Generalized Roof Duality for Pseudo-Boolean Optimization

Fredrik Kahl Petter Strandmark
Centre for Mathematical Sciences, Lund University, Sweden
{fredrik,petter}@maths.lth.se

Abstract

The number of applications in computer vision that model higher-order interactions has exploded over the last few years. The standard technique for solving such problems is to reduce the higher-order objective function to a quadratic pseudo-boolean function, and then use roof duality for obtaining a lower bound. Roof duality works by constructing the tightest possible lower-bounding submodular function, and instead of optimizing the original objective function, the relaxation is minimized.

We generalize this idea to polynomials of higher degree, where quadratic roof duality appears as a special case. Optimal relaxations are defined to be the ones that give the maximum lower bound. We demonstrate that important properties such as persistency still hold and how the relaxations can be efficiently constructed for general cubic and quartic pseudo-boolean functions. From a practical point of view, we show that our relaxations perform better than state-of-the-art for a wide range of problems, both in terms of lower bounds and in the number of assigned variables.

1. Introduction

Consider a pseudo-boolean function $f : \mathcal{B}^n \to \mathbb{R}$ where $\mathcal{B} = \{0, 1\}$ and suppose f is represented by a multilinear polynomial of the form

$$f(\boldsymbol{x}) = \sum_i a_i x_i + \sum_{i<j} a_{ij} x_i x_j + \sum_{i<j<k} a_{ijk} x_i x_j x_k + \ldots,$$

of $\mathrm{degree}(f) = m$. In this paper, we are interested in the following optimization problem:

$$\min_{\boldsymbol{x} \in \mathcal{B}^n} f(\boldsymbol{x}). \tag{1}$$

One of the major achievements over the last decade is the development of efficient optimization methods for minimizing such functions, especially, for quadratic pseudo-boolean functions. Application problems in computer vision that can be turned into such an optimization problem abound. For example, state-of-the-art methods for stereo, segmentation and image denoising are often formulated as the inference of the maximum a posteriori (MAP) estimate in a Markov Random Field (MRF) and thanks to the Hammersley-Clifford theorem, such problems can be formulated as energy minimization problems where the energy function is given by a pseudo-boolean function.

In general, the minimization problem in (1) is NP-hard so approximation algorithms are necessary. For the quadratic case ($m = 2$), one of the most popular and successful approaches is based on the roof duality bound [2, 11]. The primary focus of this paper is to generalize the roof duality framework for higher-order pseudo-boolean functions. Our main contributions are (i) how one can define a general bound for any order (for which the quadratic case is a special case) and (ii) how one can efficiently compute solutions that attain this bound in polynomial time. Naturally, as this is a fundamental problem with many important applications, our results rely on many previous significant contributions.

Related work. Graph cuts is by now a standard tool for many vision problems, in particular, for the minimization of quadratic and cubic submodular pseudo-boolean functions [3, 12]. The same technique can be used for non-submodular functions in order to compute a lower bound [2].

In recent years, there has been an increasing interest in higher-order models and approaches for minimizing the corresponding energies. For example, in [15], approximate belief propagation is used with a learned higher-order MRF model for image denoising. Similarly, in [4], an MRF model is learned for texture restoration, but the model is restricted to submodular energies which can be optimized exactly with graph cuts. Curvature regularization requires higher-order models [20, 22]. Even global potentials defined over all variables in the MRF have been considered, e.g., in [18] for ensuring connectedness, in [14] to model co-occurrence statistics of objects. Another state-of-the-art example is [24] where second-order surface priors are used for stereo reconstruction. The optimization strategies rely on dual decomposition [13, 21], move-making algorithms [9, 16], linear programming [23], belief propagation [15] and, of course, graph cuts.

The development of models, optimization techniques and applications in this area has indeed been immense. The inspiration for our work comes primarily from three different sources. First of all, as graph cuts is considered to be state-of-the-art for quadratic pseudo-boolean polynomials, reductions of higher-order polynomials ($m > 2$) have been explored, e.g., [17, 5, 19, 8]. Then, there exist several suggestions for generalizations of roof duality for higher-order polynomials. In [17], a roof duality framework is presented based on reduction, but at the same time, the authors note that their roof duality bound depends on which reductions are applied. In [10], bisubmodular relaxations are proposed as a generalization for roof duality, but no method is given for constructing or minimizing such relaxations. Finally, the complete characterization of submodular functions up to degree $m = 4$ that can be reduced to the quadratic case is instrumental to our work, see [25]. This puts natural limitations on any reduction framework.

Our framework builds on [10] using submodular relaxations directly on higher-order terms. We define *optimal relaxations* to be those that give the tightest lower bound. As an example, consider the problem of minimizing the following cubic polynomial f over $\mathcal{B}^3$:

$$f(\boldsymbol{x}) = -2x_1 + x_2 - x_3 + 4x_1x_2 + 4x_1x_3 - 2x_2x_3 - 2x_1x_2x_3. \quad (2)$$

The standard reduction scheme [8] would use the identity $-x_1x_2x_3 = \min_{z\in\mathcal{B}} z(2 - x_1 - x_2 - x_3)$ to obtain a quadratic minimization problem with one auxiliary variable z. Roof duality gives a lower bound of $f_{\text{min}} \geq -3$, but it does not reveal how to assign any of the variables in $\boldsymbol{x}$. However, there are many possible reduction schemes from which one can choose. Another possibility is $-x_1x_2x_3 = \min_{z\in\mathcal{B}} z(-x_1 + x_2 + x_3) - x_1x_2 - x_1x_3 + x_1$. For this reduction, the roof duality bound is tight and the optimal solution $\boldsymbol{x}^* = (0, 1, 1)$ is obtained (see Section 4). This simple example illustrates two facts: (i) different reductions lead to different lower bounds and (ii) it is not an obvious matter how to choose the optimal reduction. Choosing sub-optimally between a fixed set of possible reductions was proposed recently in [7].

2. Submodular Relaxations

Consider the optimization problem in (1) where f has degree m. In this paper, we will primarily investigate the cases when $m = 2$, $m = 3$ or $m = 4$. Without loss of generality, it is assumed that $f(\mathbf{0}) = 0$.

