$ . The case  $i = 0$  is easily verified. Assume now that (4) is satisfied by the values of  $\alpha_i$  up to  $i = j - 1$ . Then, from (3) and from the induction hypothesis,

$$\begin{aligned} -\epsilon \alpha_j &= k(j) + \sum_{i=0}^{j-1} \alpha_i (\epsilon + j - i) \\ &= k(j) + \sum_{i=0}^{j-1} \alpha_i (\epsilon + j - 1 - i) + \sum_{i=0}^{j-1} \alpha_i \\ &= k(j) - k(j-1) + \sum_{i=0}^{j-1} \alpha_i. \end{aligned} \quad (5)$$

Substituting (4) in the last term of (5) yields

$$\begin{aligned}
\sum_{i=0}^{j-1} \alpha_i &= \sum_{i=0}^{j-1} \left[ - \sum_{r=0}^{i-2} \frac{(\epsilon - 1)^{i-r-2}}{\epsilon^{i-r+1}} k(r) + \left( \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \right) k(i-1) - \frac{1}{\epsilon} k(i) \right] \\
&= - \sum_{r=0}^{j-3} k(r) \sum_{i=r+2}^{j-1} \frac{(\epsilon - 1)^{i-r-2}}{\epsilon^{i-r+1}} + \left( \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \right) \sum_{i=0}^{j-1} k(i-1) - \frac{1}{\epsilon} \sum_{i=0}^{j-1} k(i) \\
&= - \sum_{r=0}^{j-3} k(r) \sum_{t=0}^{j-r-3} \frac{(\epsilon - 1)^t}{\epsilon^{t+3}} - \frac{1}{\epsilon} k(j-1) + \frac{1}{\epsilon^2} \sum_{i=0}^{j-1} k(i-1) \tag{6}
\end{aligned}$$

$$= \sum_{r=0}^{j-3} \frac{(\epsilon - 1)^{j-r-2}}{\epsilon^{j-r}} k(r) - \frac{1}{\epsilon} k(j-1) + \frac{1}{\epsilon^2} k(j-2) \tag{7}$$

where the last equality is obtained by summing the geometric series which appears in the first sum of equation (6).

Combining (5) and (7), we find

$$\alpha_j = - \sum_{r=0}^{j-3} \frac{(\epsilon - 1)^{j-r-2}}{\epsilon^{j-r+1}} k(r) - \frac{1}{\epsilon^3} k(j-2) + \left( \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \right) k(j-1) - \frac{1}{\epsilon} k(j),$$

which is equivalent to (4) when  $i = j$ . □

There are two special cases of Theorem 2.2 that we will use in particular. When  $\epsilon = 1/2$ , Theorem 2.2 yields:

**Corollary 2.3.** *Every discrete function  $k : \mathbb{N} \rightarrow \mathbb{R}$  can be represented uniquely in the form*

$$k(l) = \sum_{i=0}^{\infty} \alpha_i \left[ i - \frac{1}{2} - l \right]^- \quad \text{for all } l \in \mathbb{N}$$

where

$$\alpha_i = -8 \sum_{j=0}^i (-1)^{i-j} k(j) - 2k(i-1) + 6k(i)$$

for  $i \in \mathbb{N}$ , and  $k(-1) = 0$ . □

Taking  $\epsilon = 1$  in Theorem 2.2, we obtain the following corollary.



**Corollary 2.4.** *Every discrete function  $k : \mathbb{N} \rightarrow \mathbb{R}$  can be represented in the form*

$$k(l) = k(0) + (k(1) - k(0))l + \sum_{i=1}^{\infty} (-k(i-1) + 2k(i) - k(i+1)) [i-l]^-$$

for all  $l \in \mathbb{N}$ . □

The coefficients in the infinite sum that appears in (2.4) are the second-order discrete derivatives of the function  $k$ . Corollary 2.4 can also be found in Fix [6], where it is proved directly by observing that

$$k(l) = \sum_{i=0}^{\infty} k(i) \delta_i(l),$$

where  $\delta_i(l) = 1$  if  $i = l$  and  $\delta_i(l) = 0$  otherwise, and that

$$\delta_i(l) = -[i-1-l]^- + 2[i-l]^- - [i+1-l]^-.$$

In a related, but different setting, Ishikawa [10] establishes a decomposition of (convex) discrete functions into sums of simpler components, where the weights of the components coincide with the second-order discrete derivatives of  $k$ , as in Corollary 2.4 (see also Schlesinger and Flach [24]). Ishikawa's approach focuses on transforming optimization problems for functions of non-binary variables into quadratic pseudo-Boolean ones, and it relies on graph constructions, so that establishing a formal link with our representation theorems is not straightforward.

In contrast, Theorems 2.1-2.2 and their Corollaries 2.3-2.4 are directly stated and proved as functional identities, and they appear to be more general than the results by Fix [6] or by Ishikawa [10] since they are valid for general values of  $\epsilon_i$ . We believe that these results may be of independent interest.

