A Linear Programming Approach to Max-Sum Problem: A Review

Tomáš Werner, *Member*, *IEEE Computer Society*

Abstract—The max-sum labeling problem, defined as maximizing a sum of binary (i.e., pairwise) functions of discrete variables, is a general NP-hard optimization problem with many applications, such as computing the MAP configuration of a Markov random field. We review a not widely known approach to the problem, developed by Ukrainian researchers Schlesinger et al. in 1976, and show how it contributes to recent results, most importantly, those on the convex combination of trees and tree-reweighted max-product. In particular, we review Schlesinger et al.'s upper bound on the max-sum criterion, its minimization by equivalent transformations, its relation to the constraint satisfaction problem, the fact that this minimization is dual to a linear programming relaxation of the original problem, and the three kinds of consistency necessary for optimality of the upper bound. We revisit problems with Boolean variables and supermodular problems. We describe two algorithms for decreasing the upper bound. We present an example application for structural image analysis.

Index Terms—Markov random fields, undirected graphical models, constraint satisfaction, belief propagation, linear programming relaxation, max-sum, max-plus, max-product, supermodular optimization.

1 Introduction

The binary (i.e., pairwise) max-sum labeling problem is defined as maximizing a sum of unary and binary functions of discrete variables, i.e., as computing

$$\max_{\mathbf{x}\in X^T}\left[\sum_{t\in T} g_t(x_t) + \sum_{\{t,t'\}\in E} g_{tt'}(x_t, x_{t'})\right],$$

where an undirected graph (T, E), a finite set X, and numbers $g_t(x_t)$, $g_{tt'}(x_t, x_{t'}) \in \mathbb{R}\cup\{-\infty\}$ are given. It is a very general NP-hard optimization problem which has been studied and applied in several disciplines, such as statistical physics, combinatorial optimization, artificial intelligence, pattern recognition, and computer vision. In the latter two, the problem is also known as the computing maximum posterior (MAP) configuration of Markov random fields (MRF).

This paper reviews an old and not widely known approach to the max-sum problem by Ukrainian scientists Schlesinger et al. and shows how it contributes to recent knowledge.

1.1 Approach by Schlesinger et al.

The basic elements of the old approach were given by Schlesinger in 1976 in structural pattern recognition. In [1], he generalizes locally conjunctive predicates by Minsky and Papert [2] to *two-dimensional (2D) grammars* and shows that these are useful for structural image analysis. Two tasks are considered on 2D grammars. The first task assumes analysis of ideal, noise-free images: test whether an input image belongs to the language generated by a given grammar. It leads to what is today known as the *Constraint Satisfaction Problem* (CSP) [3] or *discrete relaxation labeling*. Finding the largest *arc-consistent* subproblem provides some necessary but not sufficient conditions for satisfiability and unsatisfiability of the problem. The second task considers analysis of noisy images: Find an image belonging to the language generated by a given 2D grammar that is "nearest" to a given image. It leads to the max-sum problem.

In detail, paper [1] formulates a linear programming relaxation of the max-sum problem and its dual program. The dual is interpreted as minimizing an upper bound to the max-sum problem by *equivalent transformations*, which are redefinitions of the problem that leave the objective function unchanged. The optimality of the upper bound is equal to the *triviality* of the problem. Testing for triviality leads to a CSP.

An algorithm to decrease the upper bound, which we called the *augmenting DAG algorithm*, was suggested in [1] and presented in more detail by Koval and Schlesinger in [4] and further in [5]. Another algorithm to decrease the upper bound is a coordinate descent method, *max-sum diffusion*, discovered by Kovalevsky and Koval [6] and later independently by Flach [7]. Schlesinger noticed [8] that the termination criterion of both algorithms, arc consistency, is necessary but not sufficient for minimality of the upper bound. Thus, the algorithms sometimes find the true minimum of the upper bound and sometimes only decrease it to some point.

The material in [1], [4] is presented in detail in the book [9]. The name "2D grammars" was later assigned a different meaning in the book [10] by Schlesinger and Hlaváč. In their original meaning, they largely coincide with MRFs.

By minimizing the upper bound, some max-sum problems can be solved to optimality (the upper bound is

---

• *The author is with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, Karlovo namesti 13, 12135 Praha, Czech Republic. E-mail: werner@cmp.felk.cvut.cz.*

Manuscript received 19 Feb. 2006; revised 28 Aug. 2006; accepted 1 Nov. 2006; published online 18 Jan. 2007.
Recommended for acceptance by J. Goutsias.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0170-0206.
Digital Object Identifier no. 10.1109/TPAMI.2007.1036.

tight) and some cannot (there is an integrality gap). Schlesinger and Flach [11] proved that supermodular problems have zero integrality gap.

1.2 Relation to Recent Works

Independently of the work by Schlesinger et al., significant progress has recently been achieved in the max-sum problem. This section reviews the most relevant newer results by others and shows how they relate to the old approach.

1.2.1 Convex Relaxations and Upper Bounds

It is common in combinatorial optimization to approach NP-hard problems via continuous relaxations of their integer programming formulations. The linear programming relaxation given by Schlesinger [1] is quite natural and has been suggested independently and later by others: by Koster et al. [12], [13], who address the max-sum problem as a generalization of CSP, the *Partial CSP*; by Chekuri et al. [14] and Wainwright et al. [15]; and in bioinformatics [16]. Koster et al. in addition give two classes of nontrivial facets of the Partial CSP polytope, i.e., linear constraints missing in the relaxation.

Max-sum problems with Boolean (i.e., two-state) variables are a subclass of pseudo-Boolean and quadratic Boolean optimization, see, e.g., the review [17]. Here, several different upper bounds were suggested, which were shown equivalent by Hammer et al. [18]. These bounds are, in turn, equivalent to [1], [12], [14] with Boolean variables, as shown in [19].