By enlarging the domain, we will relax the problem and look at the following tractable problem:

$$\min_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}^{2n}} g(\boldsymbol{x}, \boldsymbol{y}), \quad (3)$$

where $g : \mathcal{B}^{2n} \to \mathbb{R}$ is a submodular function that satisfies

$$g(\boldsymbol{x}, \bar{\boldsymbol{x}}) = f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{B}^n. \quad (4)$$

Existence of such relaxations follows from the fact that polynomials with only non-positive coefficients are submodular and it is easy to construct a g satisfying (4) using only non-positive coefficients (except for possibly linear terms).

Let f_{min} denote the unknown minimum value of f, that is, $f_{\text{min}} = \min f(\boldsymbol{x})$. Ideally, we would like $g(\boldsymbol{x}, \boldsymbol{y}) \geq f_{\text{min}}$ for all points $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{B}^{2n}$. This is evidently not possible in general. However, one could try to maximize the lower bound of g, $\max \min_{\boldsymbol{x},\boldsymbol{y}} g(\boldsymbol{x}, \boldsymbol{y})$, hence,

$$\begin{aligned} \max_{g} \quad & \ell \\ \text{such that} \quad & g(\boldsymbol{x}, \boldsymbol{y}) \geq \ell, \quad \forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{B}^{2n}. \end{aligned} \quad (5)$$

Here the domain ranges over all possible multilinear polynomials g (of fixed degree) that satisfy (4) and are submodular. A relaxation g that provides the maximum lower bound will be called *optimal*. As we shall prove, when $m = 2$, the lower bound coincides with the roof duality bound and therefore this maximum lower bound will be referred to as *generalized roof duality*.

Notation. For a point $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{B}^n$, denote $\bar{\boldsymbol{x}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n) = (1 - x_1, 1 - x_2, \ldots, 1 - x_n)$. As standard, $\boldsymbol{x} \wedge \boldsymbol{y}$ and $\boldsymbol{x} \vee \boldsymbol{y}$ mean element-wise min and max, respectively. Let $\mathcal{S}^n = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{B}^{2n} \,|\, (x_i, y_i) \neq (1, 1), i = 1, \ldots, n\}$. For $(\boldsymbol{x}_1, \boldsymbol{y}_1) \in \mathcal{S}^n$ and $(\boldsymbol{x}_2, \boldsymbol{y}_2) \in \mathcal{S}^n$, the operators $\sqcap$ and $\sqcup$ are defined by

$$\begin{aligned} (\boldsymbol{x}_1, \boldsymbol{y}_1) \sqcap (\boldsymbol{x}_2, \boldsymbol{y}_2) &= (\boldsymbol{x}_1 \wedge \boldsymbol{x}_2, \boldsymbol{y}_1 \wedge \boldsymbol{y}_2) \\ (\boldsymbol{x}_1, \boldsymbol{y}_1) \sqcup (\boldsymbol{x}_2, \boldsymbol{y}_2) &= \Big((\boldsymbol{x}_1 \vee \boldsymbol{x}_2) \wedge (\overline{\boldsymbol{y}_1 \vee \boldsymbol{y}_2}), \\ &\qquad (\boldsymbol{y}_1 \vee \boldsymbol{y}_2) \wedge (\overline{\boldsymbol{x}_1 \vee \boldsymbol{x}_2})\Big). \end{aligned} \quad (6)$$

It is easy to check that the resulting points belong to $\mathcal{S}^n$. Further, for a scalar a, the positive and negative parts will be denoted a^+ and a^-, where $a^+ = \max(a, 0)$ and $a^- = -\min(a, 0)$, respectively and hence $a = a^+ - a^-$. The conventions $a^+_{ij\bullet} = \sum_k a^+_{ijk}$ and $|a|_{ij\bullet\bullet} = \sum_{k<l} |a_{ijkl}|$ are also used for ease of notation.

Symmetry. A multilinear polynomial g can be decomposed into a symmetric and an antisymmetric part, $g(\boldsymbol{x}, \boldsymbol{y}) = g_{\text{sym}}(\boldsymbol{x}, \boldsymbol{y}) + g_{\text{asym}}(\boldsymbol{x}, \boldsymbol{y})$, where the symmetric part is defined by $g_{\text{sym}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}(g(\boldsymbol{x}, \boldsymbol{y}) + g(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}))$ and the antisymmetric part by $g_{\text{asym}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}(g(\boldsymbol{x}, \boldsymbol{y}) - g(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}))$. Note that $g_{\text{sym}}(\boldsymbol{x}, \boldsymbol{y}) = g_{\text{sym}}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$ and $g_{\text{asym}}(\boldsymbol{x}, \boldsymbol{y}) = -g_{\text{asym}}(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$.

Consider the function g evaluated at the two points $(\boldsymbol{x}, \boldsymbol{y})$ and $(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$. We want the function values to be larger than some lower bound ℓ, hence $g(\boldsymbol{x}, \boldsymbol{y}) = g_{\text{sym}}(\boldsymbol{x}, \boldsymbol{y}) + g_{\text{asym}}(\boldsymbol{x}, \boldsymbol{y}) \geq \ell$ and $g(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}) = g_{\text{sym}}(\boldsymbol{x}, \boldsymbol{y}) - g_{\text{asym}}(\boldsymbol{x}, \boldsymbol{y}) \geq \ell$. In order to achieve a maximum lower bound, it follows that $g_{\text{asym}}(\boldsymbol{x}, \boldsymbol{y}) = 0$. Thus, to solve (5), it is enough to restrict our attention to symmetric polynomials[1].