### 3. From representations to quadratizations

In this section we explain how a representation of the type presented in the previous section can be used to construct quadratizations of symmetric pseudo-Boolean functions. First, as mentioned earlier, all results in Section 2 can easily be turned into representation theorems for a symmetric pseudo-Boolean function  $f$ : indeed, if  $k : \{0, 1, \dots, n\} \rightarrow \mathbb{R}$  is such that  $f(x) = k(l)$ ,

where  $l = \sum_{j=1}^n x_j$ , then Equation (2) in Theorem 2.1 implies for instance that  $f$  can be written as

$$f(x) = \sum_{i=0}^n \alpha_i \left[ i - \epsilon_i - \sum_{j=1}^n x_j \right]^- \quad \text{for all } x \in \{0, 1\}^n. \quad (8)$$

Theorem 2.2 and Corollaries 2.3-2.4 can be applied to  $f$  in a similar way.

Another useful observation is that when a coefficient  $\alpha_i$  is non-negative, the corresponding term  $\alpha_i \left[ i - \epsilon_i - \sum_{j=1}^n x_j \right]^-$  in the representation (8) of  $f$  can be quadratized as  $\min_{y_i} \alpha_i y_i (i - \epsilon_i - \sum_{j=1}^n x_j)$ . But this transformation simply does not work if  $\alpha_i$  is negative. The strategy described in this section is to take an expression as given by (8) (or one of its special cases) and add a quantity that is identically-0 and which will result in a final expression that has no terms with negative coefficients. (A similar strategy was introduced by Fix [6].) The following lemma describes three possible such quantities. The first is going to be useful when working with representations of the form given in Corollary 2.4, and the second and third will be useful when working with the representations derived from Corollary 2.3.

**Lemma 3.1.** *Let*

$$E(l) = l(l-1) + 2 \sum_{i=1}^{n-1} [i-l]^-,$$

$$E'(l) = \frac{l(l-1)}{2} + 2 \sum_{\substack{i=2: \\ i \text{ even}}}^n \left[ i - \frac{1}{2} - l \right]^-,$$

and

$$E''(l) = \frac{l(l+1)}{2} + 2 \sum_{\substack{i=1: \\ i \text{ odd}}}^n \left[ i - \frac{1}{2} - l \right]^-.$$

Then,  $E(l) = E'(l) = E''(l) = 0$  for all  $l = 0, 1, \dots, n$ .

*Proof.* First we show that  $E(l)$  is identically-0. We have

$$\begin{aligned}
E(l) &= l(l-1) + 2 \sum_{i=1}^{n-1} [i-l]^- \\
&= l(l-1) + 2 \sum_{i=1}^{l-1} (i-l) \\
&= l(l-1) - 2 \sum_{j=1}^{l-1} j \\
&= l(l-1) - l(l-1) = 0.
\end{aligned}$$

We next show that  $E'(l) = 0$  for all values of  $l$ . Fix  $l$  and note that  $i - \frac{1}{2} - l \leq 0$  if and only if  $i \leq l$ . Hence,

$$\sum_{\substack{i=2: \\ i \text{ even}}}^n \left[ i - \frac{1}{2} - l \right]^- = \sum_{\substack{i=2: \\ i \text{ even}}}^l \left( i - \frac{1}{2} - l \right) = \sum_{\substack{i=2: \\ i \text{ even}}}^l i - \left( \frac{1}{2} + l \right) \left\lfloor \frac{l}{2} \right\rfloor. \quad (9)$$

By considering separately the cases where  $l$  is respectively even or odd, one can conclude that  $E'(l) = 0$  for all  $l = 0, \dots, n$ . For, if  $l = 2r$ , then the expression on the right in equation (9) is  $r/2 - r^2 = -l(l-1)/4$  and, if  $l = 2r + 1$ , it is  $-r/2 - r^2 = -l(l-1)/4$ . The identity  $E''(l) = 0$  (for all  $l$ ) can be proved similarly, or can be deduced from the previous one by observing that, for all  $l = 0, 1, \dots, n$ ,

$$\sum_{i=1}^n \left[ i - \frac{1}{2} - l \right]^- = \sum_{i=1}^l \left( i - \frac{1}{2} - l \right) = -\frac{1}{2}l^2.$$

We then can note that

$$E'(l) + E''(l) = l^2 + 2 \sum_{i=1}^n \left[ i - \frac{1}{2} - l \right]^- = l^2 - l^2 = 0,$$

so that  $E'' = -E' = 0$ . □

We gave a direct, self-contained, proof of Lemma 3.1, but in fact these three identities also follow from Corollary 2.3 and Corollary 2.4. For, if we

apply Corollary 2.4 to the function  $f(x) = (\sum_{r=1}^n x_r) (\sum_{r=1}^n x_r - 1)$ , we see that

$$f(x) = -2 \sum_{i=1}^{n-1} \left[ i - \sum_{r=1}^n x_i \right]^{-},$$

which implies the first identity of Lemma 3.1. Applying Corollary 2.3 to  $f(x)$  shows (after some calculation) that

$$f(x) = -4 \sum_{\substack{i=2: \\ i \text{ even}}}^n \left[ i - \frac{1}{2} - l \right]^{-},$$

giving the second identity (that  $E'$  is identically-0). Applying Corollary 2.3 to the function  $g(x) = (\sum_{r=1}^n x_r) (\sum_{r=1}^n x_r + 1)$  yields the third identity.

