High-arity Interactions, Polyhedral Relaxations, and Cutting Plane Algorithm for Soft Constraint Optimisation (MAP-MRF)

Tomáš Werner
Center for Machine Perception, Department of Cybernetics, Czech Technical University
Karlovo náměstí 13, 12135 Praha, Czech Republic

Abstract

LP relaxation approach to soft constraint optimisation (i.e. MAP-MRF) has been mostly considered only for binary problems. We present its generalisation to n-ary problems, including a simple algorithm to optimise the LP bound, n-ary max-sum diffusion. As applications, we show that a hierarchy of gradually tighter polyhedral relaxations of MAP-MRF is obtained by adding zero interactions. We propose a cutting plane algorithm, where cuts correspond to adding zero interactions and the separation problem to finding an unsatisfiable constraint satisfaction subproblem. Next, we show that certain high-arity interactions, e.g. certain global constraints, can be included into the framework in a principled way. Finally, we prove that n-ary max-sum diffusion finds global optimum for n-ary supermodular problems.

1. Introduction

Computing[1] maxima of a Gibbs probability distribution (MAP inference in MRF) is widely recognised as useful in computer vision [20]. It is also known as *soft (valued) constraint optimisation* [12]. One of the approaches to this NP-hard combinatorial optimisation problem is the linear programming (LP) relaxation formulated by Schlesinger [16]. Kovalevsky and Koval [10] proposed a simple network algorithm, *max-sum diffusion*, to solve (though suboptimally) this LP relaxation. Works [16, 10] are surveyed in [24, 25].

An equivalent result was obtained by Kolmogorov [7] who, using the convex upper bound by Wainwright et al. [23], proposed the *sequential tree-reweighted max-product* (TRW-S) algorithm. Max-sum diffusion yields the same bound as TRW-S but, since it uses edge updates, is slower. TRW-S outperforms other algorithms for MAP-MRF on sparse graphs on a number of computer vision tasks [20].

The same LP relaxation was independently proposed also in [5] (but only for two-state variables), in [8] (but without considering LP dual) and recently in [2].

We present the following contributions to MAP-MRF:

[1]I thank the European Commission grant 215078 (DIPLECS) and the Czech government grant MSM6840770038 for support.

N-ary generalisation of LP relaxation approach (§3–5). The LP relaxation [16, 25] and max-sum diffusion were formulated primarily for problems with binary interactions. We generalise these to interactions of arbitrary arity.

Tighter relaxations (§6). Central to a combinatorial optimisation problem is the convex hull of the feasible set of its suitable integer LP formulation (integral hull). Following [23], we will refer to the integral hull of MAP-MRF as the *marginal polytope*. The above LP relaxation can be seen as optimising over a tractable outer bound of this polytope.

We show that gradually tighter outer bounds of the marginal polytope can be obtained simply by *adding zero interactions*. From the polyhedral view, this corresponds to lifting, adding marginalisation constraints, and projection. We propose a cutting plane algorithm which adds zero interactions dynamically and in which finding a suitable zero interaction (i.e., the separation problem) means identifying an *unsatisfiable constraint satisfaction subproblem*.

Several approaches were proposed before to tightening the existing LP-based outer bound of the marginal polytope. Koster et al. [8] gave two classes of non-trivial facets of the marginal polytope. Independently, Sontag and Jaakkola [19] proposed tighter bounds based on the relation of the marginal polytope and the *cut polytope* [4], for which several classes of non-trivial facets are known. Besides MAP-MRF, [19] addresses also approximation of marginals and partition function of a MRF. Both works [8, 19] propose also a cutting plane algorithm. Our approach is considerably simpler, in particular it avoids reference to the cut polytope. Moreover, while [8, 19] work explicitly in the space of the marginal polytope (i.e. with pseudomarginals), we show that it is more natural to consider the dual LP, which can be optimised by n-ary max-sum diffusion.

Wainwright [23, §VI] also mentioned that progressively tighter relaxations of MAP-MRF could be achieved by combining hypertrees rather than trees. But this has never been done with the sequential modification [7] that ensured convergence and lead to TRW-S. Moreover, by using hyperedges only, we bypass the (complex) hypertree formalism.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

Global interactions (§7). Any interaction, possibly of a high arity and represented by an algorithm, can be naturally handled by n-ary max-sum diffusion whenever certain optimisation problem induced by the interaction is tractable. We demonstrate this on several *global interactions*.

Rother et al. [13] also used a global constraint in MAP-MRF. To compare, they solve a more complex task but we incorporate global constraints in a principled way.

Supermodular n-ary problems (§8). We prove that for n-ary supermodular interactions, any fixed point of n-ary max-sum diffusion is a global optimum of MAP-MRF.

In the sequel, 2^V resp. $\binom{V}{k}$ denotes the set of all resp. of k-element subsets of set V. Indicator $[\![\omega]\!]$ equals 1 if logical expression ω is true and 0 if it is false. $\mathbb{R}$ denotes the reals, $\mathbb{R}_+$ the non-negative reals, and $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty\}$.

2. Problem formulation

Let V be a finite set of variables. Each variable $v \in V$ attains states $x_v \in X_v$, where X_v is a finite *domain* of varisble v. For $A \subseteq V$, the Cartesian product $X_A = \bigtimes_{v \in A} X_v$ is the *joint domain* of variables A. To fix the order of the factors in this product, we assume V is endowed with an (arbitrary) total order. A *joint state* of variables A is a tuple $x_A \in X_A$. For $B \subseteq A$, symbols x_A and x_B appearing in a single expression do not denote independent joint states but x_B is the *restriction* of joint state x_A to variables B.

Let $E \subseteq 2^V$ be a set of subsets of V, i.e., an *undirected hypergraph*. Each joint state x_A of each hyperedge A is assigned a weight $\theta_A(x_A) \in \bar{\mathbb{R}}$. All these weights together will be denoted by vector $\boldsymbol{\theta} \in \bar{\mathbb{R}}^{T(E)}$ where

$$T(E) = \{\, (A, x_A) \mid A \in E,\ x_A \in X_A \,\}$$

Alternatively, symbol θ_A can be understood as denoting a function $X_A \to \bar{\mathbb{R}}$. The topic of this paper is the combinatorial optimisation problem[2]

$$\max_{x_V} \sum_{A \in E} \theta_A(x_A) \tag{1}$$

Its instance is defined by triplet $(E, X_V, \boldsymbol{\theta})$. We will refer to θ_A as an *interaction*, to $|A|$ as the *arity* of interaction θ_A, and to $\max_{A \in E} |A|$ as the arity of the problem.