Relationship to bisubmodular functions. For any point $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{B}^{2n}$, it follows from the submodularity and symmetry of g that $g(\boldsymbol{x}, \boldsymbol{y}) \geq g(\boldsymbol{x} \wedge \bar{\boldsymbol{y}}, \boldsymbol{y} \wedge \bar{\boldsymbol{x}})$ where $(\boldsymbol{x} \wedge \bar{\boldsymbol{y}}, \boldsymbol{y} \wedge \bar{\boldsymbol{x}}) \in \mathcal{S}^n$. So, when analyzing $\max \min_{(\boldsymbol{x},\boldsymbol{y})} g(\boldsymbol{x}, \boldsymbol{y})$, it is enough to consider the points in $\mathcal{S}^n$. Similarly, for any two points $(\boldsymbol{x}_1, \boldsymbol{y}_1) \in \mathcal{S}^n$ and $(\boldsymbol{x}_2, \boldsymbol{y}_2) \in \mathcal{S}^n$, we get

$$g(\boldsymbol{x}_1, \boldsymbol{y}_1) + g(\boldsymbol{x}_2, \boldsymbol{y}_2) \geq g((\boldsymbol{x}_1, \boldsymbol{y}_1) \sqcap (\boldsymbol{x}_2, \boldsymbol{y}_2)) + g((\boldsymbol{x}_1, \boldsymbol{y}_1) \sqcup (\boldsymbol{x}_2, \boldsymbol{y}_2)), \tag{7}$$

which is the defining property of a bisubmodular function [6]. Hence, the restriction $g : \mathcal{S}^n \to \mathbb{R}$ is indeed a bisubmodular function.

Persistency. Consider a minimizer $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \mathcal{S}^n$ of the submodular function g. For any point $\boldsymbol{x} \in \mathcal{B}^n$, let $(\boldsymbol{u}, \bar{\boldsymbol{u}}) = ((\boldsymbol{x}, \bar{\boldsymbol{x}}) \sqcup (\boldsymbol{x}^*, \boldsymbol{y}^*)) \sqcup (\boldsymbol{x}^*, \boldsymbol{y}^*)$. One can check that this is well-defined and that $u_i = x_i^*$ if $(x_i^*, y_i^*) \neq (0, 0)$, and $u_i = x_i$ otherwise for $i = 1, \ldots, n$. So, $\boldsymbol{u}$ can be thought of as the result of replacing elements of $\boldsymbol{x}$ by elements $\boldsymbol{x}^*$ provided corresponding element pairs in $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ are nonzero. From bisubmodularity, it follows that, for any $\boldsymbol{v} \in \mathcal{B}^n$,

$$g((\boldsymbol{v}, \bar{\boldsymbol{v}}) \sqcup (\boldsymbol{x}^*, \boldsymbol{y}^*)) \leq g(\boldsymbol{v}, \bar{\boldsymbol{v}}) + \big[g(\boldsymbol{x}^*, \boldsymbol{y}^*) - g((\boldsymbol{v}, \bar{\boldsymbol{v}}) \sqcap (\boldsymbol{x}^*, \boldsymbol{y}^*))\big] \leq g(\boldsymbol{v}, \bar{\boldsymbol{v}}), \tag{8}$$

which implies $f(\boldsymbol{x}) = g(\boldsymbol{x}, \bar{\boldsymbol{x}}) \geq g((\boldsymbol{x}, \bar{\boldsymbol{x}}) \sqcup (\boldsymbol{x}^*, \boldsymbol{y}^*)) \geq g(\boldsymbol{u}, \bar{\boldsymbol{u}}) = f(\boldsymbol{u})$. In particular, if $\boldsymbol{x} \in \operatorname{argmin}(f)$, then $\boldsymbol{u} \in \operatorname{argmin}(f)$. This argument is due to [10].

3. Quadratic Relaxations

A symmetric polynomial $g : \mathcal{B}^{2n} \to \mathbb{R}$ with degree 2 can be represented by[2]

$$g(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2} \sum_i b_i (x_i + \bar{y}_i) + \\ + \frac{1}{2} \sum_{i<j} \Big(b_{ij} (x_i x_j + \bar{y}_i \bar{y}_j) + c_{ij} (x_i \bar{y}_j + \bar{y}_i x_j) \Big). \tag{9}$$

In addition, the constraint $g(\boldsymbol{x}, \bar{\boldsymbol{x}}) = f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{B}^n$ yields that

$$b_i = a_i \text{ and } b_{ij} + c_{ij} = a_{ij},$$

so we can eliminate both b_i and c_{ij}, and are left with the unknowns b_{ij}.

It is well-known that a necessary and sufficient condition for g to be submodular is that the coefficients of the purely quadratic terms are non-positive, that is, $b_{ij} \leq 0$ and $c_{ij} \geq 0$. From the previous discussion, it follows that $b_{ij} = a_{ij} - c_{ij} \leq a_{ij}$ and therefore $b_{ij} \leq \min(0, a_{ij}) = -a_{ij}^-$.

Now, let us return to the maximization problem in (5) for the quadratic case. A brute-force approach would be to enumerate all the 3^n points of $\mathcal{S}^n$, evaluate g and then try to find the coefficients b_{ij} so that ℓ is maximized. This is not very practical, so let us first restrict our attention to the variables x_1, x_2, y_1, y_2 and the points that involve b_{12}. For example, setting $(x_1, x_2, y_1, y_2) = (0, 0, 0, 0)$ gives

$$g(0, 0, \boldsymbol{u}, 0, 0, \boldsymbol{v}) = \frac{1}{2}(a_1 + a_2 + b_{12}) + \\ \frac{1}{2} \sum_{j>2} [(b_{1j} + b_{2j})\bar{y}_j + (c_{1j} + c_{2j}) x_j] + R(\boldsymbol{u}, \boldsymbol{v}), \tag{10}$$

where $\boldsymbol{u} = (x_3, \ldots, x_n)$ and $\boldsymbol{v} = (y_3, \ldots, y_n)$ and R is some remainder polynomial. There are 3^2 combinations, but we know that for those satisfying $x_1 = \bar{y}_1$ and $x_2 = \bar{y}_2$ are independent of b_{12} since $f(\boldsymbol{x}) = g(\boldsymbol{x}, \bar{\boldsymbol{x}})$. The remaining $3^2 - 2^2$ cases are: $(0, 0, 0, 0)$, $(0, 0, 0, 1)$, $(0, 0, 1, 0)$, $(0, 1, 0, 0)$, $(1, 0, 0, 0)$. It is easy to check that all but the first combination are independent of b_{12}. Therefore, in order to find the optimal b_{12} it is enough to consider $g(0, 0, \boldsymbol{u}, 0, 0, \boldsymbol{v})$ given in (10) subject to submodularity constraints. It follows trivially that $b_{12} = \min(0, a_{12})$ maximizes the lower bound. The same result hold for all b_{ij}. This construction of the submodular relaxation g is exactly equivalent to the one in [2] and known as the *roof duality* bound.