## 4. Upper bounds on number of auxiliary variables

### 4.1. Any symmetric function

We first have the following very general result, which provides an explicit construction of a quadratization of any symmetric pseudo-Boolean function, using no more than  $n - 2$  auxiliary variables.

**Theorem 4.1.** *Every symmetric function of  $n$  variables can be quadratized using  $n - 2$  auxiliary variables.*

*Proof.* Using Corollary 2.3, we can write any symmetric function  $f$  as

$$f(x) = -\alpha_0 \left( \frac{1}{2} + \sum_{j=1}^n x_j \right) + \sum_{i=1}^n \alpha_i \left[ i - \frac{1}{2} - \sum_{j=1}^n x_j \right]^{-}.$$

Let  $\alpha_r = \min\{\alpha_i : i \text{ even}, i \geq 2\}$  and  $\alpha_s = \min\{\alpha_i : i \text{ odd}\}$ . Now add to  $f$  the expression

$$-\frac{\alpha_r}{2} E' \left( \sum_{j=1}^n x_j \right) - \frac{\alpha_s}{2} E'' \left( \sum_{j=1}^n x_j \right),$$

which is identically-0. This results in an expression for  $f$  of the form

$$f(x) = a_0 + a_1 \sum_{j=1}^n x_j + a_2 \sum_{1 \leq i < j \leq n} x_i x_j + \sum_{i=1}^n \beta_i \left[ i - \frac{1}{2} - \sum_{j=1}^n x_j \right]^{-},$$

where, for each  $i$ , if  $i$  is even,  $\beta_i = \alpha_i - \alpha_r \geq 0$ , and if  $i$  is odd,  $\beta_i = \alpha_i - \alpha_s \geq 0$ . So all the coefficients  $\beta_i$  are non-negative. Furthermore,  $\beta_r = \beta_s = 0$ , so we have an expression for  $f$  involving no more than  $n - 2$  positive coefficients  $\beta_i$ . Then,

$$g(x, y) = a_0 + a_1 \sum_{j=1}^n x_j + a_2 \sum_{1 \leq i < j \leq n} x_i x_j + \sum_{\substack{i=1: \\ i \neq r, s}}^n \beta_i y_i \left( i - \frac{1}{2} - \sum_{j=1}^n x_j \right)$$

is a quadratization of  $f$  involving at most  $n - 2$  auxiliary variables.  $\square$

(A construction in [6] shows an upper bound of  $n - 1$ . This is obtained by adding a multiple of  $E(\sum_{r=1}^n x_r)$  to each term in the expression from Corollary 2.4, rather than to the expression as a whole, resulting in more complex quadratizations.)

Notice that the quadratization in the proof of Theorem 4.1 is  $y$ -linear, so we have in fact shown:

**Theorem 4.2.** *Every symmetric function of  $n$  variables has a  $y$ -linear quadratization involving at most  $n - 2$  auxiliary variables.*

Furthermore, these quadratizations are also symmetric in the  $x$ -variables. Not every quadratization of a symmetric function must itself be symmetric in the original variables. For example, consider the negative monomial

$$- \prod_{i=1}^n x_i = -x_1 x_2 \cdots x_n.$$

As we have seen, this has the quadratization  $y \left( n - 1 - \sum_{j=1}^n x_j \right)$ , which is symmetric. However, it also has the quadratization

$$(n - 2)x_n y - \sum_{i=1}^{n-1} x_i (y - \bar{x}_n),$$

where  $\bar{x}_n = 1 - x_n$ , which is not symmetric in the  $x$ -variables.

#### 4.2. Monomials

The quadratization of monomials (positive and negative) has been fairly well-studied. The *standard quadratization* of the negative monomial

$$f(x) = - \prod_{i=1}^n x_i = -x_1 x_2 \cdots x_n,$$

is

$$s_n(x_1, x_2, \dots, x_n, y) = y \left( n - 1 - \sum_{j=1}^n x_j \right).$$

(A related paper by the present authors [1] gives a complete characterization of all the quadratizations of negative monomials involving one auxiliary variable and this is, in a sense, one of the simplest.) If we apply Corollary 2.3 to the negative monomial, noting that  $k(i) = 0$  for  $i < n$  and  $k(n) = -1$ , we obtain the representation

$$f(x) = 2 \left[ n - \frac{1}{2} - \sum_{r=1}^n x_r \right]^{-},$$

which immediately leads to the quadratization

$$h = 2y \left( n - \frac{1}{2} - \sum_{r=1}^n x_r \right),$$

only slightly different from the standard one. We could, instead, apply Corollary 2.4, which would show that  $f(x) = [n - 1 - \sum_{r=1}^n x_r]^{-}$ , from which we obtain the standard quadratization.