Relaxations of the max-sum problem other than linear programming have been suggested, such as quadratic [20], [21], and semidefinite [22] programming relaxations. We will not discuss these.

1.2.2 Convex Combination of Trees

The max-sum problem has been studied in the terminology of graphical models; in particular, it is equivalent to finding MAP configurations of undirected graphical models, also known as MRFs. This research primarily focused on computing the partition function and marginals of MRFs and approached the max-sum problem as the limit case of this task.

Wainwright et al. [24] shows that a convex combination of problems provides a convex upper bound on the log-partition function of MRF. These subproblems can be conveniently chosen as (tractable) tree problems. For the sum-product problem on cyclic graphs, this upper bound is almost never tight. In the max-sum limit (also known as the *zero temperature limit*), the bound is tight much more often, namely, if the optima on individual trees share a common configuration, which is referred to as *tree agreement* [15], [25]. Moreover, in the max-sum case, the bound is independent of the choice of trees. Minimizing the upper bound is shown [15], [25] to be a Lagrangian dual of a linear programming relaxation of the max-sum problem. This relaxation is the same as in [1], [12], [14]. Besides directly solving this relaxation, the *tree-reweighted message passing* (TRW) algorithm is suggested to minimize the upper bound. Importantly, it is noted [26], [27] that message passing can be alternatively viewed as *reparameterizations* of the problem. TRW is guaranteed to neither converge nor decrease the upper bound monotonically. Kolmogorov [28], [29], [30], [31] suggests its sequential modification (TRW-S) and conjectures that it always converges to a state characterized by *weak tree agreement*. He further shows that the point of convergence might differ from a global minimum of the upper bound; however, for Boolean variables [19], [32], they are equal.

The approach based on convex combination of trees is closest to the approach by Schlesinger et al. The linear programming relaxation considered by Wainwright is the same as Schlesinger's one. Reparameterizations correspond to Schlesinger's equivalent transformations. If the trees are chosen as individual nodes and edges, Wainwright's upper bound becomes Schlesinger's upper bound, tree agreement becomes CSP satisfiability, and weak tree agreement becomes arc consistency. The convenient choice of subproblems as nodes and edges is without loss of generality because Wainwright's bound is independent of the choice of trees.

The approach based on convex combination of trees is more general than the approach reviewed in this paper, but the latter is simpler; hence, it may be more suitable for analysis. However, the translation between the two is not straightforward and the approach by Schlesinger et al. provides the following contributions to that by Wainwright et al. and Kolmogorov.

Duality of linear programming relaxation of the max-sum problem and minimizing Schlesinger's upper bound is proven more straightforwardly by putting both problems into matrix form [1], as is common in linear programming.

The max-sum problem is intimately related to CSP via complementary slackness. This reveals that testing for tightness of the upper bound is NP-complete, which has not been noticed by others. It leads to a relaxation of CSP, which provides a simple way [8] to characterize spurious minima of the upper bound. This has an independent value for CSP research.

The max-sum diffusion is related to TRW-S but has an advantage in its simplicity, which also might help further analysis. With its combinatorial flavor, the Koval-Schlesinger augmenting DAG algorithm [4] is dissimilar to any recent algorithm and somewhat resembles the augmenting path algorithm for the max-flow/min-cut problem.

1.2.3 Loopy Belief Propagation

It has long been known that the sum-product and max-sum problems on trees can be efficiently solved by *belief propagation* and *message passing* [33]. When applied to cyclic graphs, these algorithms were empirically found to sometimes converge and sometimes not, with the fixed points (if any) sometimes being useful approximations. The main recent result [34] is that the fixed points of this "loopy" belief propagation are local minima of a nonconvex function, known in statistical physics as Bethe free energy.

The max-sum diffusion resembles loopy belief propagation: Both repeat simple local operations and both can be interpreted as a coordinate descent minimization of some functional. However, for the diffusion, this functional is convex, while, for belief propagation, it is nonconvex.

1.2.4 CSP and Extensions

The CSP seeks to find values of discrete variables that satisfy given logical constraints. Extensions to it have been suggested in which the constraints become soft and one seeks to maximize a criterion rather than satisfy constraints. The max-sum problem is often closely related to these extensions. Examples are the Max CSP [35] (subclass of the max-sum problem), Valued CSP [36] (more general than max-sum), and Partial CSP [12] (equivalent to max-sum).

The max-sum problem also relates to CSP via complementary slackness, as mentioned above. This establishes links to the large CSP literature, which may be fruitful in both directions. This paper seems to be the first in pattern recognition and computer vision to make this link.

1.2.5 Maximum Flow (Minimum Cut)

Finding max-flow/min-cut in a graph has been recognized as being very useful for (mainly low-level) computer vision [37]. Later, it was realized that supermodular max-sum problems can be translated to max-flow/min-cut (see Section 8). For supermodular max-sum problems, Schlesinger and Flach's upper bound is tight and finding an optimal configuration is tractable [11]. The relation of this result with lattice theory is considered in [19], [38], [39], [40], [41]. We further extend this relation and give it a simpler form.

1.3 Organization of the Paper

Section 2 introduces the labeling problem on commutative semirings and basic concepts. Section 3 reviews CSP. Section 4 presents the linear programming relaxation of the max-sum problem, its dual, Schlesinger's upper bound, and equivalent and trivial problems. Section 5 characterizes minimality of the upper bound. Two algorithms for decreasing the upper bound are described in Sections 6 and 7. Section 8 establishes that the bound is tight for supermodular problems. Application to structural image analysis [1], [9] is presented in Section 9. A previous version of this paper is [42].