Theorem 3.1. *An optimal submodular relaxation g of a function f with $\operatorname{degree}(g) = \operatorname{degree}(f) = 2$ is obtained through roof duality:*

1. *Set $b_i = a_i$ for $1 \leq i \leq n$ in (9),*
2. *Set $b_{ij} = -a_{ij}^-$ and $c_{ij} = a_{ij}^+$ for $1 \leq i < j \leq n$ in (9).*

This result is already known [2, 10], but the proof we have given here is relatively simple and concise.

4. Cubic Relaxations

A cubic symmetric polynomial $g : \mathcal{B}^{2n} \to \mathbb{R}$ which fulfills $g(\boldsymbol{x}, \bar{\boldsymbol{x}}) = f(\boldsymbol{x})$ can be written

$$g(\boldsymbol{x}, \boldsymbol{y}) = L + Q + \frac{1}{2} \sum_{i<j<k} \Big(\\ b_{ijk}(x_i x_j x_k + \bar{y}_i \bar{y}_j \bar{y}_k) + c_{ijk}(x_i x_j \bar{y}_k + \bar{y}_i \bar{y}_j x_k) + \\ d_{ijk}(x_i \bar{y}_j x_k + \bar{y}_i x_j \bar{y}_k) + e_{ijk}(\bar{y}_i x_j x_k + x_i \bar{y}_j \bar{y}_k) \Big), \tag{11}$$

[1] The argument for existence of relaxations still holds for (3) and (4).

[2] We leave out terms of the type $c_{ii} x_i \bar{y}_i$. One can show that c_{ii} will be equal to 0 anyway in an optimal relaxation. Analogously, such terms are also ignored in the cubic and quartic cases.

where L and Q denote linear and quadratic terms (as in the previous section), and $b_{ijk} + c_{ijk} + d_{ijk} + e_{ijk} = a_{ijk}$.

The characterization of submodular cubic polynomials is slightly more complicated than the quadratic case [1]. A cubic polynomial $\sum a_i x_i + \sum a_{ij} x_i x_j + \sum a_{ijk} x_i x_j x_k$ is submodular if and only if, for every $i < j$,

$$a_{ij} + a^+_{ij\bullet} + a^+_{i\bullet j} + a^+_{\bullet ij} \leq 0. \tag{12}$$

Here we give a new formulation suitable for our purposes.

Lemma 4.2. *A cubic symmetric polynomial g represented by (11) is submodular if and only if*

$$b_{ij} + b^+_{ij\bullet} + b^+_{i\bullet j} + b^+_{\bullet ij} + c^+_{ij\bullet} + d^+_{i\bullet j} + e^+_{\bullet ij} \leq 0 \tag{13}$$

$$c_{ij} - c^-_{i\bullet j} - c^-_{\bullet ij} - d^-_{ij\bullet} - d^-_{\bullet ij} - e^-_{ij\bullet} - e^-_{i\bullet j} \geq 0, \tag{14}$$

for $1 \leq i < j \leq n$.

A proof of the lemma is given in the appendix.

One Term. As a start, assume that we want to minimize a pseudo-boolean function f with $n = 3$ variables, so there is at most one cubic term. The problem of computing an optimal relaxation g, cf. (5), can be solved via a linear program in the following way. There are $3^3 - 2^3 = 19$ points in $\mathcal{S}^3$ to consider, and for each of these points, a constraint of the form $g(\boldsymbol{x}, \boldsymbol{y}) \geq \ell$ is generated where ℓ is the lower bound. However, several of these inequalities are independent of the g-parameters (b_{123}, c_{123}, etc.) and therefore not needed. In total, we get

$$\begin{array}{rl} \max & \ell \\ \text{such that} & g(\mathbf{0}, \mathbf{0}) \geq \ell \\ g(\boldsymbol{e}_i, \mathbf{0}) \geq \ell & g(\mathbf{0}, \boldsymbol{e}_i) \geq \ell, \quad i = 1,2,3 \\ b_{12} + b^+_{123} + c^+_{123} \leq 0 & c_{12} - d^-_{123} - e^-_{123} \geq 0 \\ b_{13} + b^+_{123} + d^+_{123} \leq 0 & c_{13} - c^-_{123} - e^-_{123} \geq 0 \\ b_{23} + b^+_{123} + e^+_{123} \leq 0 & c_{23} - c^-_{123} - d^-_{123} \geq 0 \\ b_{12} + c_{12} = a_{12} & b_{123} + \ldots + e_{123} = a_{123} \\ b_{13} + c_{13} = a_{13} & b_{23} + c_{23} = a_{23}. \end{array} \tag{15}$$

Here $\boldsymbol{e}_1 = (1, 0, 0)$, $\boldsymbol{e}_2 = (0, 1, 0)$ and $\boldsymbol{e}_3 = (0, 0, 1)$. The solution gives the optimal relaxation g of f by construction.

Remark. In the above optimization problem, one needs to make the substitutions $b_{123} = b^+_{123} - b^-_{123}$, $c_{123} = c^+_{123} - c^-_{123}$, etc. in order to actually get an LP. Further, one can show that if $a_{123} \geq 0$ then one can set $b^-_{123} = c^-_{123} = d^-_{123} = e^-_{123} = 0$, and analogously when $a_{123} \leq 0$. This holds for all coefficients, both for cubic and quartic relaxations.

Example. Consider again the example of a cubic pseudo-boolean function f in (2). Finding a $g(\boldsymbol{x}, \boldsymbol{y})$ of the form (11) by solving (15) results in

$$\begin{aligned} g(\boldsymbol{x}, \boldsymbol{y}) = & -(x_1 + \bar{y}_1) + \frac{1}{2}(x_2 + \bar{y}_2) - \frac{1}{2}(x_3 + \bar{y}_3) \\ & + 2(x_1\bar{y}_2 + \bar{y}_1 x_2) + 2(x_1\bar{y}_3 + \bar{y}_1 x_3) \\ & - (x_2 x_3 + \bar{y}_2\bar{y}_3) - (\bar{y}_1 x_2 x_3 + x_1 \bar{y}_2 \bar{y}_3). \end{aligned} \tag{16}$$

Minimizing this submodular relaxation gives $g_{\text{min}} = -2$ for $(\boldsymbol{x}^*, \boldsymbol{y}^*) = (0, 1, 1, 1, 0, 0)$. Since $\boldsymbol{x}^* = \bar{\boldsymbol{y}}^*$, it follows that $\boldsymbol{x}^*$ is the global minimizer for f as well.