As we noted earlier, the best known result (smallest number of auxiliary variables) for positive monomials is that they can be quadratized using  $\lfloor \frac{n-1}{2} \rfloor$  auxiliary variables. This was shown by Ishikawa [11]. We can see that this many auxiliary variables suffice by using our representation theorem, Corollary 2.3, together with the argument given in the proof of Theorem 4.1.

**Theorem 4.3.** *The positive monomial  $P = \prod_{i=1}^n x_i$  can be quadratized using  $\lfloor \frac{n-1}{2} \rfloor$  auxiliary variables.*

*Proof.* Consider first the case where  $n$  is even. By Corollary 2.3, noting that  $k(i) = 0$  for  $i < n$  and  $k(n) = 1$ , we have  $P = -2 \left[ n - \frac{1}{2} - l \right]^-$  where  $l = \sum_{r=1}^n x_r$ . By Lemma 3.1,

$$\begin{aligned} P &= -2 \left[ n - \frac{1}{2} - l \right]^- + E'(l) \\ &= \frac{l(l-1)}{2} + \sum_{\substack{i=2: \\ i \text{ even}}}^{n-2} 2 \left[ i - \frac{1}{2} - l \right]^- \\ &= \sum_{1 \leq i < j \leq n} x_i x_j + \min_y \sum_{\substack{i=2: \\ i \text{ even}}}^{n-2} 2y_i \left( i - \frac{1}{2} - l \right). \end{aligned}$$

This provides the required quadratization using  $\frac{n}{2} - 1 = \lfloor \frac{n-1}{2} \rfloor$  new variables.

When  $n$  is odd, one similarly derives the following from Lemma 3.1:

$$\begin{aligned} P &= -2 \left[ n - \frac{1}{2} - l \right]^- + E''(l) \\ &= \sum_{i=1}^n x_i + \sum_{1 \leq i < j \leq n} x_i x_j + \min_y \sum_{\substack{i=1: \\ i \text{ odd}}}^{n-2} 2y_i \left( i - \frac{1}{2} - l \right). \end{aligned}$$

□

This quadratization of  $P$  requires the same number of auxiliary variables as Ishikawa's construction. Both quadratizations are, in fact, identical when  $n$  is even, but appear to be different when  $n$  is odd.

Note that an alternative approach to the case of odd  $n$  would be as follows. Write

$$P = \prod_{i=1}^{n-1} x_i - \prod_{i=1}^{n-1} x_i \bar{x}_n,$$

where  $\bar{x}_n = 1 - x_n$ . The first term can now be quadratized using  $\frac{n-1}{2} - 1$  new variables (since it contains an even number of variables), and the second term, viewed as a negative monomial in  $x_1, \dots, x_{n-1}, \bar{x}_n$ , has a standard quadratization requiring one further auxiliary variable. Thus, this leads again to a quadratization of  $P$  with  $\frac{n-1}{2} = \lfloor \frac{n-1}{2} \rfloor$  new variables. This quadratization is also different from Ishikawa's.

### 4.3. $t$ -out-of- $n$ and exact- $t$ functions

Consider now the  $t$ -out-of- $n$  function defined by:

$$f_{t,n}(x) = 1 \quad \text{if and only if} \quad \sum_{i=1}^n x_i \geq t.$$

The basic Boolean functions  $\text{And}_n(x) := \prod_{i=1}^n x_i$  (a positive monomial) and  $\text{Or}_n(x) := 1 - \prod_{i=1}^n (1 - x_i)$  are examples of  $t$ -out-of- $n$  functions with  $t = n$  and  $t = 1$ , respectively. Another well-known example is the *majority* function given by:

$$\text{Maj}_n(x) := \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq \lceil n/2 \rceil, \\ 0 & \text{otherwise,} \end{cases}$$

which breaks ties in favor of ones when  $n$  is even. In this case,  $t = \lceil n/2 \rceil$ .

**Corollary 4.4.** *The  $t$ -out-of- $n$  function  $f_{t,n}$  can be quadratized using  $\lceil n/2 \rceil$  auxiliary variables.*

*Proof.* From Corollary 2.3,  $f_{t,n}$  can be represented in the form

$$f_{t,n}(x) = \sum_{i=0}^n \alpha_i \left[ i - \frac{1}{2} - \sum_{j=1}^n x_j \right]^- \quad (10)$$

where  $\alpha_i = 0$  when  $i < t$ ,  $\alpha_t = -2$ , and  $\alpha_i = 4(-1)^{i-t-1}$  when  $i > t$ .

Since the terms of  $f_{t,n}$  alternate in sign when  $i \geq t$ , we can again use Lemma 3.1 to make all coefficients non-negative by adding either  $2E'(l)$  or  $2E''(l)$  to (10), depending on the parity of  $t$ . The resulting expression has  $\lceil n/2 \rceil$  positive coefficients, and its remaining coefficients are zero. Thus, it can be quadratized with  $\lceil n/2 \rceil$  auxiliary variables.  $\square$

A related function is the *exact- $t$*  (out of  $n$ ) function, defined as  $f_{t,n}^{\bar{=}}(x) = 1$  if and only if the Hamming weight of  $x$  equals  $t$ . Using Corollary 2.3 again, we have that  $f_{t,n}^{\bar{=}}$  can be represented in the form given in (10) with  $\alpha_i = 0$  when  $i < t$ ,  $\alpha_t = -2$ ,  $\alpha_{t+1} = 6$ , and  $\alpha_i = 0$  when  $i > t + 1$ . Depending on the parity of  $t$ , we add  $E'(l)$  or  $E''(l)$  to (10) to obtain an expression with  $\lceil n/2 \rceil$  positive coefficients, which can then be quadratized with  $\lceil n/2 \rceil$  auxiliary variables. We have just proved the following.