Example. Let $V = \{1, 2, 3, 4\}$ with natural ordering on it. Let $E = \{\{2, 3, 4\}, \{1, 2\}, \{3, 4\}, \{3\}\}$. Problem (1) reads

$$\max_{x_1, x_2, x_3, x_4} [\theta_{234}(x_2, x_3, x_4) + \theta_{12}(x_1, x_2) + \theta_{34}(x_3, x_4) + \theta_3(x_3)]$$

where e.g. θ_{234} denotes a ternary function of variables $2, 3, 4$ and symbol $\theta_{234}(x_{234}) = \theta_{234}(x_2, x_3, x_4)$ denotes the value of this function evaluated at given states x_2, x_3, x_4 of these variables.

[2] Strictly speaking, problem (1) is more general than MAP-MRF because we allow arbitrary hyperedges (not only those induced by cliques of an ordinary graph) and we allow that some weights equal $-\infty$.

3. LP relaxation for n-ary problems

In §3, we generalise the LP relaxation approach to (1) by Schlesinger [16], originally formulated for binary problems, to n-ary problems. Doing that, we will closely follow the author's survey [25] of this approach.

Central to the generalisation is pair (2) of linear programs (see the next page). The left-hand (primal) program is an LP relaxation of (1) and the right-hand program is its dual. Duality is demonstrated more transparently by writing the pair in a matrix form,

$$\begin{aligned}
\boldsymbol{\theta}^\top \boldsymbol{\mu} &\to \max & \boldsymbol{\psi}^\top \mathbf{1} &\to \min && \text{(3a)}\\
\mathbf{M}\boldsymbol{\mu} &= \mathbf{0} & \boldsymbol{\varphi} &\in \mathbb{R}^J && \text{(3b)}\\
\mathbf{N}\boldsymbol{\mu} &= \mathbf{1} & \boldsymbol{\psi} &\in \mathbb{R}^E && \text{(3c)}\\
\boldsymbol{\mu} &\in \mathbb{R}_+^{T(E)} & \boldsymbol{\varphi}^\top \mathbf{M} + \boldsymbol{\psi}^\top \mathbf{N} &\geq \boldsymbol{\theta}^\top && \text{(3d)}
\end{aligned}$$

Besides E, $\{\, X_v \mid v \in V \,\}$ and $\boldsymbol{\theta}$, the pair has one more free parameter, J. It is a set such that $J \subseteq I(E)$ where

$$I(E) = \{\, (A, B, x_B) \mid A \in E,\ B \in E,\ B \subset A,\ x_B \in X_B \,\}$$

For simplicity of exposition, we assume that J satisfies

$$(A, B, x_B) \in J \implies \forall x_B \in X_B \colon (A, B, x_B) \in J \tag{4}$$

Relaxing assumption (4) is sometimes useful, and is possible under mild conditions. But we keep it in the paper.

Further in §3 we will interpret the primal and the dual.

3.1. The primal program

The primal variables $\boldsymbol{\mu}$ represent a collection of functions $\mu_A \colon X_A \to \mathbb{R}_+$, one for each hyperedge. For $A \in E$, constraints (2c)+(2d) enforce that μ_A is a probability distribution on joint states X_A. Marginalisation constraint (2b) couples pairs of distributions[3], enforcing that for each triplet (called a *pencil*) $(A, B, x_B) \in J$, number $\mu_B(x_B)$ is the marginal of μ_A. The set J determines which of all $I(E)$ possible marginalisation constraints are imposed.

If $\boldsymbol{\mu}$ is integral, i.e. $\boldsymbol{\mu} \in \{0, 1\}^{T(E)}$, each distribution μ_A represents a single joint state $x_A \in X_A$ and the marginalisation constraint (2b) represents the restriction of x_A to variables B. Thus, an integral primal feasible $\boldsymbol{\mu}$ represents a set of joint states, each for one hyperedge, that are incident wherever two hyperedges are coupled by marginalisation. The primal becomes an integer (0-1) LP, which is equivalent to (1) provided that each variable (i.e., one-element hyperedge) is coupled with some hyperedge, i.e., if

$$\forall A \in E,\ v \in V,\ x_v \in X_v \colon\ (A, \{v\}, x_v) \in J \tag{5}$$

The LP (2) is the continuous relaxation of this integer LP.

We will say more about the primal in §6.

[3] Alternatively, the marginalisation constraint could be formulated in a more general form, $\sum_{x_C} \mu_A(x_A) = \sum_{x_C} \mu_B(x_B)$ for $C \subseteq A \cap B$ and $x_C \in X_C$, and the dual program modified accordingly.

$$\sum_{A\in E}\sum_{x_A}\theta_A(x_A)\,\mu_A(x_A)\to\max \qquad\qquad \sum_{A\in E}\psi_A\to\min \tag{2a}$$

$$\sum_{x_{A\setminus B}}\mu_A(x_A)=\mu_B(x_B) \qquad\qquad \varphi_{A,B}(x_B)\in\mathbb{R} \qquad (A,B,x_B)\in J \tag{2b}$$

$$\sum_{x_A}\mu_A(x_A)=1 \qquad\qquad \psi_A\in\mathbb{R} \qquad A\in E \tag{2c}$$

$$\mu_A(x_A)\geq 0 \qquad\qquad \theta_A(x_A)+\sum_{B|(A,B,x_B)\in J}\varphi_{A,B}(x_B)-\sum_{B|(B,A,x_A)\in J}\varphi_{B,A}(x_A)\leq\psi_A \qquad (A,x_A)\in T(E) \tag{2d}$$

3.2. The dual program

Weight vectors $\boldsymbol{\theta},\boldsymbol{\theta}'\in\bar{\mathbb{R}}^{T(E)}$ are called *equivalent* if they give rise to the same objective function of problem (1), i.e. if $\sum_{A\in E}\theta_A(x_A)=\sum_{A\in E}\theta'_A(x_A)$ for all $x_V\in X_V$. A transformation mapping a weight vector to its equivalent is called an *equivalent transformation* [16, 25].