Logical conjunction (disjunction) is denoted by $\wedge$ ($\vee$). Function $[\![\psi]\!]$ returns 1 if logical expression ψ is true and 0 if it is false. The set of *all* maximizers of $f(x)$ is $\operatorname{argmax}_x f(x)$. Assignment is denoted by $x := y$, symbol $x \mathrel{+}= y$ denotes $x := x + y$. New concepts are in **boldface**.

2 Labeling Problems on Commutative Semirings

This section defines a class of labeling problems of which the CSP and the max-sum problem are special cases. Here, we introduce basic terminology used in the rest of the paper.

We will use the terminology from [11], where the variables are called objects and their values are called labels. Let $G = (T, E)$ be an undirected graph, where T is a discrete set of **objects** and $E \subseteq \binom{T}{2}$ is a set of (object) **pairs**. The set of neighbors of an object t is $N_t = \{t' \mid \{t, t'\} \in E\}$. Each object $t \in T$ is assigned a **label** $x_t \in X$, where X is a discrete set. A **labeling** $\mathbf{x} \in X^T$ is an $|T|$-tuple that assigns a single label x_t to each object t. When not viewed as components of $\mathbf{x}$, elements of X will be denoted by x, x' without subscripts.

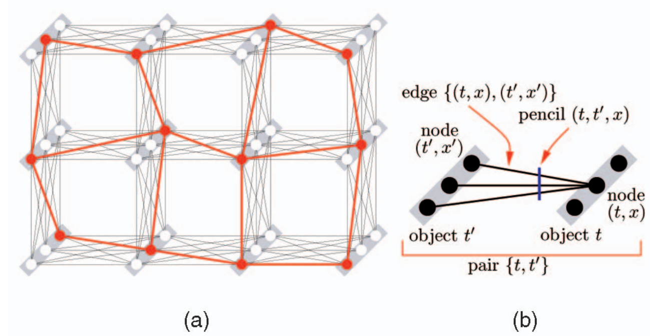


Fig. 1. (a) The 3×4 grid graph G (i.e., $|T| = 12$), graph $(T \times X, E_X)$ for $|X| = 3$ labels, and a labeling $\mathbf{x}$ (emphasized). (b) Parts of both graphs.

Let $(T \times X, E_X)$ be another undirected graph with edges $E_X = \{\{(t, x), (t', x')\} \mid \{t, t'\} \in E, x, x' \in X\}$. When G is a chain, this graph corresponds to the *trellis diagram*, frequently used to visualize Markov chains. The nodes and edges of G will be called objects and pairs, respectively, whereas the terms **nodes** and **edges** will refer to $(T \times X, E_X)$. The set of all nodes and edges is $I = (T \times X) \cup E_X$. The set of edges leading from a node (t, x) to all nodes of a neighboring object $t' \in N_t$ is a **pencil** (t, t', x). The set of all pencils is $P = \{(t, t', x) \mid \{t, t'\} \in E, x \in X\}$. Fig. 1 shows how both graphs, their parts, and labelings will be visualized.

Let an element $g_t(x)$ of a set S be assigned to each node (t, x) and an element $g_{tt'}(x, x')$ to each edge $\{(t, x), (t', x')\}$, where $g_{tt'}(x, x') = g_{t't}(x', x)$. The vector obtained by concatenating all $g_t(x)$ and $g_{tt'}(x, x')$ is denoted by $\mathbf{g} \in S^I$.

Before starting with the max-sum labeling problem, we introduce labeling problems in a more general form. It was observed [11], [43], [44], [45], [46] that different labeling problems can be unified by letting a suitable commutative semiring specify how different constraints are combined together. Let S endowed with two binary operations $\oplus$ and $\otimes$ form a commutative semiring (S, $\oplus$, $\otimes$). The semiring formulation of the labeling problem [11] is defined as computing

$$\bigoplus_{\mathbf{x} \in X^T} \left[\bigotimes_{t \in T} g_t(x_t) \otimes \bigotimes_{\{t,t'\} \in E} g_{tt'}(x_t, x'_t) \right]. \tag{1}$$

More exactly, this is the *binary* labeling problem, according to the highest arity of the functions in the brackets. We will not consider problems of higher arity.

Interesting problems are obtained, modulo isomorphisms, by the following choices of the semiring:

$(S, \oplus, \otimes)$	task
$(\{0,1\}, \vee, \wedge)$	or-and problem, CSP
$([-\infty, \infty), \min, \max)$	min-max problem
$([-\infty, \infty), \max, +)$	max-sum problem
$([0, \infty), +, *)$	sum-product problem

Note that the *extended domain*, $S = [-\infty, \infty)$, of min-max and max-sum problems yields a more general formulation than is usually used, $S = (-\infty, \infty)$.

The topic of this paper is the max-sum problem, but we will also briefly cover the closely related CSP. Since semiring $(\{0, 1\}, \vee, \wedge)$ is isomorphic with $(\{-\infty, 0\}, \max, +)$, CSP is a

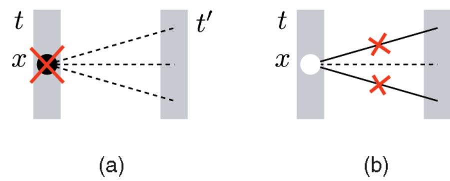


Fig. 2. The arc consistency algorithm deletes (a) nodes not linked with some neighbor by any edge and (b) edges lacking an end node.

subclass of the max-sum problem. However, we will treat CSP separately since a lot of independent research has been done on it. We will not discuss the sum-product problem (i.e., computing MRF partition function) and the min-max problem.