**Corollary 4.5.** *The exact- $t$  function  $f_{t,n}^-$  can be quadratized using  $\lfloor n/2 \rfloor$  auxiliary variables.  $\square$*

The positive monomial and the  $\text{And}_n$  Boolean function are also special cases of exact- $t$  functions, both with  $t = n$ . It is apparent from the argument leading to Corollary 4.5 that the reason the positive monomial (and hence, the  $\text{And}_n$  function) requires  $\lfloor \frac{n-1}{2} \rfloor$  auxiliary variables instead of  $\lfloor n/2 \rfloor$  is precisely because  $t = n$ .

#### 4.4. Parity and its complement

The *parity* function is the (pseudo-)Boolean function  $\Pi(x)$  such that  $\Pi(x) = 1$  if the Hamming weight of  $x$  is odd, and  $\Pi(x) = 0$  otherwise. To derive a quadratization of this function, we will use Corollary 2.4 rather than Corollary 2.3, and will make use of a variant of the argument given to establish Theorem 4.1. By Corollary 2.4, we can see that  $\Pi$  has the representation

$$\Pi(x) = \sum_{j=1}^n x_j + 2 \sum_{i=1}^{n-1} (-1)^{i-1} \left[ i - \sum_{j=1}^n x_j \right]^-. \quad (11)$$

Let  $E(l)$  be as in Lemma 3.1. By adding  $E(\sum_{j=1}^n x_j)$  to this representation of  $\Pi$ , we obtain a representation with non-negative coefficients, which leads to a quadratization with  $m = \lfloor n/2 \rfloor$  auxiliary variables:  $\Pi(x) = \min_{y \in \{0,1\}^m} g(x, y)$  where

$$g(x, y) = 2 \sum_{i < j} x_i x_j + \sum_{j=1}^n x_j + 4 \sum_{\substack{i=1: \\ i \text{ odd}}}^{n-1} y_i \left( i - \sum_{j=1}^n x_j \right).$$

(The terms with coefficient  $-2$  in the expansion (11) disappear on the addition of  $E$ .)

The complement,  $\bar{\Pi}$  of  $\Pi$  can be represented as

$$\bar{\Pi}(x) = 1 - \sum_{j=1}^n x_j + 2 \sum_{i=1}^{n-1} (-1)^i \left[ i - \sum_{j=1}^n x_j \right]^-,$$

so, by adding  $E(\sum_{j=1}^n x_j)$ , to eliminate negative coefficients, we arrive at the following quadratization involving  $m = \lfloor \frac{n-1}{2} \rfloor$  auxiliary variables:

$$g'(x, y) = 1 + 2 \sum_{i < j} x_i x_j - \sum_{j=1}^n x_j + 4 \sum_{\substack{i=2: \\ i \text{ even}}}^{n-1} y_i \left( i - \sum_{j=1}^n x_j \right).$$

So we conclude the following:

**Theorem 4.6.** *The parity function of  $n$  variables has a  $y$ -linear quadratization involving  $\lfloor \frac{n}{2} \rfloor$  auxiliary variables, and its complement has a  $y$ -linear quadratization involving  $\lfloor \frac{n-1}{2} \rfloor$  auxiliary variables.*

## 5. Lower bounds on the number of auxiliary variables

### 5.1. Generic lower bounds

The following result is inspired by (but is different and does not follow from) a transformation given in Siu, Roychowdhury and Kailath [25], in the framework of the representation of Boolean functions by threshold circuits. This result relates quadratizations of arbitrary (possibly non-symmetric) pseudo-Boolean functions to the quadratization of symmetric functions on a larger number of variables. We will then use a lower bound result from [1] in order to obtain a lower bound result for symmetric functions.

**Lemma 5.1.** *Suppose that  $n, m$  are positive integers and suppose that every symmetric pseudo-Boolean function  $F(z)$  of  $N = 2^n - 1$  variables (that is, every symmetric function  $F : \{0, 1\}^{2^n - 1} \rightarrow \mathbb{R}$ ) has an  $m$ -quadratization. Then every (arbitrary) pseudo-Boolean function  $f(x)$  on  $\{0, 1\}^n$  also has an  $m$ -quadratization.*

*Proof.* Let  $f(x)$  be an arbitrary pseudo-Boolean function of  $n$  variables. We are going to construct a sequence of four functions  $k, F, G, g$ , such that  $g$  is a quadratization of  $f$ . For this purpose, let  $N = 2^n - 1$ .