The *elementary equivalent transformation* is applied on a single pencil $(A,B,x_B)\in J$ as follows: add a constant[4] $\varphi_{A,B}(x_B)$ to the weights $\{\,\theta_A(x_A)\mid x_{A\setminus B}\in X_{A\setminus B}\,\}$ and subtract the same constant from $\theta_B(x_B)$. Applying these transformations on all $(A,B,x_B)\in J$ yields

$$\theta^{\boldsymbol{\varphi}}_A(x_A)=\theta_A(x_A)+\sum_{B|(A,B,x_B)\in J}\varphi_{A,B}(x_B)-\sum_{B|(B,A,x_A)\in J}\varphi_{B,A}(x_A) \tag{6}$$

where $\boldsymbol{\theta}^{\boldsymbol{\varphi}}=\boldsymbol{\theta}-\mathbf{M}^\top\boldsymbol{\varphi}$ denotes the equivalent of $\boldsymbol{\theta}$ corresponding to vector $\boldsymbol{\varphi}\in\mathbb{R}^J$. The matrix form (3) shows clearly why equivalent transformations preserve the primal objective: since $\mathbf{M}\boldsymbol{\mu}=\mathbf{0}$ implies $(\boldsymbol{\theta}^\top-\boldsymbol{\varphi}^\top\mathbf{M})\boldsymbol{\mu}=\boldsymbol{\theta}^\top\boldsymbol{\mu}$.

Since $\max_i\sum_j a_{ij}\leq\sum_j\max_i a_{ij}$ for any a_{ij}, we have

$$\max_{x_V}\sum_{A\in E}\theta_A(x_A)\leq\sum_{A\in E}\max_{x_A}\theta_A(x_A) \tag{7}$$

thus the right-hand expression is an upper bound on (1). By eliminating variables ψ_A, the dual in (2) can be written as

$$\min_{\boldsymbol{\varphi}\in\mathbb{R}^J}\sum_{A\in E}\max_{x_A}\theta^{\boldsymbol{\varphi}}_A(x_A) \tag{8}$$

which can be interpreted as minimising the upper bound (7) over the equivalents of $\boldsymbol{\theta}$.

4. When is the bound exact?

The bound is exact, i.e. (7) holds with equality, if and only if maximisers taken separately for each hyperedge agree on a common solution[5], i.e., if there exists $x_V\in X_V$ such that $\theta_A(x_A)=\max_{y_A}\theta_A(y_A)$ for all $A\in E$. As shown in [16, 25], this can be translated to the well-known constraint satisfaction problem as follows.

Given an indicator vector $\boldsymbol{\sigma}\in\{0,1\}^{T(E)}$, the *constraint satisfaction problem* (CSP) [11] represented by $(E,X_V,\boldsymbol{\sigma})$ seeks to find a joint state $x_V\in X_V$ such that

$$\min_{A\in E}\sigma_A(x_A)=1$$

If such x_V exists, the CSP instance is *satisfiable*. The CSP is NP-complete but proving or disproving satisfiability is tractable for its certain subclasses.

Joint state (A,x_A) such that $\theta_A(x_A)=\max_{y_A}\theta_A(y_A)$ is called *active*. Let $\sigma_A(x_A)=1$ if (A,x_A) is active and $\sigma_A(x_A)=0$ if it is inactive. Then (7) holds with equality if and only if the CSP instance $(E,X_V,\boldsymbol{\sigma})$ is satisfiable.

5. Decreasing the upper bound

Max-sum diffusion [10, 25] is a very simple network algorithm to decrease the upper bound (7) by equivalent transformations (6). While it was originally defined for binary problems, in §5 we generalise it to n-ary problems.

The single iteration of the algorithm is the elementary equivalent transformation (see §3.2) that makes the equality

$$\max_{x_{A\setminus B}}\theta_A(x_A)=\theta_B(x_B) \tag{9}$$

satisfied for a single pencil (A,B,x_B), which is done by letting $\varphi_{A,B}(x_B)=[\theta_B(x_B)-\max_{x_{A\setminus B}}\theta_A(x_A)]/2$. This improves the bound (7) in the sense of Theorem 1.

Theorem 1. *Let $A,B\in E$ and $B\subset A$ be fixed. The iteration done on the pencils $\{\,(A,B,x_B)\mid x_B\in X_B\,\}$ (in any order) does not increase the upper bound*[6].

Proof. Since A and B are fixed, we can denote for brevity $a(x_B)=\max_{x_{A\setminus B}}\theta_A(x_A)$, $b(x_B)=\theta_B(x_B)$, and $c(x_B)=[b(x_B)-a(x_B)]/2$. Before the iterations, the contribution of hyperedges A and B to the bound (7) is

$$\max_{x_A}\theta_A(x_A)+\max_{x_B}\theta_B(x_B)=\max_{x_B}a(x_B)+\max_{x_B}b(x_B) \tag{10}$$

After the iterations, this contribution is

$$\begin{aligned}&\max_{x_B}[a(x_B)+c(x_B)]+\max_{x_B}[b(x_B)-c(x_B)]\\ &=\max_{x_B}[a(x_B)+b(x_B)]\end{aligned} \tag{11}$$

[4] In works developed from belief propagation, e.g. [23, 7], equivalent transformations are called *reparameterisations* and $\varphi_{A,B}(x_B)$ *messages*.

[5] This can be seen as a reformulation of *(hyper)tree agreement* [23] for (hyper)trees being individual (hyper)edges.

[6] But the iteration on a *single* pencil may increase the upper bound.

Clearly, expression (11) is not greater than (10). ∎

The complete algorithm then looks as follows.

Algorithm 1. (n-ary max-sum diffusion)

loop
  for $(A, B, x_B) \in J$ **do**
    $\varphi_{A,B}(x_B) \mathrel{+}= [\theta^{\boldsymbol{\varphi}}_B(x_B) - \max_{x_{A\setminus B}} \theta^{\boldsymbol{\varphi}}_A(x_A)]/2$
  end for
end loop

The algorithm converges to a state when (9) holds for all $(A, B, x_B) \in J$. As shown in [25] (but only for binary problems), it follows that the CSP formed by the active joint states $\boldsymbol{\sigma}$ is *arc consistent*[7] (AC) [1], i.e., it satisfies

$$\max_{x_{A\setminus B}} \sigma_A(x_A) = \sigma_B(x_B), \quad (A, B, x_B) \in J \tag{12}$$

In general, the fixed point of the algorithm need not be a global minimum of (2); examples were given in [15, 25, 7]. As pointed out in [26], this is because Algorithm 1 is a coordinate descent method, which need not find the global minimum of a *nonsmooth* convex function, here function (7). Despite this, global minimum is often found in practice.