Multiple Terms. For a cubic pseudo-boolean function f in n variables, the above strategy will not work as $3^n - 2^n$ grows very quickly for increasing n. Two obvious heuristic alternatives are:

1. Decompose f into a sum of the form $f(\boldsymbol{x}) = \sum_{i<j<k} f_{ijk}(x_i, x_j, x_k)$ and compute an optimal relaxation for each term f_{ijk}. However, the sum of optimal relaxations is generally not optimal.
2. Use a subset of the points in $\mathcal{S}^n$ for $g(\boldsymbol{x}, \boldsymbol{y}) \geq \ell$ to get an approximate optimal relaxation.

Neither of these approaches are satisfactory. One may even wonder if the optimal relaxation g is polynomial time computable at all?

In Section 2, we showed that persistency holds for any bisubmodular relaxation g — optimal or not. From the example in (2), it is clear, however, that not all relaxations are equally powerful. Instead of solving (5), which, although possible, may require a large number of constraints, one can consider a simpler problem:

$$\begin{array}{rl} \max\limits_{g} & g(\mathbf{0}, \mathbf{0}) \\ \text{such that} & g \text{ symmetric and submodular} \\ & g(\boldsymbol{x}, \bar{\boldsymbol{x}}) = f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{B}^n. \end{array} \tag{17}$$

Instead of maximizing $\min g(\boldsymbol{x}, \boldsymbol{y})$, we are only maximizing $g(\mathbf{0}, \mathbf{0})$. This problem is considerably less arduous and can be solved in polynomial time. Given the minimizer $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \text{argmin}(g)$ which is also polynomial time computable via graph cuts, we can make the following important observations:

- If $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is non-zero, then we can use persistency to get a partial solution and reduce the number of variables in f.
- Otherwise, as the optimum is indeed the trivial solution, and as $g(\mathbf{0}, \mathbf{0})$ is maximized in the construction of g, then we can conclude that g is an optimal relaxation and we have obtained the generalized roof duality bound.

This observation leads to the following algorithm that computes the generalized roof duality bound.

1. Construct g by solving (17).
2. Compute $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \operatorname{argmin}(g)$.
3. If $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is non-zero then use persistency to simplify f and start over from 1. Otherwise, stop.

The algorithm can obviously not run for more than n iterations, since in each iteration either persistencies are found and f is simplified or the algorithm terminates. With all steps being solvable in polynomial time, the algorithm itself is polynomial.

Theorem 4.3. *A solution that attains the generalized roof duality bound for a cubic pseudo-boolean function can be computed in polynomial time.*

In fact, as we shall see in the experimental section, the number of iterations is usually quite small and the linear program can be solved fast provided the number of cubic terms is limited.

5. Quartic Relaxations

Similar to previous derivations, a quartic symmetric polynomial g can be written

$$g(\boldsymbol{x}, \boldsymbol{y}) = L + Q + C + \frac{1}{2} \sum_{i<j<k<l} \Big(\tag{18}$$

$$b_{ijkl}(x_i x_j x_k x_l + \bar{y}_i \bar{y}_j \bar{y}_k \bar{y}_l) + c_{ijkl}(x_i x_j x_k \bar{y}_l + \bar{y}_i \bar{y}_j \bar{y}_k x_l) +$$
$$d_{ijkl}(x_i x_j \bar{y}_k x_l + \bar{y}_i \bar{y}_j x_k \bar{y}_l) + e_{ijkl}(x_i \bar{y}_j x_k x_l + \bar{y}_i x_j \bar{y}_k \bar{y}_l) +$$
$$p_{ijkl}(\bar{y}_i x_j x_k x_l + x_i \bar{y}_j \bar{y}_k \bar{y}_l) + q_{ijkl}(x_i x_j \bar{y}_k \bar{y}_l + \bar{y}_i \bar{y}_j x_k x_l) +$$
$$r_{ijkl}(x_i \bar{y}_j x_k \bar{y}_l + \bar{y}_i x_j \bar{y}_k x_l) + s_{ijkl}(x_i \bar{y}_j \bar{y}_k x_l + \bar{y}_i x_j x_k \bar{y}_l) \Big),$$

where L, Q and C denote lower-order terms, and $b_{ijkl} + c_{ijkl} + \ldots + s_{ijkl} = a_{ijkl}$.

Determining whether a given quartic polynomial is submodular or not is known to be NP-hard, and not all submodular quartic polynomials can be reduced to a quadratic submodular function [25]. Therefore, a compromise is required. We choose to work with the quartic polynomials $\sum a_i x_i + \ldots + \sum a_{ijkl} x_i x_j x_k x_l$ that satisfy, for every $i < j$,

$$a_{ij} + a^+_{ij\bullet} + a^+_{i\bullet j} + a^+_{\bullet ij} + a^+_{ij\bullet\bullet} + a^+_{i\bullet j\bullet} + \ldots + a^+_{\bullet\bullet ij} \leq 0. \tag{19}$$

This choice can be seen as a natural generalization of the cubic case, cf. (12). The class has a number of advantageous properties. First, it is still a very rich class of submodular functions [25]. Second, each quartic term only needs one auxiliary variable for reduction. Finally, only $O(n^2)$ inequalities are needed to make sure our relaxation is submodular and reducible. The corresponding conditions for a symmetric quartic polynomial is given in Lemma A.5.

The algorithm derived in the previous section for cubic pseudo-boolean polynomials applies for the quartic case as well. The only qualitative difference is that working with the full set of submodular quartics is not tractable, hence a restriction to a subset is necessary.