1. Let  $k : \{0, 1, \dots, N\} \rightarrow \mathbb{R}$  be defined as follows:  $k(w) := f(x)$  where  $x$  is the binary representation of  $w$ , that is,  $w = \sum_{i=1}^n 2^{i-1} x_i$ .
2. Let  $F$  be the symmetric pseudo-Boolean function of  $N$  variables defined by: for all  $z \in \{0, 1\}^N$ ,  $F(z) := k(|z|)$ , where  $|z|$  is the Hamming weight of  $z$ . (This defines  $F$  completely, given that it is symmetric.)
3. Let  $G(z, y)$  be an arbitrary quadratization of  $F(z)$  using  $m$  auxiliary variables. (The hypothesis of the theorem is that such quadratizations exist.)
4. Finally, let  $g(x, y)$  be the pseudo-Boolean function on  $\{0, 1\}^{n+m}$  that is obtained by identifying each of the variables  $z_{2^j-1}, z_{2^j-1+1}, \dots, z_{2^j-1}$  with  $x_j$  in  $G(z, y)$ , for  $j = 1, 2, \dots, n$ ; that is,

$$g(x_1, x_2, x_3, \dots, x_n, y) := G(x_1, x_2, x_2, x_3, x_3, x_3, x_3, \dots, x_n, \dots, x_n, y).$$

We claim that  $g(x, y)$  is a quadratization of  $f$ . Indeed,  $g$  is clearly quadratic, because  $G$  is. Moreover, for every point  $x \in \{0, 1\}^n$ ,

$$\begin{aligned} & \min\{g(x, y) : y \in \{0, 1\}^m\} \\ &= \min\{G(x_1, x_2, x_2, x_3, x_3, x_3, x_3, \dots, x_n, y) : y \in \{0, 1\}^m\} \end{aligned} \quad (12)$$

$$= F(x_1, x_2, x_2, x_3, x_3, x_3, x_3, \dots, x_n) \quad (13)$$

$$= k \left( \sum_{i=1}^n 2^{i-1} x_i \right) \quad (14)$$

$$= f(x) \quad (15)$$

(where equality (12) is by definition of  $g$ , (13) is by definition of  $G$ , (14) is by definition of  $F$ , and (15) is by definition of  $k$ ).  $\square$

We will now make use of the following result from [1].

**Theorem 5.2.** *There are pseudo-Boolean functions of  $n$  variables for which any quadratization must involve at least  $\Omega(2^{n/2})$  auxiliary variables.*

To be more concrete, the analysis in [1] implies that for any  $n \geq 8$ , there is a pseudo-Boolean function on  $n$  variables for which any quadratization will require at least  $2^{n/2}/8$  auxiliary variables.

This leads to the following lower bound result for symmetric functions.

**Theorem 5.3.** *There exist symmetric functions of  $n$  variables for which any quadratization must involve at least  $\Omega(\sqrt{n})$  auxiliary variables.*

*Proof.* Lemma 5.1 shows that, if every symmetric function  $F(z)$  on  $\{0, 1\}^N$ , with  $N = 2^n - 1$ , has an  $m$ -quadratization, then every (arbitrary) function  $f(x)$  on  $\{0, 1\}^n$  also has an  $m$ -quadratization. From Theorem 5.2, it follows that some symmetric functions on  $N$  variables must need  $\Omega(\sqrt{N})$  auxiliary variables in every quadratization.  $\square$

We also have a similar lower bound result for  $y$ -linear quadratizations. It rests on the following result from [1]:

**Theorem 5.4.** *There are pseudo-Boolean functions of  $n$  variables for which any  $y$ -linear quadratization must involve at least  $\Omega(2^n/n)$  auxiliary variables.*

We then have the following.

**Theorem 5.5.** *There exist symmetric functions of  $n$  variables for which any  $y$ -linear quadratization must involve at least  $\Omega(n/\log n)$  auxiliary variables.*

*Proof.* The proof is similar to the previous one: it suffices to observe that when  $G(z, y)$  is  $y$ -linear, then so is  $g(x, y)$ , and to rely on the generic lower bound  $\Omega(2^n/n)$  of Theorem 5.4 for the number of auxiliary variables required in every  $y$ -linear quadratization of certain pseudo-Boolean functions.  $\square$

Note that the lower bound in Theorem 5.5 for the number of auxiliary variables in  $y$ -linear quadratizations comes within a factor  $O(\log n)$  of the upper bound of  $n - 2$  from Theorem 4.2.

### 5.2. A lower bound for the parity function

The results just obtained prove the existence of symmetric pseudo-Boolean functions which require a significant number of auxiliary variables to quadratize. Specifically, there exist functions needing  $\Omega(\sqrt{n})$  auxiliary variables in any quadratization, and functions needing  $\Omega(n/\log n)$  auxiliary variables in any  $y$ -linear quadratization. Those results do not, however, explicitly exhibit particular such functions. We next give a concrete example of a function which needs a significant number of auxiliary variables in any  $y$ -linear quadratization.