The only non-trivial part of the algorithm is the calculation of $\max_{x_{A\setminus B}} \theta^{\boldsymbol{\varphi}}_A(x_A)$. This means to solve a *smaller instance of problem (1)*, induced by hyperedge A (more precisely, to compute its max-marginals associated with B). This instance is specified by triplet $(E', X_A, \boldsymbol{\theta}')$ where E' and $\boldsymbol{\theta}'$ are obtained from (6) – in particular, hypergraph E' consists of hyperedge A and its subsets involved in J.

For a large arity of interaction θ_A, the number of primal variables and dual constraints on line (2d) is intractable. Importantly, this is not an obstacle to use the algorithm provided that calculating $\max_{x_{A\setminus B}} \theta^{\boldsymbol{\varphi}}_A(x_A)$ is tractable.

For binary problems, Algorithm 1 can be shown equivalent to TRW-S [7] with the trees being individual edges. It is known that using edge updates is slower than using tree updates [23, 7]. Thus, e.g. on images Algorithm 1 is typically several times slower than TRW-S on monotonic chains with the chains being image rows and columns [7].

6. A hierarchy of polyhedral relaxations

Consider adding a hyperedge $A \notin E$ to hypergraph E and setting the interaction θ_A identically to zero. This does not change the original problem (1) but it may improve its LP relaxation (2). The rest of §6 formalises this observation.

6.1. The marginal polytope

Let mapping $\boldsymbol{\delta}_E$: $X_V \to \{0,1\}^{T(E)}$ be defined by: the (A, y_A)-component of vector $\boldsymbol{\delta}_E(x_V)$ equals $[\![x_A = y_A]\!]$, i.e., it equals 1 iff y_A is the restriction of x_V to variables A. This lets us rewrite the objective of (1) as a scalar product,

$$\sum_{A\in E} \theta_A(x_A) = \sum_{A\in E} \sum_{y_A} \theta_A(y_A)\, [\![x_A = y_A]\!] = \boldsymbol{\theta}^\top \boldsymbol{\delta}_E(x_V)$$

Problem (1) can be reformulated by the equalities

$$\max_{x_V} \sum_{A\in E} \theta_A(x_A) = \max \{\, \boldsymbol{\theta}^\top \boldsymbol{\delta}_E(x_V) \mid x_V \in X_V \,\} \tag{13a}$$
$$= \max \{\, \boldsymbol{\theta}^\top \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \boldsymbol{\delta}_E(X_V) \,\} \tag{13b}$$
$$= \max \{\, \boldsymbol{\theta}^\top \boldsymbol{\mu} \mid \boldsymbol{\mu} \in \operatorname{conv} \boldsymbol{\delta}_E(X_V) \,\} \tag{13c}$$

where $\boldsymbol{\delta}_E(X_V) = \{\, \boldsymbol{\delta}_E(x_V) \mid x_V \in X_V \,\}$. Equality (13c) follows from the fact that a linear program attains its optimum at least one vertex. Note that (13c) formulates problem (1) as a linear program over polytope $\operatorname{conv} \boldsymbol{\delta}_E(X_V)$.

Denoting the coefficients of convex combination by μ_V, an element $\boldsymbol{\mu}$ of the polytope $\operatorname{conv} \boldsymbol{\delta}_E(X_V)$ is given by

$$\boldsymbol{\mu} = \sum_{x_V} \mu_V(x_V)\, \boldsymbol{\delta}_E(x_V) \tag{14}$$

where $\mu_V(x_V) \geq 0$ and $\sum_{x_V} \mu_V(x_V) = 1$. Substituting for $\boldsymbol{\delta}_E(x_V)$ yields that the (A, y_A)-component of $\boldsymbol{\mu}$ equals

$$\mu_A(y_A) = \sum_{x_V} \mu_V(x_V)\, [\![x_A = y_A]\!] = \sum_{y_{V\setminus A}} \mu_V(y_V) \tag{15}$$

where the second equality is easily verified. Thus, $\boldsymbol{\mu} \in \operatorname{conv} \boldsymbol{\delta}_E(X_V)$ if and only if $\boldsymbol{\mu}$ are the marginals of some distribution μ_V. For this reason, $\operatorname{conv} \boldsymbol{\delta}_E(X_V)$ is referred to as the *marginal polytope* (associated with E) in [23].

This shows that the marginal polytope can be seen as a *projection* of a larger polytope, living in a space that besides dimensions $\{\, \mu_A \mid A \in E \,\}$ includes also dimensions μ_V:

$$\operatorname{conv} \boldsymbol{\delta}_E(X_V) = \pi_{T(E)}[P(\, E \cup \{V\},\, I(E \cup \{V\}\,))] \tag{16a}$$
$$= \pi_{T(E)}[P(2^V, I(2^V))] \tag{16b}$$

where $P(E, J)$ denotes the polytope of vectors $\boldsymbol{\mu}$ satisfying primal constraints (2b)+(2c)+(2d) and $\pi_{D'}(Y) \in \mathbb{R}^{D'}$ denotes the projection of set $Y \subseteq \mathbb{R}^D$ on dimensions $D' \subseteq D$. Equality (16a) is obtained by rewriting (15) using the projection. Equality (16b) is true because extra dimensions are just removed by the projection. Note, $P(2^V, I(2^V))$ is the marginal polytope of the complete hypergraph 2^V.