Corollary 5.4. *Let $\mathcal{G}$ denote the set of submodular symmetric quartic functions given in Lemma A.5. A solution that attains the generalized duality bound over relaxations in $\mathcal{G}$ for a quartic pseudo-boolean function can be computed in polynomial time.*

Example. In [8], the following reduction identity is proposed: $x_1x_2x_3x_4 = \min_{z\in\mathcal{B}} z(3-2x_1-2x_2-2x_3-2x_4) + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$. This can be used to transform

$$f(\boldsymbol{x}) = x_1 + x_3 - x_4 + 2x_1x_4 + 2x_2x_3 - x_3x_4 + x_1x_2x_3x_4$$

to a quadratic polynomial with one auxiliary variable z. The quadratic roof duality bound gives $f_{\min} \geq -2$ and no partial assignments. On the other hand, solving the linear program (17), one obtains the relaxation

$$g(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}(x_1 + \bar{y}_1) + \frac{1}{2}(x_3 + \bar{y}_3) - \frac{1}{2}(x_4 + \bar{y}_4)$$
$$- \frac{1}{2}(x_1x_3 + \bar{y}_1\bar{y}_3) + \frac{1}{2}(x_1\bar{y}_3 + \bar{y}_1x_3) - \frac{1}{2}(x_1x_4 + \bar{y}_1\bar{y}_4)$$
$$+ \frac{3}{2}(x_1\bar{y}_4 + x_1\bar{y}_4) + (x_2\bar{y}_3 + \bar{y}_2x_3) - \frac{1}{2}(x_3x_4 + \bar{y}_3\bar{y}_4)$$
$$+ \frac{1}{2}(x_1\bar{y}_2x_3x_4 + \bar{y}_1x_2\bar{y}_3\bar{y}_4).$$

Solving the submodular problem $\min g(\boldsymbol{x}, \boldsymbol{y})$ via graph cuts yields $(\boldsymbol{x}^*, \boldsymbol{y}^*) = (0, 0, 0, 1, 1, 1, 1, 0)$. Again, since $\boldsymbol{x}^* = \bar{\boldsymbol{y}}^*$, it follows that $\boldsymbol{x}^*$ is the global minimizer for f.

Remark. It is possible to construct examples for which the optimal relaxation gives no persistencies, whereas solving (17) does, and via iterations, a final solution is obtained that gives a better lower bound than the minimum of the optimal relaxation.

6. Experiments

In this section we experimentally evaluate the generalized roof duality bound. The current state-of-the-art for minimizing pseudo-boolean functions with persistency is higher-order clique reduction (HOCR) [8], to which our method is compared.

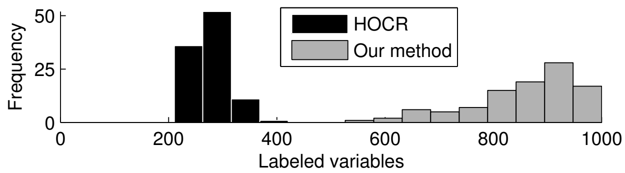


Figure 1. Number of labeled variables for 100 random cubic polynomials with $n = 1000$ and $|T| = 1000$.

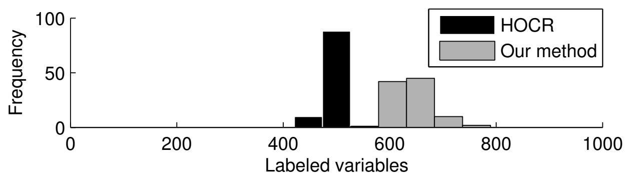


Figure 2. Number of labeled variables for 100 random quartic polynomials with $n = 1000$ and $|T| = 200$.

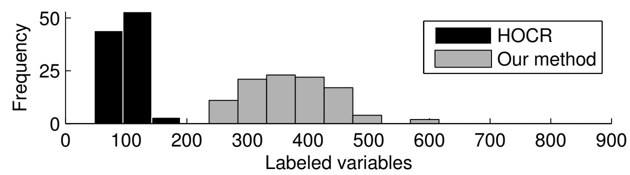


Figure 3. Number of labeled variables for 100 random problems on a 30×30 grid with 2 cubic terms per variable.

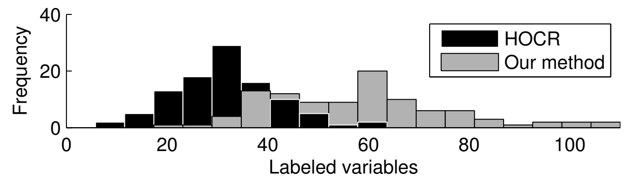


Figure 4. Random problems on a 30×30 grid with 4 cubic terms per variable. Note the scale of the x-axis compared to Fig. 3.

6.1. Synthetic Polynomials

In the first experiment, we apply our method to synthetically generated polynomials with random coefficients:

$$f(\boldsymbol{x}) = \sum_{(i,j,k)\in T} f_{ijk}(x_i, x_j, x_k), \tag{20}$$

where $T \subseteq \{1 \ldots n\}^3$ is a random set of triplets and each f_{ijk} is a cubic polynomial in x_i, x_j and x_k with all its coefficients picked uniformly in $\{-100, \ldots, 100\}$. We minimize f using both standard reduction to a quadratic function as proposed in [8] and the method introduced in this paper. After the algorithm finishes, we count the number of labeled variables (i.e. the number of persistencies). The results from 100 problem instances with $n = 1000$ and $|T| = 1000$ can be seen in Fig. 1. For this type of polynomial, an optimal submodular relaxation significantly outperforms the method in [8] for every problem. The time required to solve the linear program (17) was 200–300 milliseconds.[3] The minimum and maximum number of iterations required was 3 and 12, respectively, with 93% of the problem instances requiring 6 or less. In addition to comparing the number of labeled variables, we also compared the achieved lower bounds by computing the relative difference: $(\ell_{\text{optimal}} - \ell_{\text{HOCR}})/|\ell_{\text{optimal}}|$. The minimum, median and maximum were 0.09, 0.14 and 0.22, respectively.

We also generated random quartic polynomials in the same manner, see Fig. 2. Here, the difference in number of labeled variables was smaller, but the relative lower bounds were very large: 0.44, 0.54 and 0.64 for the minimum, median and maximum, respectively.

[3] We used Clp (http://www.coin-or.org/Clp) as our LP solver.