**Theorem 5.6.** *Every  $y$ -linear quadratization of the parity function on  $n$  variables must involve at least  $\Omega(\sqrt{n})$  auxiliary variables.*

*Proof.* Let  $g(x, y)$  be an arbitrary  $y$ -linear quadratization of the parity function. Then it can be written as

$$g(x, y) = q(x) + \sum_{i=1}^m y_i(\ell_i(x) - b_i) \quad (16)$$

where  $q(x)$  is quadratic, and  $\ell_1(x), \dots, \ell_m(x)$  are linear functions of  $x$  only.

For each  $i \in [m] = \{1, 2, \dots, m\}$ , consider the regions

$$R_i^+ = \{x \in \mathbb{R}^n : \ell_i(x) \geq b_i\}, \quad R_i^- = \{x \in \mathbb{R}^n : \ell_i(x) \leq b_i\},$$

which are closed half-spaces defined by the linear functions  $\ell_i$ . For each  $S \subseteq [m]$ , let  $R_S$  denote the region  $R_S = (\bigcap_{i \in S} R_i^-) \cap (\bigcap_{i \notin S} R_i^+)$ . This is one of the ‘cells’ into which the  $m$  hyperplanes defining the linear functions  $\ell_i$  partition  $\mathbb{R}^n$ .

On every cell  $R_S$ , the function  $f(x) = \min\{g(x, y) : y \in \{0, 1\}^m\}$  is quadratic. Indeed, for  $x \in R_S$ , we have

$$\min\{g(x, y) : y \in \{0, 1\}^m\} = q(x) + \sum_{i \in S} (\ell_i(x) - b_i).$$

We now use a result from Saks [23] and Impagliazzo, Paturi and Saks [9] (which was used to obtain lower bounds on the size of threshold circuits representing the parity function). Let us say that a set of hyperplanes *slices* all  $r$ -dimensional subcubes of the Boolean hypercube  $\{0, 1\}^n$  if for each subcube (or face) of  $\{0, 1\}^n$  of dimension  $r$ , there are two vertices of the subcube that lie on opposite sides of one of these hyperplanes. Then (Proposition 3.82 of [23]), if a set of  $m$  hyperplanes slices all  $r$ -dimensional subcubes, we have  $m > \sqrt{n/(r+1)} - 1$ . In particular, therefore, any set of hyperplanes that slices every 3-dimensional subcube of  $\{0, 1\}^n$  must contain more than  $\sqrt{n/4} - 1$  planes. Suppose the hyperplanes defined by the linear functions  $\ell_i$  do *not* slice all 3-dimensional subcubes. Then there would be some cell  $R_S$  containing a subcube of dimension 3. The parity function restricted to that subcube would then be equal to the quadratic expression  $q(x) + \sum_{i \in S} (\ell_i(x) - b_i)$ . However, it is well-known (see, for instance [23, 18, 26]) that the parity function on a subcube of dimension  $r$  cannot be represented as a pseudo-Boolean function of degree less than  $r$  (and it cannot even be represented as the sign of a pseudo-Boolean function of degree less than  $r$ ). So, we would then have a quadratic, degree-2, representation of parity on a cube of dimension 3, which is not possible. It follows, therefore, that the set of hyperplanes in question must slice all 3-dimensional subcubes and therefore has size  $m > \sqrt{n/4} - 1$ .  $\square$

## 6. Conclusions

In this paper, we have studied the number of auxiliary variables required in quadratizations (and  $y$ -linear quadratizations) of symmetric pseudo-Boolean function. We have presented explicit general constructions of quadratizations, via special types of representations of the functions. This shows that every such function can be quadratized (with a  $y$ -linear quadratization, symmetric in the original variables) using at most  $n - 2$  auxiliary variables. We investigated in more detail the quadratizations of special functions (monomials,  $t$ -out-of- $n$ , exact- $t$ , and parity functions), where it was possible to obtain quadratizations using significantly fewer than  $n - 2$  auxiliary variables. By

drawing on a general result from our related paper [1] and establishing a connection between quadratizations of general functions and of symmetric functions on a related number of variables, we showed that there exist symmetric functions requiring  $\Omega(\sqrt{n})$  auxiliary variables in any quadratization, and that  $y$ -linear quadratization can require  $\Omega(n/\log n)$  variables. It would clearly be of interest to close the gaps between these lower bounds and the linear upper bound. We established, further, that any  $y$ -linear representation of the parity function needs  $\Omega(\sqrt{n})$  auxiliary variables. An open question is to determine whether a similar (or better) lower bound can be obtained for any (not necessarily  $y$ -linear) quadratization of this, or another specific, symmetric function. For instance, any example of a symmetric function where a non  $y$ -linear quadratization needs fewer variables than the  $y$ -linear ones would be of interest, as would be any non constant lower bound on the number of auxiliary variables for positive monomials. Furthermore, the number of positive quadratic terms in any known quadratization of the positive monomial is at least  $n - 1$ , but no lower bound on such quantity has been found so far. Settling this question is also of great interest, as it is related to the quality of relaxations based on quadratizations for PBO problems.