6.2. Outer bounds of the marginal polytope

Let us have $\bar{E}_1$, J_1, $\bar{E}_2$, J_2 such that $E \subseteq E_1 \subseteq 2^V$, $J_1 \subseteq I(E_1)$, $E \subseteq E_2 \subseteq 2^V$, and $J_2 \subseteq I(E_2)$. Then

$$J_2 \subseteq J_1 \implies \pi_{T(E)}[P(\bar{E}_1, J_1))] \subseteq \pi_{T(E)}[P(\bar{E}_2, J_2))] \tag{17}$$

To see (17), see that $\pi_{T(E)}[P(\bar{E}, J))] = \pi_{T(E)}[P(2^V, J))]$ because the dimensions not coupled with any other dimensions (i.e., not participating in J) are removed by the pro-

[7] Often, the term *arc consistency* is used only for binary problems and for n-ary problems the term *generalised arc consistency* is used [1]. Binary arc consistency can be shown equivalent to *weak tree agreement* [7].

jection without any effect. Now, see that $J_2 \subseteq J_1$ implies $\pi_{T(E)}[P(2^V, J_1))] \subseteq \pi_{T(E)}[P(2^V, J_2))]$ because the latter polytope is less constrained.

By (16b) and (17), for any $\bar{E}$ and J such that

$$E \subseteq \bar{E} \subseteq 2^V, \quad J \subseteq I(\bar{E}) \tag{18}$$

the polytope $\pi_{T(E)}[P(\bar{E}, J))]$ is an *outer bound of the marginal polytope*. This polytope is defined by $\bar{E}$ and J, but it is in fact fully determined by J alone because once an arbitrary $J \subseteq I(2^V)$ is chosen, taking any (e.g. the smallest) hypergraph $\bar{E}$ satisfying (18) yields the same polytope[8].

Note, two sets $J_1 \neq J_2$ may sometimes yield the same relaxation, $\pi_{T(E)}[P(\bar{E}_1, J_1)] = \pi_{T(E)}[P(\bar{E}_2, J_2)]$. Characterising when exactly this happens is an open problem.

Implication (17) further establishes that the outer bounds $\pi_{T(E)}[P(\bar{E}, J))]$ form a *hierarchy*, partially ordered by inclusion on $I(2^V)$. The extreme choices are $J = I(2^V)$ for exact solution and $J = \emptyset$ for maximising each interaction independently. In between, there is a range of intermediate relaxations. One of them is the LP relaxation [16, 8, 23], obtained for $J = I(E)$; then $\pi_{T(E)}[P(\bar{E}, J))] = P(E, I(E))$ is called the *local polytope* [23].

6.3. Projection = adding zero interactions

By (13c), for any $\bar{E}$ and J satisfying (18),

$$\max \left\{ \boldsymbol{\theta}^\top \boldsymbol{\mu} \,\middle|\, \boldsymbol{\mu} \in \pi_{T(E)}[P(\bar{E}, J)] \right\} \tag{19}$$

is an upper bound on (1). But (19) is clearly equal to

$$\max \left\{ \bar{\boldsymbol{\theta}}^\top \bar{\boldsymbol{\mu}} \,\middle|\, \bar{\boldsymbol{\mu}} \in P(\bar{E}, J) \right\} \tag{20}$$

where $\bar{\boldsymbol{\theta}} \in \bar{\mathbb{R}}^{T(\bar{E})}$ is given by

$$\bar{\theta}_A(x_A) = \begin{cases} \theta_A(x_A) & \text{if } A \in E \\ 0 & \text{if } A \in \bar{E} \setminus E \end{cases}$$

i.e., projection can be realised simply by adding extra zero components to $\boldsymbol{\theta}$. Note that (20) is an instance of LP (2).

Thus, *the LP relaxation of problem (1) can be improved by adding interaction(s) with zero weight*. From the polyhedral point of view, this corresponds to lifting the marginal polytope to space $\mathbb{R}^{T(\bar{E})}$, imposing the marginalisation constraints given by J, and projecting back to $\mathbb{R}^{T(E)}$.

6.4. Cutting plane algorithm

The *cutting plane algorithm* is a well-known approach to integer LPs. Let P^* be the convex hull of the feasible set of an integer LP (integral hull), typically described by an intractable number of constraints. Let polytope $P \supseteq P^*$ be an outer bound of P^*, described by an LP. If the current optimal solution is $\boldsymbol{\mu} \in P \setminus P^*$, a halfspace is found containing P^* but not $\boldsymbol{\mu}$. The linear inequality representing this halfspace is then added to the LP description of P. Finding a violated inequality is known as the *separation problem*.

Suppose Algorithm 1 has converged and found an optimum of (2). Let us replace E with $E \cup \{A\}$, set interaction θ_A to zero, and replace J by a suitable larger set $J \cup J'$. This has no effect on the current upper bound (7) because it just means adding zero to it. However, since the algorithm can never increase the upper bound, its future iterations will either leave it unchanged or decrease it. If they happen to decrease it, we have succeeded to cut off a part of the polytope $P(E, J)$ that was not in $\operatorname{conv} \boldsymbol{\delta}_E(X_V)$.

Implicitly, this means we have added a set of primal constraints (2b) for $(A, B, x_B) \in J'$. Explicitly, we have added dual variables $\varphi_{A,B}(x_B)$. Note, adding a *single* marginalisation constraint in the space $\mathbb{R}^{T(\bar{E})}$ may mean adding *several* cutting planes in $\mathbb{R}^{T(E)}$, induced in a non-trivial way by the primal constraints and the projection. This is in contrast to the algorithms in [8, 19] which add a single plane at a time and they do it explicitly in the primal space $\mathbb{R}^{T(E)}$.

In fact, we can add a zero interaction even before Algorithm 1 converged. This may or may not lead to a better relaxation in future. If it does not, all we will lose is the memory occupied by the added dual variables $\varphi_{A,B}(x_B)$.

The separation problem corresponds to *finding unsatisfiable CSP subproblems*: if the active joint states of the auxiliary problem $(E', X_A, \boldsymbol{\theta}')$ induced by hyperedge A (see §5) do not form a satisfiable CSP, then adding A results in a strictly better bound. We currently omit the proof of this statement and demonstrate it in §6.5 on an example.

6.5. Example: adding cycles to binary problems

To demonstrate higher-order relaxations and the cutting plane algorithm, we will show for binary problems how the well-known LP relaxation improves by adding cycles[9].