6.2. Random Cliques in a Grid

Instead of choosing the set T completely at random, one can pick pairs (i, j, k) representing cliques in a grid graph. Fig. 5a shows a couple of examples. These types of polynomials commonly occur in computer vision. We continue to pick all coefficients at random, which generates polynomials which are difficult to minimize.

The results when adding two cliques per variable can be seen in Fig. 3. On average, generalized roof duality labeled more than 200 additional variables. In Fig. 4 we added 4 random cliques per variable. As expected, this problem was very difficult and no method labeled more than about 10% of the variables (it is of course NP-hard in general to label even one). Still, the generalized roof duality bound performed slightly better.

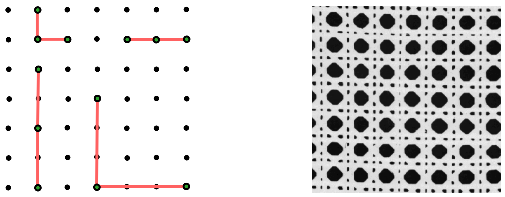
6.3. Binary Texture Restoration

The problem of (binary) texture restoration is often used to evaluate minimization methods for pseudo-boolean functions (see e.g. [11, 19]). Fig. 5b shows a Brodatz texture which can be used to train a higher-order models using cliques of the type shown in Fig. 5a. A noisy observation of a piece of the texture (Fig. 6a) can then be restored by minimizing a pseudo-boolean polynomial.

Reducing the resulting cubic polynomial to a quadratic one with HOCR and solving for the generalized roof duality bound give different results. Fig. 6b shows the result when reducing the cubic terms and Fig. 6c the result for the generalized roof duality bound.

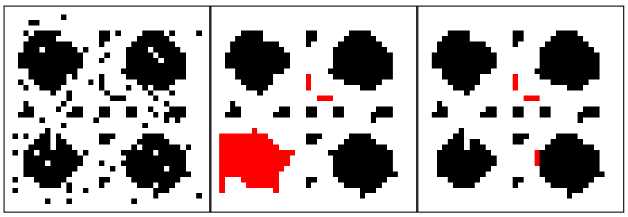
6.4. Image Denoising

Ishikawa [8] demonstrated that quartic terms are useful for denoising images. In each iteration a proposal is generated in two possible ways which are alternated: by blurring



(a) Examples of cubic cliques. (b) Training image

Figure 5. An example texture used to train a higher-order model for restoration.



(a) Texture with noise. (b) HOCR. (c) Our method.

Figure 6. Texture restoration. Quadratic reduction with roof duality results in 117 variables unlabeled (red). Our method labels all but 9 variables.

(a) Original image (b) Noisy image (c) Restored image

Figure 7. Quartic cliques are useful for image denoising.

the current image and picking all pixels at random. Each pixel then either stays the same or is changed to the proposal image. Unlabeled pixels are not changed. The smoothness term consists of a Fields of Experts (FoE) model using patches of size 2×2. Thus, quartic polynomials are needed to formulate the image restoration task as a pseudo-boolean minimization problem.

Figure 7 shows the final restoration result and Fig. 8 shows a comparison between the reduction method in [8] and generalized roof duality proposed in this paper. We ran the experiment in a way that both methods solved exactly the same optimization problem in each iteration and the image was updated using the result from the quadratic reduction. Generalized roof duality performed very well, often labeling very close to 100% of the problem variables.

We did not compare to any more sophisticated techniques, such as probing, which was done in [8]. However, we note that probing applies similarly to both quadratic reduction methods and our roof duality bound. Investigating probing would be one direction of future studies.

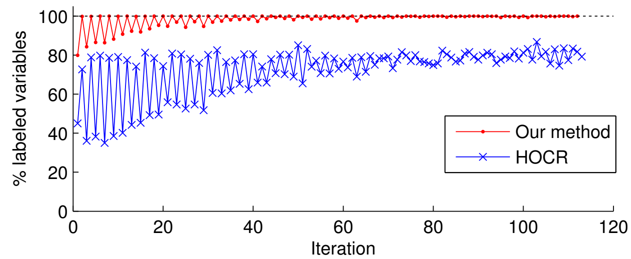


Figure 8. Image denoising as described in [8]. Blue: reduction to quadratic (HOCR); Red: our method. In each iteration both methods solved the same optimization problem.

7. Conclusions

We have described a generalization of roof duality for polynomials of any degree. By directly working with the higher-order polynomials, we can achieve better lower bounds than standard techniques that first reduce to a quadratic polynomial and then apply the roof duality bound. We have shown that this framework is not only of theoretical interest, but performs significantly better in terms of number of persistencies and lower bounds for both synthetic data and real problems in computer vision.

Optimal relaxations are not necessary for all problems. There are (easy) problems for which any reduction technique seems to work equally well. There are also very hard problems, for which even optimal relaxations cannot label a single variable. Still, our experiments indicate that optimal relaxations are very useful compared to standard reduction techniques. Also, this work puts a natural bound on how well any submodular relaxation technique can work.

The price to pay for computing a better lower bound is in the effort of solving a linear programming problem, cf. (17). Instead of solving this problem exactly, one can employ a simple heuristic to approximately maximize $g(\mathbf{0}, \mathbf{0})$ while fulfilling all constraints. If the minimizer of the resulting g is not equal to $(\mathbf{0}, \mathbf{0})$, persistencies can still be used to simplify the problem (recall that we have persistency for *all* relaxations g, not just the optimal one). By analyzing the linear program more closely, one can see that there is a lot of structure, and one can even hope for a closed-form solution. We conjecture that there is a closed-form solution, or at least, an effective heuristic. Eliminating the need for solving a linear program is one important direction of future research.

Source code. The source code for computing the generalized roof duality may be downloaded from our web page.

Acknowledgments. The starting point for this work was an inspiring talk given by Vladimir Kolmogorov in Paris, June 2010. The work has also benifited from discussions with Carl Olsson. Funding from the Swedish Foundation for Strategic Research (FFL and VINST) and by the European Research Council (GlobalVision grant no. 209480) is gratefully acknowledged.