**Acknowledgements.** We thank György Turán for several discussions and references on Boolean circuits for symmetric functions, and we thank the reviewers for their helpful comments on a first version of this paper. The second author thanks the National Science Foundation (Grant IIS-1161476) for partial support. The third author was partially funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (grant P7/36) and by a sabbatical grant from FNRS. The fourth author thanks the joint CAPES (Brazil)/Fulbright (USA) fellowship process BEX-2387050/15061676 for partial support.

## References

- [1] M. Anthony, E. Boros, Y. Crama and M. Gruber, Quadratic reformulations of nonlinear binary optimization problems, Working paper, HEC Management School, University of Liege (2015). <http://hdl.handle.net/2268/184526>
- [2] E. Boros and A. Gruber, On quadratization of pseudo-Boolean functions, *International Symposium on Artificial Intelligence and Mathematics* (ISAIM 2012), Fort Lauderdale, Florida, USA, January 9-11 (2012).

- [3] E. Boros and P.L. Hammer, Pseudo-Boolean optimization, *Discrete Applied Mathematics* 123 (2002) 155–225.
- [4] Y. Boykov, O. Veksler and R. Zabih, Fast approximate energy minimization via graph cuts, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (2001) 1222–1239.
- [5] Y. Crama and P.L. Hammer, *Boolean Functions: Theory, Algorithms, and Applications*, Cambridge University Press, New York, N.Y., 2011.
- [6] A. Fix, Reductions for rewriting QPBFs with spanning trees, Unpublished notes, 2011.
- [7] A. Fix, A. Gruber, E. Boros and R. Zabih, A graph cut algorithm for higher-order Markov random fields, in: *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV)* (2011) pp. 1020–1027.
- [8] D. Freedman and P. Drineas, Energy minimization via graph cuts: Settling what is possible, in: *IEEE Conference on Computer Vision and Pattern Recognition* (2) (2005) pp. 939–946.
- [9] R. Impagliazzo, R. Paturi, M. E. Saks, Size-depth tradeoffs for threshold circuits, *SIAM Journal on Computing* 26(3) (1997) 693–707.
- [10] H. Ishikawa, Exact optimization for Markov random fields with convex priors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(10) (2003) 1333–1336.
- [11] H. Ishikawa, Transformation of general binary MRF minimization to the first-order case, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(6) (2011) 1234–1249.
- [12] J.H. Kappes et al., A comparative study of modern inference techniques for discrete energy minimization problems, in: *IEEE Conference on Computer Vision and Pattern Recognition CVPR’13* (2013) pp. 1328–1335.
- [13] P. Kohli, L. Ladický and P.H.S. Torr, Robust higher order potentials for enforcing label consistency, *International Journal of Computer Vision* 82 (2009) 302–324.

- [14] P. Kohli, M. Pawan Kumar and P.H.S. Torr,  $P^3$  & beyond: Move making algorithms for solving higher order functions *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (9) (2009) 1645–1656.
- [15] V. Kolmogorov and C. Rother, Minimizing non-submodular functions with graph cuts - a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007) 1274–1279.
- [16] V. Kolmogorov and R. Zabih, What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(2) (2004) 147–159.
- [17] J. Lee and S. Leyffer, Sven (eds.), *Mixed Integer Nonlinear Programming*, The IMA Volumes in Mathematics and its Applications, Vol. 154, Springer, 2012.
- [18] M. Minsky and S. Papert, *Perceptrons*, MIT Press, Cambridge, MA., 1969. (Expanded edition 1988.)
- [19] S. Ramalingam, Ch. Russell, L. Ladický and Ph.H.S. Torr, Efficient minimization of higher order submodular functions using monotonic Boolean functions, arXiv:1109.2304v1, 2011.
- [20] I.G. Rosenberg, Reduction of bivalent maximization to the quadratic case, *Cahiers du Centre d’Etudes de Recherche Opérationnelle* 17 (1975) 71–74.
- [21] C. Rother, P. Kohli, W. Feng and J. Jia, Minimizing sparse higher order energy functions of discrete variables, in: *IEEE Conference on Computer Vision and Pattern Recognition* (2009) pp. 1382–1389.
- [22] C. Rother, V. Kolmogorov, V. Lempitsky and M. Szummer, Optimizing binary MRFs via extended roof duality, in: *IEEE Conference on Computer Vision and Pattern Recognition* (2007) pp. 1–8.
- [23] M. Saks, Slicing the hypercube, in: K. Walker, ed., *Surveys in Combinatorics*, Cambridge University Press, Cambridge, 1993, pp. 211–255.
- [24] D. Schlesinger and B. Flach, Transforming an arbitrary minsum problem into a binary one, Report TUD-F106-0, Technical University Dresden, (2006). [citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.68.2846](http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.68.2846).



- [25] K.-Y. Siu, V. Roychowdhury and T. Kailath, *Discrete Neural Computation: A Theoretical Foundation*, Prentice Hall, NJ, 1995.
- [26] C. Wang and A.C. Williams, The threshold order of a Boolean function, *Discrete Applied Mathematics* 31 (1991) 51–69.