Tighter Relaxations for Higher-Order Models based on Generalized Roof Duality

Johan Fredriksson, Carl Olsson, Petter Strandmark, and Fredrik Kahl

Centre for Mathematical Sciences, Lund University, Sweden
http://www.maths.lth.se/vision/

Abstract. Many problems in computer vision can be turned into a large-scale boolean optimization problem, which is in general NP-hard. In this paper, we further develop one of the most successful approaches, namely roof duality, for approximately solving such problems for higher-order models. Two new methods that can be applied independently or in combination are investigated. The first one is based on constructing relaxations using generators of the submodular function cone. In the second method, it is shown that the roof dual bound can be applied in an iterated way in order to obtain a tighter relaxation. We also provide experimental results that demonstrate better performance with respect to the state-of-the-art, both in terms of improved bounds and the number of optimally assigned variables.

1 Introduction

Discrete energy minimization methods have become the golden standard for many computer vision and machine learning problems. Their ability to compute globally optimal solutions or strong relaxations makes them suitable for a large class of problems such as dense matching/stereo, segmentation, image synthesis [1]. Often formulations with pair-wise cliques are used to incorporate length regularization [2]. In this case graph cuts are able to compute optimal or guaranteed near optimal solutions for binary and multi-class problems [3].

The modeling power of pair-wise cliques is however limited and there has been an increasing interest in higher-order interactions. For example, to avoid the well known shrinking bias of length or area based approaches, curvature regularization which requires higher-order models is considered in [4, 5]. Other examples are [6] where approximate belief propagation is used for inference of a higher-order learned model, [7] where second order smoothness priors are used for stereo and [8] which uses a higher-order model for texture restoration. Even potentials where the cliques involve all variables have been considered [9].

In this paper we are interested in minimizing energies with higher-order interactions. Specifically, we will consider the quartic case (4th order energies), but in principle our methods can be applied to interactions of any order. Our work builds on the generalized roof duality methods presented in [10–12]. Here it is shown that a lower bound on the minimum of an nth order pseudo-boolean function can be found by performing a maximization over a set of nth order symmetric submodular pseudo-boolean functions. In practice however, determining whether a function is submodular is co-NP complete if $n \geq 4$ [13]. Therefore searching over the set of submodular functions is difficult. We propose to circumvent this problem by optimizing over positive linear combinations of a set of generators (or extreme rays). The idea of generators was first introduced in [12]. In this paper, even though the resulting function space is only a subset of all submodular functions, we show for $4th$ order models that optimizing over this subset yields significantly better results than previously published methods.

We also present a method for improving the lower bound and increasing the number of persistent variables based on a symmetric extension of the original objective function. The approach is iterative and is guaranteed to produce at least the lower bound of the roof dual. A similar approach was developed in [14] for the special case of quadratic pseudo-boolean functions.

Related work. In recent years a number of strategies for optimizing higher-order energies have been proposed. In [15, 16] dual decomposition is used and move-making algorithms are proposed in [17, 18]. Furthermore, linear programming approaches have been considered [19] as well as belief propagation [6].

Our approach is based on max-flow/min-cut methods which are considered to be state-of-the-art for quadratic polynomials [1]. To handle higher-order interactions, reduction techniques have been developed [20–22]. In [20], 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 contrast our work is based on [10, 12] which introduce a framework that works directly on the higher-order potentials.

The extension method has similarities to the probing method of [1] in the sense that both methods fixate one variable at the time and re-run the max-flow computations for the modified graph. However, we need only to fixate a variable once whereas probing requires fixating a variable to both 0 and 1. The methods can be used in combination for tighter relaxations.