Appendix

Proof of Lemma 4.2. First, expand the symmetric representation to a cubic polynomial in the variables x_i and y_i, $i = 1, \ldots, n$. Then, apply the necessary and sufficient condition in (12) for every $i < j$ on $x_i x_j$ (and its twin $y_i y_j$) which gives (13) as well as on $x_i y_j$ (and its twin $y_i x_j$) which gives (14). □

In the same way, by first expanding all conjugate factors in the symmetric, quartic form and then applying the sufficient condition (19), one obtains the following constraints.

Lemma A.5. *A quartic symmetric polynomial g represented by (18) is submodular and reducible to a quadratic submodular polynomial if*

$$\begin{aligned}\tilde{b}_{ij} + b^+_{ij\bullet\bullet} + b^+_{i\bullet j\bullet} + b^+_{i\bullet\bullet j} + b^+_{\bullet ij\bullet} + b^+_{\bullet i\bullet j} + b^+_{\bullet\bullet ij} + \\ |c|_{ij\bullet\bullet} + |c|_{i\bullet j\bullet} + |c|_{\bullet ij\bullet} + |d|_{ij\bullet\bullet} + |d|_{i\bullet\bullet j} + |d|_{\bullet i\bullet j} + \\ |e|_{i\bullet j\bullet} + |e|_{i\bullet\bullet j} + |e|_{\bullet\bullet ij} + |p|_{\bullet ij\bullet} + |p|_{\bullet i\bullet j} + |p|_{\bullet\bullet ij} + \\ q^+_{ij\bullet\bullet} + |q|_{ij\bullet\bullet} + q^+_{\bullet\bullet ij} + r^+_{i\bullet j\bullet} + |r|_{i\bullet j\bullet} + r^+_{\bullet i\bullet j} + \\ s^+_{i\bullet\bullet j} + |s|_{i\bullet\bullet j} + s^+_{\bullet ij\bullet} \leq 0\end{aligned}$$

$$\begin{aligned}\tilde{c}_{ij} - c^-_{i\bullet\bullet j} - c^-_{\bullet i\bullet j} - c^-_{\bullet\bullet ij} - d^-_{i\bullet j\bullet} - d^-_{\bullet ij\bullet} - d^-_{\bullet\bullet ij} \\ - e^-_{ij\bullet\bullet} - e^-_{\bullet ij\bullet} - e^-_{\bullet i\bullet j} - p^-_{ij\bullet\bullet} - p^-_{i\bullet j\bullet} - p^-_{i\bullet\bullet j} \\ - |q|_{i\bullet j\bullet} - |q|_{i\bullet\bullet j} - |q|_{\bullet ij\bullet} - |q|_{\bullet i\bullet j} \\ - |r|_{ij\bullet\bullet} - |r|_{i\bullet\bullet j} - |r|_{\bullet ij\bullet} - |r|_{\bullet\bullet ij} \\ - |s|_{ij\bullet\bullet} - |s|_{i\bullet j\bullet} - |s|_{\bullet i\bullet j} - |s|_{\bullet\bullet ij} \geq 0,\end{aligned}$$

for $1 \leq i < j \leq n$, where $\tilde{b}_{ij}$ and $\tilde{c}_{ij}$ denote the left-hand side of inequalities (13) and (14), respectively.

References

[1] A. Billionet and M. Minoux. Maximizing a supermodular pseudo-boolean function: a polynomial algorithm for cubic functions. *Discrete Appl. Math.*, 12:1–11, 1985.

[2] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Appl. Math.*, 123:155–225, 2002.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[4] D. Cremers and L. Grady. Statistical priors for efficient combinatorial optimization via graph cuts. In *European Conf. Computer Vision*, Graz, Austria, 2006.

[5] D. Freedman and P. Drineas. Energy minimization via graph cuts: Settling what is possible. In *Conf. Computer Vision and Pattern Recognition*, San Diego, USA, 2005.

[6] S. Fujishige and S. Iwata. Bisubmodular function minimization. *SIAM J. Discrete Math.*, 19(4):1065–1073, 2006.

[7] A. C. Gallagher, D. Batra, and D. Parikh. Inference for order reduction in Markov random fields. In *Conf. Computer Vision and Pattern Recognition*, Colorado Springs, USA, 2011.

[8] H. Ishikawa. Transformation of general binary MRF minimization to the first order case. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(6):1234–1249, 2011.

[9] P. Kohli, M. Kumar, and P. Torr. P^3 & beyond: Move making algorithms for solving higher order functions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(9):1645–1656, 2009.

[10] V. Kolmogorov. Generalized roof duality and bisubmodular functions. In *Neural Information Processing Systems*, Vancouver, Canada, 2010.

[11] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts - a review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007.

[12] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[13] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *Conf. Computer Vision and Pattern Recognition*, Miami, USA, 2009.

[14] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph cut based inference with co-occurrence statistics. In *European Conf. Computer Vision*, Crete, Greece, 2010.

[15] X. Lan, S. Roth, D. Huttenlocher, and M. Black. Efficient belief propagation with learned higher-order Markov random fields. In *European Conf. Computer Vision*, Graz, Austria, 2006.

[16] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for Markov random field optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(8):1392–1405, 2010.

[17] S. H. Lu and A. C. Williams. Roof duality for polynomial 0-1 optimization. *Math. Programming*, 37(3):357–360, 1987.

[18] S. Nowozin and C. Lampert. Global interactions in random field models: A potential function ensuring connectedness. *SIAM J. Imaging Sciences*, 3(4):1048–1074, 2010.

[19] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *Conf. Computer Vision and Pattern Recognition*, Miami, USA, 2009.

[20] T. Schoenemann, F. Kahl, and D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In *Int. Conf. Computer Vision*, Kyoto, Japan, 2009.

[21] P. Strandmark and F. Kahl. Parallel and distributed graph cuts by dual decomposition. In *Conf. Computer Vision and Pattern Recognition*, San Francisco, USA, 2010.

[22] P. Strandmark and F. Kahl. Curvature regularization for curves and surfaces in a global optimization framework. In *EMMCVPR*, St Petersburg, Russia, 2011.

[23] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for MAP-MRF. In *Conf. Computer Vision and Pattern Recognition*, Anchorage, USA, 2008.

[24] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(12):2115–2128, 2009.

[25] S. Zivny, D. A. Cohen, and P. G. Jeavons. The expressive power of binary submodular functions. *Discrete Appl. Math.*, 157(15):3347–3358, 2009.