3 Constraint Satisfaction Problem

The **constraint satisfaction problem** (CSP) [3] is defined as finding a labeling that satisfies given unary and binary constraints, i.e., that passes through some or all of given nodes and edges. It was introduced, often independently, several times in computer vision [1], [47], [48], [49] and artificial intelligence [50], often under different names, such as the *Consistent Labeling Problem* [51]. CSP is NP-complete. Tractable subclasses are obtained either by restricting the structure of G (such as limiting its tree width) or the constraint language. In the latter, a lot of research has been done and mathematicians seem to be close to complete classification [52]. Independently of this, Schlesinger and Flach discovered a tractable CSP subclass defined by the *interval condition* [11], [39]. In particular, binary CSP with Boolean variables is known to be tractable.

We denote a CSP instance by $(G, X, \bar{\mathbf{g}})$. Indicators $\bar{g}_t(x)$, $\bar{g}_{tt'}(x, x') \in \{0, 1\}$ state whether the corresponding node or edge is allowed or forbidden. The task is to compute the set

$$\bar{L}_{G,X}(\bar{\mathbf{g}}) = \left\{ \mathbf{x} \in X^T \;\middle|\; \bigwedge_t \bar{g}_t(x_t) \wedge \bigwedge_{\{t,t'\}} \bar{g}_{tt'}(x_t, x_{t'}) = 1 \right\}. \quad (2)$$

A CSP is **satisfiable** if $\bar{L}_{G,X}(\bar{\mathbf{g}}) \neq \emptyset$.

Some conditions necessary or sufficient (but not both) for satisfiability can be given in terms of local consistencies, surveyed, e.g., in [53]. The simplest local consistency is arc consistency. A CSP is **arc-consistent** if

$$\bigvee_{x'} \bar{g}_{tt'}(x, x') = \bar{g}_t(x), \quad \{t, t'\} \in E, \; x \in X. \quad (3)$$

CSP $(G, X, \bar{\mathbf{g}}')$ is a **subproblem** of $(G, X, \bar{\mathbf{g}})$ if $\bar{\mathbf{g}}' \leq \bar{\mathbf{g}}$. The **union** of CSPs $(G, X, \bar{\mathbf{g}})$ and $(G, X, \bar{\mathbf{g}}')$ is $(G, X, \bar{\mathbf{g}} \vee \bar{\mathbf{g}}')$. Here, operations $\leq$ and $\vee$ are meant componentwise. Following [1], [9], we define the **kernel** of a CSP as follows: First, note that the union of arc-consistent CSPs is arc-consistent. To see this, write the disjunction of (3) for arc-consistent $\bar{\mathbf{g}}$ and $\bar{\mathbf{g}}'$ as

$$\left[\bigvee_{x'} \bar{g}_{tt'}(x, x')\right] \vee \left[\bigvee_{x'} \bar{g}'_{tt'}(x, x')\right] = \bigvee_{x'} [\bar{g}_{tt'}(x, x') \vee \bar{g}'_{tt'}(x, x')]$$
$$= \bar{g}_t(x) \vee \bar{g}'_t(x),$$

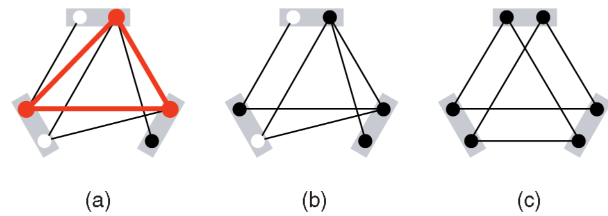


Fig. 3. Examples of CSPs: (a) satisfiable, hence with a nonempty kernel which allows the formation of a labeling (the labeling is emphasized); (b) with an empty kernel, hence unsatisfiable; (c) arc-consistent but unsatisfiable. The forbidden nodes are in white, the forbidden edges are not shown.

obtaining that $\bar{\mathbf{g}} \vee \bar{\mathbf{g}}'$ satisfies (3). The kernel of a CSP is the union of all of its arc-consistent subproblems. Arc consistent subproblems of a problem form a join semilattice with respect to the partial ordering by inclusion $\leq$. The greatest element of this semilattice is the kernel. Equivalently, the kernel is the largest arc-consistent subproblem.

The kernel can be found by the **arc consistency algorithm**, also known as *discrete relaxation labeling* [49]. Starting with their initial values, the variables $\bar{g}_t(x)$ and $\bar{g}_{tt'}(x, x')$ violating (3) are iteratively set to zero by applying the following rules (Fig. 2):

$$\bar{g}_t(x) := \bar{g}_t(x) \wedge \bigvee_{x'} \bar{g}_{tt'}(x, x'), \quad (4a)$$

$$\bar{g}_{tt'}(x, x') := \bar{g}_{tt'}(x, x') \wedge \bar{g}_t(x) \wedge \bar{g}_{t'}(x'). \quad (4b)$$

The algorithm halts when no further variable can be set to zero. It is well-known that the result does not depend on the order of the operations.

Theorem 1. *Let $(G, X, \bar{\mathbf{g}}^*)$ be the kernel of a CSP $(G, X, \bar{\mathbf{g}})$. It holds that $\bar{L}_{G,X}(\bar{\mathbf{g}}) = \bar{L}_{G,X}(\bar{\mathbf{g}}^*)$.*

Proof. The theorem is a corollary of the more general Theorem 6, given later.

It can also be proved by the following induction argument: If a pencil (t, t', x) contains no edge, the node (t, x) clearly cannot belong to any labeling (Fig. 2a). Therefore, the node (t, x) can be deleted without changing $\bar{L}_{G,X}(\bar{\mathbf{g}})$. Similarly, if a node (t, x) is forbidden, then no labeling can pass through any of the pencils $\{(t, t', x) \mid t' \in N_t\}$ (Fig. 2b). □

A corollary of Theorem 1 is given by the following conditions proving or disproving satisfiability. Fig. 3 shows examples.