2 Generalized Roof Duality

In this section we will briefly state some of the results from [12] that serve as a basis for our methods. The basic problem is that of minimizing a pseudo-boolean polynomial $f : \{0,1\}^n \mapsto \mathbb{R}$ of degree m. Since this problem is in general NP-hard the following family of relaxations is considered:

$$\begin{aligned} l(g) := \min\nolimits_{(\boldsymbol{x},\boldsymbol{y})\in\{0,1\}^{2n}} \; & g(\boldsymbol{x},\boldsymbol{y}) && (1)\\ \text{s.t.} \quad & g \text{ is submodular} && (2)\\ & g(\boldsymbol{x},\bar{\boldsymbol{x}}) = f(\boldsymbol{x}) && (3)\\ & g(\boldsymbol{x},\boldsymbol{y}) = g(\bar{\boldsymbol{y}},\bar{\boldsymbol{x}}). && (4) \end{aligned}$$

Here the notation $\bar{\boldsymbol{x}} = (1-x_1, 1-x_2, \ldots, 1-x_n)$ is used. Note that the relaxation g has twice the number of variables, that is, $g : \{0,1\}^{2n} \mapsto \mathbb{R}$. A function g

satisfying the last condition $g(\boldsymbol{x}, \boldsymbol{y}) = g(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}})$ is said to be *symmetric*. It can be seen that $l(g)$ is a lower bound on the minimum of f for any feasible g. Furthermore, [10] shows persistency of the minimizers of g: if $x_i^* = \bar{y}_i^*$ for a minimizer $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ of g then any minimizer $\boldsymbol{z}^*$ of f must have $z_i^* = x_i^*$. Hence, persistency can be used to determine the optimal assignment of a single variable.

The goal is now to find the strongest possible bound $l(g)$, that is, to maximize $l(g)$ over the set of feasible functions. To avoid solving the max-min problem, [12] proposes the following procedure:

1. Find $g^* \in \operatorname{argmax} g(\mathbf{0}, \mathbf{0})$, where the maximum is taken over all functions g fulfilling (2)-(4).
2. Compute a minimizer $(\boldsymbol{x}^*, \boldsymbol{y}^*) \in \operatorname{argmin} g^*(\boldsymbol{x}, \boldsymbol{y})$.
3. If $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is not identically zero, use persistency to simplify f and goto 1. Otherwise, stop.

The above procedure can be proved to provide a solution that gives a bound which is equal to or higher than $\max l(g)$. It turns out that if f is a quadratic pseudo-boolean function, then the obtained maximum bound is the same as the roof dual bound. Furthermore, for the cubic case ($m = 3$) it is shown that the maximization of $g(\mathbf{0}, \mathbf{0})$ can be computed using linear programming.

Unfortunately, when $m > 3$ determining whether a given polynomial is submodular is co-NP-complete. One way of avoiding this problem is to restrict the space of functions in the maximization of $g(\mathbf{0}, \mathbf{0})$ to a subset that is easy to generate. Ideally this set should be selected so that the maximum lower bound is not weakened too much. Next we will show how to use a set of generators to approximate the set of feasible functions well in the quartic case ($m = 4$).

3 Generators for Submodular Functions

The submodular symmetric functions of degree 4 form a cone. In this section we will construct a basis for a subcone that enables us to optimize over it efficiently. The elements of the basis are called *generators* and each submodular symmetric function can be written as a linear combination (with positive coefficients) of the generators. Unfortunately, not all generators of degree 4 can be optimized using max-flow/min-cut algorithms. Therefore we have to settle for optimizing over a subset of the cone.

The generators of the submodular (non-symmetric) cone are given in [23] for $n = 4$. There are 10 generator classes, however, one of the classes cannot be optimized using max-flow/min-cut [24]. For each generator $e(x_i, x_j, x_k, x_l)$ in the remaining 9 classes, we can construct a symmetric generator as

$$e(x_i, x_j, x_k, x_l) + e(\bar{y}_i, \bar{y}_j, \bar{y}_k, \bar{y}_l). \tag{5}$$

Such a generator can only generate monomials with either $\boldsymbol{x}$-variables or $\boldsymbol{y}$-variables. Therefore, we also incorporate generators where $\boldsymbol{x}$- and $\boldsymbol{y}$-variables have switched places, such as

$$e(y_i, x_j, x_k, x_l) + e(\bar{x}_i, \bar{y}_j, \bar{y}_k, \bar{y}_l). \tag{6}$$

The procedure gives 132 quartic generators for each combination of indices i, j, k, l. The same procedure can be applied to the lower order generators which result in 8 cubic generators for each i, j, k and 2 quadratic for each i, j.