The problem structure, $E = \binom{V}{1} \cup E'$ where $E' \subseteq \binom{V}{2}$, was the 2-dimensional 4-connected $m \times m$ grid. We compared the relaxation (19) for two choices of $(\bar{E}, J)$:

$$\bar{E} = E, \quad J = I(E) \tag{21a}$$

$$\bar{E} = E \cup \{ \text{ all cycles of length four } \}, \quad J = I(\bar{E}) \tag{21b}$$

Recall that (21a) is equivalent to the tree-based relaxation in the sense of [23, 7]. In (21b), $\max_{x_{A \setminus B}} \theta_A^{\boldsymbol{\varphi}}(x_A)$ for the cycle subproblems was computed by dynamic programming.

Figure 1a shows the active joint states (here, individual states and state pairs) after convergence of the algorithm for relaxation (21a) applied on an instance with size $m = 8$, $|X_v| = 4$ labels, and random weights $\boldsymbol{\theta}$. This solution cannot be optimal because of the *unsatisfiable 4-cycle* enclosed by the red curve. Let A denote the four enclosed variables.

[8] Without any constraints on $\bar{E}$ and J, not all integral vertices of $\pi_{T(E)}[P(\bar{E}, J)]$ need be optimal; for that, we would need e.g. (5).

[9] When we say 'adding a cycle', we in fact mean adding a hyperedge A inducing subproblem $(E', X_A, \boldsymbol{\theta}')$ (defined in §5) such that E' is a cycle.

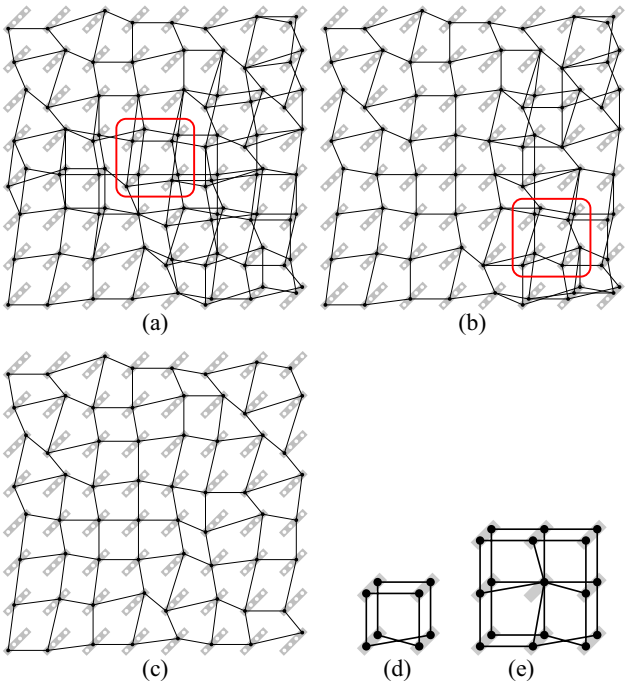

Figure 1. (a,b,c) show the steps of the cutting plane algorithm for for an instance `random` with $m = 8$ and $|X_v| = 4$. (d,e) is the smallest unsatisfiable CSP for relaxation (21a) resp. (21b).

We added hyperedge A to E and extended J to maintain[10] $J = I(E)$. Running the algorithm on this extended problem yielded Figure 1b with an improved bound. Adding the inconsistent 4-cycle shown in Figure 1b resulted in the optimal solution, Figure 1c. In fact, this solution is equivalent to the result of relaxation (21b), in which all the 4-cycles are present all the time, but we needed considerably fewer dual variables because we added only some of these cycles.

On a number of instances, we counted how many instances were solved to optimality (i.e., the relaxation was exact). We tested five types of instances:

- `random`: all weights $\theta_v(x_v)$ and $\theta_{vv'}(x_v, x_{v'})$ were independently drawn from the normal distribution $\mathcal{N}[0; 1]$.
- `Potts`: $\theta_{vv'}(x_v, x_{v'}) = [\![x_v = x_{v'}]\!]$ and $\theta_v(x_v)$ were drawn from $\mathcal{N}[0; 1.6]$. We chose standard deviation 1.6 because it yielded (by trial) the hardest Potts instances.

In the three other types, $\theta_v(x_v)$ were drawn from $\mathcal{N}[0; 1]$ and $\theta_{vv'}(x_v, x_{v'}) \in \{-\infty, 0\}$, i.e., the binary interactions were crisp. They were taken over from [24] as follows:

- `lines`: $\theta_{vv'}(x_v, x_{v'})$ are as in Figure 19a in [24].
- `curve`: $\theta_{vv'}(x_v, x_{v'})$ are as in Figure 15a in [24].
- `Pi`: $\theta_{vv'}(x_v, x_{v'})$ are as in Figure 12d in [24].

[10] Maintaining $J = I(E)$ is in fact a bit wasteful. It requires adding the pencils (A, B, x_B) to J where (B, x_B) are all joint states such that $B \subset A$, i.e., $4 \times 4 + 4 \times 16 = 80$ pencils. But the inconsistent cycle is supported by only 2 (rather than 4) states in each variable, thus we could add less pencils. Note, this would require to relax assumption (4).

type	m	$\lvert X_v \rvert$	r_{tree}	r_{4cycle}
`random`	15	5	0.01	1.00
`random`	25	3	0.00	0.98
`random`	100	3	0.00	0.72
`Potts`	15	5	0.79	0.99
`Potts`	25	5	0.48	0.98
`Potts`	100	5	0.00	0.81
`lines`	10	4	0.72	0.88
`lines`	25	4	0.00	0.00
`curve`	10	9	0.17	0.65
`curve`	15	9	0.00	0.24
`curve`	25	9	0.00	0.00
`Pi`	15	5	0.00	0.82

Table 1. Experimental comparison of relaxations (21a) and (21b). The columns mean: 'type' is the problem type, m is the image side, $|X_v|$ is the number of labels, r_{tree} resp. r_{4cycle} is the proportion of instances solved to optimality by relaxation (21a) resp. (21b). There were 100 instances randomly drawn from each type.

Table 1 shows the results. For `random` and `Potts`, relaxation (21b) was exact far more often than relaxation (21a). For $m = 25$, relaxation (21b) solved `random` in fact always and relaxation (21a) never. For crisp binary interactions, relaxation (21b) also clearly beat (21a). However, for $m \geq 25$ `lines` and `curve` were unsolvable. To conclude, (21b) was surprisingly successful for `random` and `Potts`.