Theorem 2. *Let $(G, X, \bar{\mathbf{g}}^*)$ denote the kernel of CSP $(G, X, \bar{\mathbf{g}})$.*

- *If the kernel is empty ($\bar{\mathbf{g}}^* = \mathbf{0}$), then the CSP is not satisfiable.*
- *If there is a unique label in each object ($\sum_x \bar{g}^*_t(x) = 1$ for $t \in T$), then the CSP is satisfiable.*

4 The Max-Sum Problem

We now turn our attention to the central topic of the paper, **the max-sum problem**. Its instance is denoted by $(G, X, \mathbf{g})$, where $g_t(x)$ and $g_{tt'}(x, x')$ will be called **qualities**. The **quality of a labeling** $\mathbf{x}$ is

$$F(\mathbf{x} \mid \mathbf{g}) = \sum_{t \in T} g_t(x_t) + \sum_{\{t,t'\} \in E} g_{tt'}(x_t, x_{t'}). \tag{5}$$

Solving the problem means finding (one, several, or all elements of) the set of optimal labelings

$$L_{G,X}(\mathbf{g}) = \operatorname*{argmax}_{\mathbf{x} \in X^T} F(\mathbf{x} \mid \mathbf{g}). \tag{6}$$

4.1 Linear Programming Relaxation

Let us formulate a linear programming relaxation of the max-sum problem (6). For that, we introduce a different representation of labelings that allows us to represent "partially decided" labelings. A **relaxed labeling** is a vector $\boldsymbol{\alpha}$ with components $\alpha_t(x)$ and $\alpha_{tt'}(x, x')$ satisfying

$$\sum_{x'} \alpha_{tt'}(x, x') = \alpha_t(x), \;\; \{t, t'\} \in E, \; x \in X, \tag{7a}$$

$$\sum_{x} \alpha_t(x) = 1, \qquad t \in T, \tag{7b}$$

$$\boldsymbol{\alpha} \geq \mathbf{0}, \tag{7c}$$

where $\alpha_{tt'}(x, x') = \alpha_{t't}(x', x)$. Number $\alpha_t(x)$ is assigned to node (t, x), number $\alpha_{tt'}(x, x')$ to edge $\{(t, x), (t', x')\}$. The set of all $\boldsymbol{\alpha}$ satisfying (7) is a polytope, denoted by $\Lambda_{G,X}$. A binary vector $\boldsymbol{\alpha}$ represents a "decided" labeling; there is a bijection between the sets X^T and $\Lambda_{G,X} \cap \{0,1\}^I$, given by $\alpha_t(x) = [\![x_t = x]\!]$ and $\alpha_{tt'}(x, x') = \alpha_t(x)\alpha_{t'}(x')$. A noninteger $\boldsymbol{\alpha}$ represents an "undecided" labeling.

Remark 1. Constraints (7a)+(7b) are linearly dependent. To see this, denote $\alpha_t = \sum_x \alpha_x(t)$ and $\alpha_{tt'} = \sum_{x,x'} \alpha_{tt'}(x, x')$ and sum (7a) over x, which gives $\alpha_t = \alpha_{tt'}$. Since G is connected, (7a) alone implies that α_t and $\alpha_{tt'}$ are equal for the whole G. Thus, $\Lambda_{G,X}$ could be represented in a less redundant way, e.g., by replacing (7b) with $\sum_t \alpha_t = |T|$. It is shown in [12] that $\dim \Lambda_{G,X} = |T|(|X| - 1) + |E|(|X| - 1)^2$.

Remark 2. Conditions (7a)+(7c) can be viewed as a continuous generalization of arc consistency (3) in the following sense: For any $\boldsymbol{\alpha}$ satisfying (7a)+(7c), the CSP $\bar{\mathbf{g}}$ given by $\bar{g}_t(x) = [\![\alpha_t(x) > 0]\!]$ and $\bar{g}_{tt'}(x, x') = [\![\alpha_{tt'}(x, x') > 0]\!]$ satisfies (3).

The quality and equivalence of max-sum problems can be extended from ordinary to relaxed labelings. The quality of a relaxed labeling $\boldsymbol{\alpha}$ is the scalar product $\langle \mathbf{g}, \boldsymbol{\alpha} \rangle$. Like $F(\bullet|\mathbf{g})$, function $\langle \mathbf{g}, \bullet \rangle$ is invariant to equivalent transformations because $\langle \mathbf{0}^\varphi, \boldsymbol{\alpha} \rangle$ identically vanishes, as is verified by substituting (9) and (7a). The **relaxed max-sum problem** is the linear program

$$\Lambda_{G,X}(\mathbf{g}) = \operatorname*{argmax}_{\alpha \in \Lambda_{G,X}} \langle \mathbf{g}, \boldsymbol{\alpha} \rangle.$$

The set $\Lambda_{G,X}(\mathbf{g})$ is a polytope, as it is the convex hull of the optimal vertices of $\Lambda_{G,X}$. If $\Lambda_{G,X}(\mathbf{g})$ has integer elements, they coincide with $L_{G,X}(\mathbf{g})$.

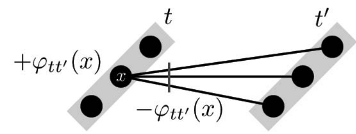


Fig. 4. The elementary equivalent transformation.

The linear programming relaxation (8) was suggested by several researchers independently: by Schlesinger in structural pattern recognition [1], by Koster et al. as an extension of CSP [12], by Chekuri et al. [14] for metric Markov random fields, and in bioinformatics [16].