In order to generate the roof dual bound we need to be able to maximize $g(\mathbf{0}, \mathbf{0})$ over feasible functions g. Given the generators above we can construct symmetric submodular functions from positive linear combinations

$$g(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{k} \alpha_i e_i(\boldsymbol{x}, \boldsymbol{y}), \quad \alpha_i \geq 0, \tag{7}$$

where $\{e_i\}_{i=1}^k$ are all of our generators. Moreover, by identifying coefficients, the constraint $g(\boldsymbol{x}, \bar{\boldsymbol{x}}) = f(\boldsymbol{x})$ can be implemented as a linear system of equations $A\boldsymbol{\alpha} = \boldsymbol{a}$, where $\boldsymbol{\alpha}$ is a vector containing all the coefficients α_i in the linear combination (7). Therefore the maximization of $g(\mathbf{0}, \mathbf{0})$ can be formulated as the linear program

$$\max \ \boldsymbol{c}^T \boldsymbol{\alpha} \tag{8}$$
$$\text{s.t.} \ \ A\boldsymbol{\alpha} = \boldsymbol{a} \tag{9}$$
$$\boldsymbol{\alpha} \geq \mathbf{0}, \tag{10}$$

where the vector $\boldsymbol{c}$ contains the coefficients for $e_i(\mathbf{0}, \mathbf{0})$.

In [11], a completely different way is used to construct g, not using generators. It can be shown that the feasible set of functions in this construction is a strict subset of the function cone generated by our generators.

4 Symmetric Extension

In order to improve the lower bound and increase the number of persistencies of $f : \{0, 1\}^n \to \mathbb{R}$, we will extend $f(\boldsymbol{x})$ by introducing an additional variable x_0. Let $\phi : \{0, 1\}^{n+1} \to \mathbb{R}$ be the extension of $f(\boldsymbol{x})$ such that

$$\phi(\boldsymbol{x}, x_0) = x_0 f(\boldsymbol{x}) + \bar{x}_0 f(\bar{\boldsymbol{x}}). \tag{11}$$

By construction we have $\phi(\boldsymbol{x}, 1) = f(\boldsymbol{x})$ and $\phi(\bar{\boldsymbol{x}}, 0) = f(\boldsymbol{x})$. The function ϕ is symmetric in the sense that $\phi(\boldsymbol{x}, x_0) = \phi(\bar{\boldsymbol{x}}, \bar{x}_0)$ and therefore

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \min_{x_0=1} \phi(\boldsymbol{x}, x_0) = \min_{x_0=0} \phi(\boldsymbol{x}, x_0). \tag{12}$$

It is easy to see that the same holds for any other variable x_k, $k = 1, \ldots, n$,

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \min_{x_k=1} \phi(\boldsymbol{x}, x_0) = \min_{x_k=0} \phi(\boldsymbol{x}, x_0). \tag{13}$$

The key observation is that if we can determine the optimal value of any of the variables for $\min_{x_k=1} \phi(\boldsymbol{x}, x_0)$ through persistency, then we can simplify the original problem $\min f(\boldsymbol{x})$.

Lemma 1. *If*

$$(\boldsymbol{x}^*, x_0^*) \in \operatorname{argmin}_{x_k=1} \phi(\boldsymbol{x}, x_0) \Rightarrow x_i^* = 1 \tag{14}$$

for some $i \neq k$, then, for every

$$\boldsymbol{z}^* \in \operatorname{argmin} f(\boldsymbol{z}) \Rightarrow z_k^* = z_i^*. \tag{15}$$

Proof. Let us assume that (14) holds but there is some solution $\boldsymbol{z}^*$ with $z_k^* \neq z_i^*$ and $f(\boldsymbol{z}^*) = \min f(\boldsymbol{z})$. We get two cases; either $z_k^* = 1$ or $z_k^* = 0$.