Let us remark that `lines`, `curve` and `Pi` are much harder than instances typical in low-level vision, such as the benchmarks in [20]. In more detail, as observed in [24], they are easy if the data terms $\theta_v(x_v)$ are 'close to a feasible image' but this is not at all the case for random $\theta_v(x_v)$.

An insight why (21b) is notably more successful than (21a) is as follows. Figure 1d resp. 1e shows the smallest unsatisfiable CSP for relaxation (21a) resp. (21b). The subproblem in Figure 1d is more likely to occur than Figure 1e.

The number of 4-cycles added by the cutting plane algorithm ranged between $0-10\%$ of the total number of 4-cycles. Rarely, it was even more.

Currently, the described cutting plane algorithm is inefficient – we have shown only the principle. This is because, e.g., Algorithm 1 is based on hyperedge updates (which is slow) and it is re-run until convergence each time after adding a zero interaction. In fact, the runtime was lower when all the 4-cycles were present all the time: then, the runtime for `random` or `Potts` with $m = 100$ and $|K| = 4$ was several minutes (for a non-optimised Matlab+C code).

Remarks. One can think of adding more complex tractable subproblems than cycles, such as problems with small treewidth. It is even possible to add (reasonably small) subproblems that are not tractable and solve them with a non-polynomial algorithm, like branch&bound.

The results of §6 and §6.4 can be used with other algorithms to minimize the upper bound (7). E.g., the *augmenting DAG algorithm* [9, 25, 24] may turn out more appropriate for the cutting plane scheme than Algorithm 1.

7. Global interactions

A low-arity interaction θ_A is typically represented *extensionally*, i.e., by explicitly storing the weights $\{\, \theta_A(x_A) \mid x_A \in X_A \,\}$. This is impossible for high-arity interactions (because the weights are too many), which have to be represented *intensionally*, i.e., by an algorithm. An intensionally defined interaction can be used in Algorithm 1 whenever computing $\max_{x_{A\setminus B}} \theta^{\varphi}_A(x_A)$ is tractable. Although the last statement is trivial given the results of §5, it is *very significant*: it shows how to incorporate high-arity interactions into the LP relaxation framework in a principled way.

For instance, consider a binary problem plus an interaction on all variables, $E = \binom{V}{1} \cup E' \cup \{V\}$ where $E' \subseteq \binom{V}{2}$. In Algorithm 1, besides the updates done between unary and binary interactions, we need to compute $\max_{x_{V\setminus A}} \theta^{\varphi}_V(x_V)$ for $(V, A, x_A) \in J$. If $J = I(E)$, this would read

$$\max_{x_{V\setminus A}} \Big[\theta_V(x_V) - \sum_{v\in V} \varphi_{V,v}(x_v) - \sum_{vv'\in E} \varphi_{V,vv'}(x_v, x_{v'})\Big]$$

for all $A \in \binom{V}{1} \cup E'$. This is clearly intractable because the last two terms already form a binary problem (1). But let $J = \{\, (A, \{v\}, x_v) \mid A \in E' \cup \{V\},\, v \in V,\, x_v \in X_v \,\}$. This yields a looser relaxation but we need to compute only

$$\max_{x_{V\setminus\{v\}}} \theta^{\varphi}_V(x_V) = \max_{x_{V\setminus\{v\}}} \Big[\theta_V(x_V) - \sum_{u\in V} \varphi_{V,u}(x_u)\Big] \quad (22)$$

for all $v \in V$, which is tractable for a number of interesting functions θ_V which may be useful in applications.

To demonstrate this, we will define several *class size constraints*, constraining the number of variables with a given state in various ways. They are examples of *global interactions*, which in constraint satisfaction are understood as functions of a *non-fixed* set of variables [22, 1] (thus, not necessarily of *all* the variables as in [13]).

E.g., let θ_V be the *crisp equality constraint*, given by

$$\theta_V(x_V) = \begin{cases} 0 & \text{if } \sum_{v\in V} [\![x_v = \bar{x}]\!] = n \\ -\infty & \text{otherwise} \end{cases} \quad (23)$$

which enforces the number of variables with state $\bar{x}$ to be n. To compute (22), a simple greedy algorithm is optimal, the core of which is sorting the numbers $\{\, \varphi_{V,v} - \max_{x\neq\bar{x}} \varphi_{V,v}(\bar{x}) \mid v \in V \,\}$. Using a suitable data structure, this can be done efficiently in an incremental manner.

As an evidence that the approach yields plausible approximation for real instances, we will show an experiment with image segmentation. The first image in Figure 2 is the input binary image corrupted with additive Gaussian noise. Let $X_v = \{\mathsf{BG}, \mathsf{FG}\}$ (background, foreground), $\theta_v = -[f(x_v) - g_v]^2$ where $f(x)$ is the expected intensity of a pixel with label x and g_v is the actual intensity of pixel v in the input image, and $\theta_{vv'}(x_v, x_{v'}) = [\![x_v = x_{v'}]\!]$ (Ising model). The constraint (23) is imposed with $\bar{x} = \mathsf{FG}$.

2000 3000 4000 5000 5368 6000 7000 8000 9000
2008 3004 4011 5006 5368 6004 7024 7982 9032

Figure 2. Applying Ising model with global costraint (23) to image segmentation. The left-hand image: a binary image corrupted with additive Gaussian noise; the uncorrupted image had 5368 black pixels. Other images: the segmentation results; the first and second number under each image is the number of black pixels required (n in (23)) and actually achieved, respectively.

Note, this task is equivalent to the minimum st-cut problem with prescribed partition size, which is NP-hard. The binary images in Figure 2 show the results for different n.

Another example is the soft version of (23) given by $\theta_V(x_V) = -\big|\sum_{v\in V} [\![x_v = \bar{x}]\!] - n\big|$. Or the *region balance constraint* $\theta_{A\cup B}$ that constrains the number of variables with state $\bar{x}$ to be the same in two given regions $A, B \subset V$.

The interaction $\theta_V(x_V) = \big|\sum_{v\in V} [\![x_v = \bar{x}]\!] - n\big|$ encourages the number of variables with state $\bar{x}$ to be far from n. Interestingly, this function is supermodular[11] and if the binary interactions are also supermodular, Algorithm 1 finds the global optimum of (1), as we shall prove in §8.

8. Supermodular n-ary problems

Since maximising a supermodular function on a distributive lattice is tractable [18], problem (1) is tractable if all interactions are supermodular. For binary problems, it is in addition known that the LP relaxation is exact [17, 25].