Solving (8) by a general linear programming algorithm, such as the simplex or interior point method, would be inefficient and virtually impossible for large instances which occur, e.g., in computer vision. There are two ways to do better. First, the linear programming dual of (8) is more suitable for optimization because it has fewer variables. Second, a special algorithm utilizing the structure of the task has to be designed.

Further on in Section 4, we formulate the dual of (11) and interpret it as minimizing an *upper bound* on problem quality by *equivalent transformations* and that the tightness of the relaxation is equivalent to the satisfiability of a CSP. The subsequent Section 5 gives conditions for minimality of the upper bound, implied by complementary slackness.

4.2 Equivalent Max-Sum Problems

Problems $(G, X, \mathbf{g})$ and $(G, X, \mathbf{g}')$ are called **equivalent** (denoted by $\mathbf{g} \sim \mathbf{g}'$) if functions $F(\bullet \mid \mathbf{g})$ and $F(\bullet \mid \mathbf{g}')$ are identical [1], [26], [29]. An **equivalent transformation** is a change of $\mathbf{g}$ taking a max-sum problem to its equivalent. Fig. 4 shows the simplest such transformation: Choose a pencil (t, t', x), add a number $\varphi_{tt'}(x)$ to $g_t(x)$, and subtract the same number from all edges in pencil (t, t', x).

A special equivalence class is formed by **zero problems** for which $F(\bullet|\mathbf{g})$ is the zero function. By (5), the zero class $\{\mathbf{g} \mid \mathbf{g} \sim \mathbf{0}\}$ is a linear subspace of $\mathbb{R}^I$. Problems $\mathbf{g}$ and $\mathbf{g}'$ are equivalent if and only if $\mathbf{g} - \mathbf{g}'$ is a zero problem.

We will parameterize any equivalence class by a vector $\varphi \in \mathbb{R}^P$ with components $\varphi_{tt'}(x)$ assigned to pencils (t, t', x). Variables $\varphi_{tt'}(x)$ are called *potentials* in [1], [4], [9] and correspond to *messages* in the belief propagation literature. The equivalent of a problem $\mathbf{g}$ given by φ is denoted by $\mathbf{g}^\varphi = \mathbf{g} + \mathbf{0}^\varphi$. It is obtained by composing the elementary transformations shown in Fig. 4 for all pencils, which yields

$$g_t^\varphi(x) = g_t(x) + \sum_{t' \in N_t} \varphi_{tt'}(x), \tag{9a}$$

$$g_{tt'}^\varphi(x, x') = g_{tt'}(x, x') - \varphi_{tt'}(x) - \varphi_{t't}(x'). \tag{9b}$$

It is easy to see that problems $\mathbf{g}$ and $\mathbf{g}^\varphi$ are equivalent for any φ since inserting (9) to (5) shows that $F(\mathbf{x} \mid \mathbf{g}^\varphi)$ identically equals $F(\mathbf{x} \mid \mathbf{g})$. We would also like the converse to hold, i.e., any two equivalent problems to be related by (9) for some φ. However, this holds only if G is connected and all qualities $\mathbf{g}$ are finite, as is given by Theorem 3. Connectedness of G is naturally satisfied in applications. The second assumption does not seem to be an obstacle in

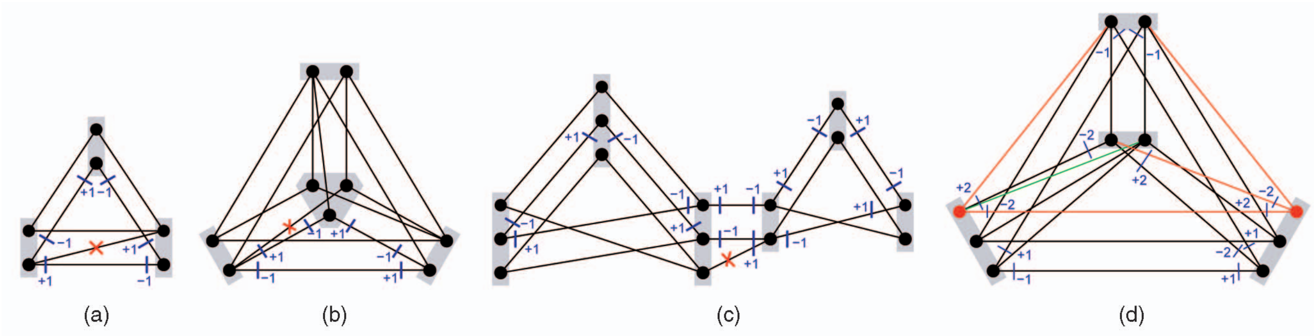


Fig. 5. Examples of kernels not invariant to equivalent transformations. The shown edges have quality 0 and the not shown edges $-\infty$. Problem (a) has minimal height, problems (b), (c) do not; in particular, for (b), (c) system (7a)+(7b)+(17) is unsolvable. For problem (d), system (7a)+(7b)+(17) is solvable but system (7a)+(7b)+(7c)+(17) is not.

algorithms even when the extended domain $\mathbf{g} \in [-\infty, \infty)^I$ is used, though we still do not fully understand why.

Theorem 3 [54], [1], [29], [31]. *Let the graph G be connected and $\mathbf{g} \in \mathbb{R}^I$. $F(\bullet|\mathbf{g})$ is the zero function if and only if there exist numbers $\varphi_{tt'}(x) \in \mathbb{R}$ such that*

$$g_t(x) = \sum_{t' \in N_t} \varphi_{tt'}(x), \tag{10a}$$

$$g_{tt'}(x, x') = -\varphi_{tt'}(x) - \varphi_{t't}(x'). \tag{10b}$$

The reader may skip the proof in the first reading.

Proof. The *if* part is easy, by verifying that (5) identically vanishes after substituting (10). We will prove the *only if* part.