If $z_k^* = 1$ then we have

$$\phi(\boldsymbol{z}^*, 1) = f(\boldsymbol{z}^*) = \min f(\boldsymbol{z}) = \min_{x_k=1} \phi(\boldsymbol{x}, x_0). \tag{16}$$

Therefore $(\boldsymbol{z}^*, 1) \in \operatorname{argmin}_{x_k=1} \phi(\boldsymbol{x}, x_0)$ but $z_i^* = \bar{z}_k^* = 0$ contradicting (14).

If $z_k^* = 0$ then we have

$$\phi(\bar{\boldsymbol{z}}^*, 0) = \phi(\boldsymbol{z}^*, 1) = f(\boldsymbol{z}^*) = \min f(\boldsymbol{z}) = \min_{x_k=1} \phi(\boldsymbol{x}, x_0). \tag{17}$$

Therefore $(\bar{\boldsymbol{z}}^*, 0) \in \operatorname{argmin}_{x_k=1} \phi(\boldsymbol{x}, x_0)$ but $\bar{z}_i^* = z_k^* = 0$ contradicting (14).

Hence, we can reduce the number of variables of our original problem for each persistency obtained from the fixations of the symmetric extension. In case we can additionally determine x_0^*, the following stronger result holds.

Corollary 1 *If*

$$(\boldsymbol{x}^*, x_0^*) \in \operatorname{argmin}_{x_k=1} \phi(\boldsymbol{x}, x_0) \Rightarrow x_i^* = x_0^* = 1 \tag{18}$$

for some $i \neq k$, then, for every

$$\boldsymbol{z}^* \in \operatorname{argmin} f(\boldsymbol{z}) \Rightarrow z_k^* = z_i^* = 1. \tag{19}$$

Proof. Suppose (18) holds and there is a solution $\boldsymbol{z}^*$ with $z_i^* = 0$. First $z_i^* = z_k^*$ according to the Lemma 1. Then

$$\phi(\bar{\boldsymbol{z}}^*, 0) = \phi(\boldsymbol{z}^*, 1) = f(\boldsymbol{z}^*) = \min f(\boldsymbol{z}) = \min_{x_k=1} \phi(\boldsymbol{x}, x_0). \tag{20}$$

Therefore $(\bar{\boldsymbol{z}}^*, 0) \in \operatorname{argmin}_{x_k=1} \phi(\boldsymbol{x}, x_0)$ but $z_0^* = 0$ contradicting (18).

In Lemma 1 and Corollary 1 we have only considered the cases when $x_i^* = 1$ is persistent. Similar results can of course be derived when $x_i^* = 0$ is persistent. We summarize the results in Table 1.

To reduce the number of variables in f we use generalized roof duality to determine persistencies of $\min_{x_k=1} \phi(\boldsymbol{x}, x_0)$. Depending on which of the variables we fix to be one, different reductions can be obtained. Our approach is to go through all the possible fixations systematically and reduce f as soon as a persistency is obtained. We summarize our algorithm below.

1. Construct $\phi(\boldsymbol{x}, x_0)$ and set lower bound $l := -\infty$.

Table 1: Persistency in $\min_{x_k=1} \phi(\boldsymbol{x}, x_0)$ and the resulting reductions in f

persistency of x_0^*	persistency of x_i^*	reductions in $f(\boldsymbol{z})$
-	$x_i^* = 1$	$z_k = z_i$
-	$x_i^* = 0$	$z_k = \bar{z}_i$
$x_0^* = 0$	$x_i^* = 1$	$z_k = z_i = 0$
$x_0^* = 0$	$x_i^* = 0$	$z_k = \bar{z}_i = 0$
$x_0^* = 1$	$x_i^* = 1$	$z_k = z_i = 1$
$x_0^* = 1$	$x_i^* = 0$	$z_k = \bar{z}_i = 1$

2. For $k = 0, \ldots, n$ do
   (i) Compute persistencies and lower bound l_k of $\min_{x_k=1} \phi(\boldsymbol{x}, x_0)$.
   (ii) Reduce f using the persistencies and Table 1.
   (iii) Update the lower bound $l := \max(l, l_k)$.