In §8, we show that the LP relaxation is exact for n-ary supermodular problems. We do this by generalising Schlesinger's proof [17], using our revision of this proof in [25]. More strongly, we show that for n-ary supermodular problems, any fixed point of Algorithm 1 is an optimum of (2) (which may not be the case for non-supermodular problems, see §5) and hence also of (1).

We assume that each domain X_v is endowed with a known total order $\leq_v$. A function θ_A is *supermodular* if

$$\theta_A(x_A \wedge y_A) + \theta_A(x_A \vee y_A) \geq \theta_A(x_A) + \theta_A(y_A)$$

for any $x_A, y_A \in X_A$, where $\wedge$ ($\vee$) denotes the componentwise minimum (maximum) w.r.t. $\leq_v$.

Theorem 2. *Let (5) hold. Let functions θ_A be supermodular for $A \in E$. Let Algorithm 1 converge, such that (9) holds for $(A, B, x_B) \in J$. Let x_V be the joint state defined as follows: x_v is the lowest (w.r.t. the order $\leq_v$) active state of variable v. Then x_V is a solution to problem (1).*

[11] In this light, tractability of (22) for θ_V equal to (23) may be surprising because the function (23) (and also its soft version) is *submodular*.

Proof. After convergence of Algorithm 1, all interactions are supermodular because supermodularity of θ_A is preserved by equivalent transformations. Moreover, the CSP formed by the active joint states $\boldsymbol{\sigma}$ is AC (as said in §5).

It is known [21] that maximisers of a supermodular function on a distributive lattice form a sublattice of this lattice. Therefore, the active joint states σ_A of each interaction θ_A form a lattice, i.e., $\sigma_A(x_A) = \sigma_A(y_A) = 1$ implies that $\sigma_A(x_A \wedge y_A) = \sigma_A(x_A \vee y_A) = 1$ for any $x_A, y_A \in X_A$.

Due to this property, the CSP formed by the active joint states $\boldsymbol{\sigma}$ belongs to a tractable CSP subclass, the *semilattice (min-closed) CSP* [6]. It is known that if a semilattice CSP is AC then it is satisfiable and the solution can be read off simply by taking the lowest state x_v of each variable.

Finally, it is not hard to see that assumption (5) is needed for an AC semilattice CSP to be satisfiable. ■

Optimality of the LP relaxation for n-ary supermodular problems has been recently shown also by Cooper [3]. To compare, our statement is stronger (it suffices that (9) holds for all pencils) and our proof is *much* simpler. Furthermore, note that while binary supermodular problems can be translated to the minimum graph st-cut problem [14], our result does not imply that this can be done for n-ary problems.

References

[1] C. Bessiere. Constraint propagation. In *Handbook of Constraint Programming*, chapter 3. Elsevier, 2006. Also technical report LIRMM 06020.

[2] M. Cooper, S. de Givry, and T. Schiex. Optimal soft arc consistency. In *Proc. IJCAI, Hyderabad, India*, 2007.

[3] M. C. Cooper. Minimization of locally-defined submodular functions by optimal soft arc consistency. *Constraints*, 13(4), 2008.

[4] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Number 15 in Algorithms and Combinatorics. Springer, Berlin, Germany, 1997.

[5] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Math. Programming*, 28:121–155, 1984.

[6] P. Jeavons and M. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):327–339, 1995.

[7] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[8] A. Koster, C. P. M. van Hoesel, and A. W. J. Kolen. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 23(3–5):89–97, 1998.

[9] V. K. Koval and M. I. Schlesinger. Two-dimensional programming in image analysis problems. *USSR Academy of Science, Automatics and Telemechanics*, 8:149–168, 1976. In Russian.

[10] V. A. Kovalevsky and V. K. Koval. A diffusion algorithm for decreasing energy of max-sum labeling problem. Glushkov Institute of Cybernetics, Kiev, USSR. Unpublished, approx. 1975.

[11] A. Mackworth. Constraint satisfaction. In *Encyclopaedia of Artificial Intelligence*, pages 285–292. Wiley, 1991.

[12] P. Meseguer, F. Rossi, and T. Schiex. Soft constraints. In *Handbook of Constraint Programming*, chapter 9. Elsevier, 2006.

[13] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching – incorporating a global constraint into MRFs. In *Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[14] D. Schlesinger and B. Flach. Transforming an arbitrary MinSum problem into a binary one. Technical Report TUD-FI06-01, Dresden University of Technology, Germany, April 2006.

[15] M. I. Schlesinger. False minima of the algorithm for minimizing energy of max-sum labeling problem. Glushkov Institute of Cybernetics, Kiev, USSR. Unpublished, 1976.

[16] M. I. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions. *Kibernetika*, 4:113–130, 1976. In Russian.

[17] M. I. Schlesinger and B. Flach. Some solvable subclasses of structural recognition problems. In *Czech Pattern Recognition Workshop*, 2000.

[18] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Combinatorial Theory, Ser. B*, 80(2):346–355, 2000.

[19] D. Sontag and T. Jaakkola. New outer bounds on the marginal polytope. In *Proc. Neural Information Processing Systems (NIPS)*. MIT Press, 2007.

[20] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *European Conf. Computer Vision (ECCV)*, pages II: 16–29, 2006.

[21] D. M. Topkis. *Supermodularity and Complementarity*. Frontiers of Economic Research. Princeton University Press, Princeton, NJ, 1998.

[22] W.-J. van Hoeve and I. Katriel. Global constraints. In *Handbook of Constraint Programming*, chapter 7. Elsevier, 2006.

[23] M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on (hyper)trees: message passing and linear programming approaches. *IEEE Trans. Information Theory*, 51(11):3697–3717, 2005.

[24] T. Werner. A linear programming approach to max-sum problem: A review. Technical Report CTU–CMP–2005–25, Center for Machine Perception, Czech Technical University, December 2005.

[25] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(7):1165–1179, July 2007.

[26] T. Werner and A. Shekhovtsov. Unified framework for semiring-based arc consistency and relaxation labeling. In M. Grabner and H. Grabner, editors, *12th Computer Vision Winter Workshop, St. Lambrecht, Austria*, pages 27–34. Graz University of Technology, February 2007.