Since $F(\bullet|\mathbf{g})$ is the zero function and, therefore, it is modular (i.e., both sub and supermodular with respect to to any order $\leq$), by Theorem 12, given later, functions $g_{tt'}(\bullet,\bullet)$ are also modular. Any modular function is a sum of univariate functions [55]. This implies (10b).

Let $\mathbf{x}$ and $\mathbf{y}$ be two labelings that differ only in an object t, where they satisfy $x_t = x$ and $y_t = y$. After substituting (5) and (10b) to the equality $F(\mathbf{x}|\mathbf{g}) = F(\mathbf{y}|\mathbf{g})$, most terms cancel out, giving $g_t(x) - \sum_{t'} \varphi_{tt'}(x) = g_t(y) - \sum_{t'} \varphi_{tt'}(y)$. Since this holds for any x and y, neither side depends on x. Thus, we can denote $\varphi_t = g_t(x) - \sum_{t'} \varphi_{tt'}(x)$. Substituting (10) into $F(\bullet|\mathbf{g}) = 0$ yields $\sum_t \varphi_t = 0$.

To show (10a), we will give an equivalent transformation that sets all φ_t to zero. Let G' be a spanning tree of G. It exists because G is connected. Find a pair $\{t, t'\}$ in G' such that t is a leaf. Do the following transformation of $(G, X, \mathbf{g})$: Set $\varphi_{tt'}(x) \mathrel{+}= \varphi_t$ for all x and $\varphi_{t't}(x') \mathrel{-}= \varphi_t$ for all x'. Set $\varphi_{t'} \mathrel{+}= \varphi_t$ and $\varphi_t := 0$. Remove t and $\{t, t'\}$ from G'. Repeat until G' is empty. □

As a counterexample for infinite $\mathbf{g}$, consider the problem in Fig. 5a and the same problem with the crossed edge being $-\infty$. These two problems are equivalent, but they are not related by (9) for any $\varphi \in \mathbb{R}^P$.

4.3 Schlesinger's Upper Bound and Its Minimization

Let the **height of object** t and the **height of pair** $\{t, t'\}$ be, respectively,

$$u_t = \max_x g_t(x), \quad u_{tt'} = \max_{x,x'} g_{tt'}(x, x'). \tag{12}$$

The **height of a max-sum problem** $(G, X, \mathbf{g})$ is

$$U(\mathbf{g}) = \sum_t u_t + \sum_{\{t,t'\}} u_{tt'}. \tag{13}$$

Comparing corresponding terms in (5) and (13) yields that the problem height is an upper bound of quality, i.e., any $\mathbf{g}$ and any $\mathbf{x}$ satisfy $F(\mathbf{x}|\mathbf{g}) \leq U(\mathbf{g})$.

Unlike the quality function, the problem height is not invariant to equivalent transformations. This naturally leads to minimizing this upper bound by equivalent transformations, expressed by the linear program

$$U^*(\mathbf{g}) = \min_{\mathbf{g}' \sim \mathbf{g}} U(\mathbf{g}') \tag{14a}$$

$$= \min_{\varphi \in \mathbb{R}^P} \left[\sum_t \max_x g_t^\varphi(x) + \sum_{\{t,t'\}} \max_{x,x'} g_{tt'}^\varphi(x, x') \right]. \tag{14b}$$

Remark 3. Some equivalent transformations preserve $U(\mathbf{g})$, e.g., adding a constant to all nodes of an object and subtracting the same constant from all nodes of another object. Thus, there may be many problems with the same height within every equivalence class. This gives an option to impose constraints on u_t and $u_{tt'}$ in the minimization and reformulate (13) in a number of ways, e.g.,

$$U^*(\mathbf{g}) = \min_{\varphi \in \mathbb{R}^P | g_{tt'}^\varphi(x,x') \leq 0} \sum_t \max_x g_t^\varphi(x) \tag{15a}$$

$$= |T| \min_{\varphi \in \mathbb{R}^P | g_{tt'}^\varphi(x,x') \leq 0} \max_t \max_x g_t^\varphi(x). \tag{15b}$$

Form (15a) corresponds to imposing $u_{tt'} \leq 0$. Form (15b) corresponds to $u_{tt'} \leq 0$ and $u_t = u_{t'} = u$. Other natural constraints are $u_t = 0$ or $u_t = u_{t'} = u_{tt'}$.

4.4 Trivial Problems

Node (t, x) is a **maximal node** if $g_t(x) = u_t$. Edge $\{(t, x), (t', x')\}$ is a **maximal edge** if $g_{tt'}(x, x') = u_{tt'}$, where $\mathbf{u}$ is given by (12). Let this be expressed by Boolean variables

$$\bar{g}_t(x) = [\![g_t(x) = u_t]\!], \quad \bar{g}_{tt'}(x, x') = [\![g_{tt'}(x, x') = u_{tt'}]\!]. \tag{16}$$

TABLE 1
Equation (11)

$\langle \mathbf{g}, \boldsymbol{\alpha} \rangle \to \max_{\boldsymbol{\alpha}}$	$\sum_{t\in T} u_t + \sum_{\{t,t'\}\in E} u_{tt'} \to \min_{\boldsymbol{\varphi},\mathbf{u}}$		(11a)
$\sum_{x'\in X} \alpha_{tt'}(x,x') = \alpha_t(x)$	$\varphi_{tt'}(x) \in \mathbb{R},$	$\{t,t'\}\in E,\ x\in X$	(11b)
$\sum_{x\in X} \alpha_t(x) = 1$	$u_t \in \mathbb{R},$	$t\in T$	(11c)
$\sum_{x,x'\in X} \alpha_{tt'}(x,x') = 1$	$u_{tt'} \in \mathbb{R},$	$\{t,t'\}\in E$	(11d)
$\alpha_t(x) \geq 0$	$u_t - \sum_{t'\in N_t} \varphi_{tt'}(x) \geq g_t(x),$	$t\in T,\ x\in X$	(11e)
$\alpha_{tt'}(x,x') \geq 0$	$u_{tt'} + \varphi_{tt'}(x) + \varphi_{t't}(x') \geq g_{tt'}(x,x'),$	$\{t,t'\}\in E,\ x,x'\in X$	(11f)