The fixation $x_k = 1$ for $k = 0$ corresponds to the original function f as $\phi(\boldsymbol{x}, 1) = f(\boldsymbol{x})$ and therefore the procedure will always give at least as many persistencies as the usual procedure. Note that no additional persistencies are obtained if one were to fixate $x_k = 0$ due to symmetry.

5 Experiments

In this section we will describe some challenging optimization problems in order to test and compare the performance of the proposed methods. We will use the methods listed in Table 2.

Table 2: Abbreviations for the different methods

RD	Standard roof duality [1]
GRD	Generalized Roof Duality (GRD) as in [12]
GRD-gen	GRD using generators (Section 3)
GRD-ext	GRD-gen in combination with symmetric extension (Section 4)
Fix et al.	The reductions proposed in [25]
HOCR	The reductions proposed in [22]

5.1 Segmentation with Curvature Regularization

We first present a segmentation experiment where higher order cliques model the curvature regularization. A discretized version of the following energy is used:

$$E(S) = \int_S f(\boldsymbol{x})\ dx + \int_{\partial S} (\rho + \sigma\kappa(s)^2) ds. \qquad (21)$$

Here $f(\boldsymbol{x})$ is the cost of assigning $\boldsymbol{x}$ to the interior of S and the second term is a combined length and curvature regularizer. We will use the pseudo-boolean optimization approach suggested in [5] for an 8-connected grid which requires quartic interactions. The construction can be understood by examining Figure 1. The boolean variables x_a,x_b,x_c and x_d are assigned interior or exterior. The two arrows will contribute to the curvature if and only if both of them are on the boundary, that is, $x_a \neq x_b$ and $x_c \neq x_d$. This can be encoded using the quartic term

$$b_{ij}\Big(x_a x_c(1-x_b)(1-x_d) + (1-x_a)(1-x_c)x_b x_d\Big), \tag{22}$$

where b_{ij} is the contributed curvature penalization.

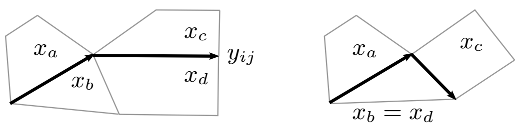

Fig. 1: Examples of four incident region variables x_a, x_b, x_c and x_d in an arbitrary mesh. The region variables may coincide for some edge pairs.

We use the cameraman as a test image, Figure 2. The unary data costs for the foreground and background are set to $\lambda(1 - I(\boldsymbol{x}))$ and $\lambda I(\boldsymbol{x})$, respectively, where $I(\boldsymbol{x}) \in [0, 1]$ is the gray scale value at position $\boldsymbol{x}$ and $\lambda = 75$. The length and curvature weights are set to $\rho = 1$ and $\sigma = \{1, 2\}$, respectively. Experimental data are collected in Table 3.

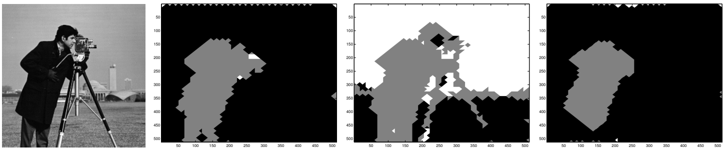

Fig. 2: Input image and the results for GRD, GRD-gen, and Fix et al. Unlabeled variables are colored black. The HOCR method returned no assigned variables.

.

None of the methods are able to produce a complete labeling which indicates the difficulty of the problem. The resulting (incomplete) segmentations for $\sigma = 1$ are plotted in Figure 2. Note that GRD-gen outperforms the competitors with more than 3 times as many assigned variables for $\sigma = 1$ and it is the only method to assign labels for $\sigma = 2$. While the runtime for GRD-gen was much faster than GRD it was considerably slower than both of the other methods. The GRD-gen method has potential for this problem but the approximation tightness needs to be improved in order to beat state-of-the-art based on LP relaxations [5].

Table 3: Results for the curvature experiments

$\sigma = 1$	Assigned variables	Runtime [s]	Lower bounds
GRD	20.8%	5050	4.24×10^4
GRD-gen	76.5%	1080	8.29×10^4
Fix et al.	16.0%	1.00	4.54×10^4
HOCR	0.00%	1.00	-7.42×10^4
$\sigma = 2$	Assigned variables	Runtime [s]	Lower bounds
GRD	0.00%	10300	-0.48×10^5
GRD-gen	34.4%	4995	0.719×10^5
Fix et al.	0.00%	1.00	-0.38×10^5
HOCR	0.00%	1.00	-2.872×10^5

5.2 Synthetic Data

In the final experiment, we test the various methods on synthetically generated polynomials with random coefficients:

$$f(\boldsymbol{x}) = \sum_{(i,j,k,l)\in T} f_{ijkl}(x_i, x_j, x_k, x_l), \tag{23}$$

where $T \subseteq \{1 \ldots n\}^4$ is a random set of quadruples and each f_{ijkl} is a fourth degree polynomial with its coefficient picked uniformly from $[-100, 100]$. The persistency results for problem instances with $n = 1000$, $|T| = \{50, 100, 200, 300\}$ are given in Table 4. The persistency distributions for $n = 1000$, $|T| = 300$ are also visualized in Figure 3. Note that the results for GRD-gen and GRD-ext are similar and therefore only GRD-gen is present in the left diagram. We also compare the relative lower bounds $(\ell - \ell_{GRD})/|\ell_{GRD}|$, where ℓ_{GRD} is the generalized roof dual bound for $f(\boldsymbol{x})$, see Table 5. The relative lower bounds follow the same trend as the persistency, GRD-gen and GRD-ext give similar lower bounds and significantly better than GRD.

Table 4: Results for the synthetic experiments

Assigned variables	$\lvert T\rvert = 50$	Runtime [s]	$\lvert T\rvert = 100$	Runtime [s]	$\lvert T\rvert = 300$	Runtime [s]
GRD-ext	74.5%	3.95	73.6%	17.9	68.3%	430
GRD-gen	73.6%	0.06	72.9%	0.10	66.5%	0.61
GRD	59.1%	0.06	56.3%	0.12	48.0%	1.01
Fix et al.	34.8%	0.00	33.2%	0.00	23.7%	0.01
HOCR	23.0%	0.00	21.4%	0.00	14.5%	0.01

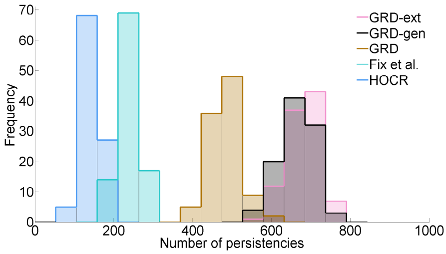

Fig. 3: Average persistency for the synthetic experiments with $|T| = 300$.

Table 5: Relative lower for the synthetic experiments with $|T| = 300$.

Rel. bounds	Min	Median	Max
GRD-ext	5.9%	7.7%	10.1%
GRD-gen	5.6%	7.5%	9.4%
GRD	0.0%	0.0%	0.0%
Fix et al.	−14.0%	−16.3%	−18.0%
HOCR	−44.6%	−48.9%	−52.6%

6 Discussion

Two new methods have been investigated with the objective to improve the performance of the well-known roof dual bound for pseudo-boolean optimization. We have experimentally demonstrated that (i) constructing submodular relaxations using generators significantly outperforms previously published methods and that (ii) applying the roof dual in an iterated manner does lead to stronger bounds and more persistencies, but may not be worth-while for the problems considered unless the extra computational cost can be drastically reduced.

We are currently working on improving the running times for the two methods. Most of the time is spent on the LP for constructing the relaxation. In [12], a heuristic scheme for GRD which completely avoids the LP is presented without any significant loss in relaxation performance. Essentially the same heuristics can be applied to GRD based on generators. Further, the max-flow computations in each step of the extension method are typically very similar. Therefore, reusing flows is a likely to speed up the computations.

Acknowledgments. This work was funded by the Swedish Foundation for Strategic Research through the programmes *Future Research Leaders* and *Wearable Visual Information Systems* and by the European Research Council through the programme *GlobalVision*.

References

1. Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts - a review. IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007) 1274–1279
2. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Int. Conf. Computer Vision, Nice, France (2003)
3. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23** (2001) 1222–1239
4. Schoenemann, T., Kahl, F., Cremers, D.: Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In: Int. Conf. Computer Vision, Kyoto, Japan (2009)
5. Strandmark, P., Kahl, F.: Curvature regularization for curves and surfaces in a global optimization framework. In: EMMCVPR, St Petersburg, Russia (2011)

6. Lan, X., Roth, S., Huttenlocher, D., Black, M.: Efficient belief propagation with learned higher-order Markov random fields. In: European Conf. Computer Vision, Graz, Austria (2006)
7. Woodford, O., Torr, P.H., Reid, I., Fitzgibbon, A.: Global stereo reconstruction under second-order smoothness priors. IEEE Trans. Pattern Analysis and Machine Intelligence **31** (2009) 2115–2128
8. Cremers, D., Grady, L.: Statistical priors for efficient combinatorial optimization via graph cuts. In: European Conf. Computer Vision, Graz, Austria (2006)
9. Nowozin, S., Lampert, C.H.: Global interactions in random field models: A potential function ensuring connectedness. SIAM J. Imaging Sciences **3** (2010) 1048–1074
10. Kolmogorov, V.: Generalized roof duality and bisubmodular functions. In: Neural Information Processing Systems. (2010)
11. Kahl, F., Strandmark, P.: Generalized roof duality for pseudo-boolean optimization. In: Int. Conf. Computer Vision, Barcelona, Spain (2011)
12. Kahl, F., Strandmark, P.: Generalized roof duality. Discrete Applied Mathematics **160** (2012) 2419–2434
13. Crama, Y.: Recognition problems for special classes of polynomials in 0-1 variables. Math. Programming **44** (1989) 139–155
14. Boros, E., Hammer, P., Sun, R., Tavares, G.: A max-flow approach to improved lower bounds for quadratic unconstrained binary optimization (QUBO). Discrete Optimization **5** (2008) 501–529
15. Komodakis, N., Paragios, N.: Beyond pairwise energies: Efficient optimization for higher-order MRFs. In: Conf. Computer Vision and Pattern Recognition, Miami, USA (2009)
16. Strandmark, P., Kahl, F.: Parallel and distributed graph cuts by dual decomposition. In: Conf. Computer Vision and Pattern Recognition, San Francisco, USA (2010)
17. Kohli, P., Kumar, M.P., Torr, P.: P^3 & beyond: Move making algorithms for solving higher order functions. IEEE Trans. Pattern Analysis and Machine Intelligence **31** (2009) 1645–1656
18. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for Markov random field optimization. IEEE Trans. Pattern Analysis and Machine Intelligence **32** (2010) 1392–1405
19. Werner, T.: High-arity interactions, polyhedral relaxations, and cutting plane algorithm for MAP-MRF. In: Conf. Computer Vision and Pattern Recognition, Anchorage, USA (2008)
20. Lu, S., Williams, A.: Roof duality for polynomial 0-1 optimization. Math. Programming **37** (1987) 357–360
21. Freedman, D., Drineas, P.: Energy minimization via graph cuts: Settling what is possible. In: Conf. Computer Vision and Pattern Recognition, San Diego, USA (2005)
22. Ishikawa, H.: Transformation of general binary MRF minimization to the first order case. IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 1234–1249
23. Promislow, S.D., Young, V.R.: Supermodular functions on finite lattices. Order **22** (2005) 389–413
24. Živný, S., Cohen, D.A., Jeavons, P.G.: The expressive power of binary submodular functions. Discrete Appl. Math. **157** (2009) 3347–3358
25. Fix, A., Grubner, A., Boros, E., Zabih, R.: A graph cut algorithm for higher-order markov random fields. In: Int. Conf. Computer Vision, Barcelona, Spain (2011)