A max-sum problem is **trivial** if a labeling can be formed of (some or all of) its maximal nodes and edges, i.e., if the CSP $(G, X, \bar{\mathbf{g}})$ with $\bar{\mathbf{g}}$ given by (16) is satisfiable. It is easy to see that the upper bound is tight, i.e., $F(\mathbf{x}|\mathbf{g}) = U(\mathbf{g})$ for some $\mathbf{x}$, for and only for trivial problems. This allows us to formulate the following theorem, central to the whole approach:

Theorem 4. *Let C be a class of equivalent max-sum problems. Let C contain a trivial problem. Then, any problem in C is trivial if and only if its height is minimal in C.*

Proof. Let $(G, X, \mathbf{g})$ be a trivial problem in C. Let a labeling $\mathbf{x}$ be composed of the maximal nodes and edges of $(G, X, \mathbf{g})$. Any $\mathbf{g}' \sim \mathbf{g}$ satisfies $U(\mathbf{g}') \geq F(\mathbf{x}|\mathbf{g}') = F(\mathbf{x}|\mathbf{g}) = U(\mathbf{g})$. Thus, $(G, X, \mathbf{g})$ has minimal height.

Let $(G, X, \mathbf{g})$ be a nontrivial problem with minimal height in C. Any $\mathbf{g}' \sim \mathbf{g}$ and any optimal $\mathbf{x}$ satisfy $U(\mathbf{g}') \geq U(\mathbf{g}) > F(\mathbf{x}|\mathbf{g}) = F(\mathbf{x}|\mathbf{g}')$. Thus, C contains no trivial problem. □

Theorem 4 allows us to divide the solution of a max-sum problem into two steps:

1. minimize the problem height by equivalent transformations and
2. test the resulting problem for triviality.

If the resulting problem with minimal height is trivial, i.e., $(G, X, \bar{\mathbf{g}})$ is satisfiable, then $L_{G,X}(\mathbf{g}) = \bar{L}_{G,X}(\bar{\mathbf{g}})$. If not, by Theorem 4, the max-sum problem has no trivial equivalent and remains unsolved. In the former case, the relaxation (8) is tight and, in the latter case, it is not.

Testing for triviality is NP-complete, equivalent to CSP. Thus, recognizing whether a given upper bound is tight is NP-complete. Even if we *knew* that a given upper bound $U(\mathbf{g})$ was tight, finding a labeling $\mathbf{x}$ such that $F(\mathbf{x}|\mathbf{g}) = U(\mathbf{g})$ still would be NP-complete. We can prove or disprove the tightness of an upper bound only in special cases, such as those given by Theorem 2.

Fig. 3, giving examples of CSPs, can also be interpreted in terms of triviality if we imagine that the black nodes are maximal, the white nodes are nonmaximal, and the shown edges are maximal. Then, Fig. 3a shows a trivial problem (thus having minimal height), 3b shows a problem with a nonminimal height (hence nontrivial), and 3c shows a nontrivial problem with minimal height.

Note that not every polynomially solvable subclass of the max-sum problem has a trivial equivalent, e.g., if G is a simple loop, dynamic programming is applicable, but Fig. 3c shows that there might be no trivial equivalent.

4.5 Linear Programming Duality

The linear programs (8) and (14) are dual to each other [1, Theorem 2]. To show this, we wrote them together in (11) (Table 1) such that a constraint and its Lagrange multiplier are on the same line, as is usual in linear programming.

The pair (11) can be slightly modified, corresponding to modifications of the primal constraints (7) and imposing constraints on dual variables $\mathbf{u}$, as discussed in Remarks 1 and 3.

The duality of (8) and upper bound minimization was also independently shown by Wainwright et al. [15], [25] in the framework of convex combinations of trees. In our case, when the trees are objects and object pairs, proving the duality is more straightforward than for general trees.

Schlesinger and Kovalevsky [56] proposed elegant physical models of the pair (11). We described one of them in [42].

5 Conditions for Minimal Upper Bound

This section discusses how we can recognize that the height $U(\mathbf{g})$ of a max-sum problem is minimal among its equivalents, i.e., that $\mathbf{g}$ is optimal to (11). The main result will be that a nonempty kernel of the CSP formed by the maximal nodes and edges is necessary but not sufficient for minimal height.

To test for the optimality of (11), linear programming duality theorems [57] give us a starting point. By weak duality, any $\mathbf{g}$ and any $\boldsymbol{\alpha} \in \Lambda_{G,X}$ satisfy $\langle \mathbf{g}, \boldsymbol{\alpha} \rangle \leq U(\mathbf{g})$. By strong duality, $\langle \mathbf{g}, \boldsymbol{\alpha} \rangle = U(\mathbf{g})$ if and only if $\mathbf{g}$ has minimal height and $\boldsymbol{\alpha}$ has maximal quality. By complementary slackness, $\langle \mathbf{g}, \boldsymbol{\alpha} \rangle = U(\mathbf{g})$ if and only if $\boldsymbol{\alpha}$ is zero on nonmaximal nodes and edges.

To formalize the last statement, we define the **relaxed CSP** $(G, X, \bar{\mathbf{g}})$ as finding relaxed labelings on given nodes and edges, i.e., finding the set $\bar{\Lambda}_{G,X}(\bar{\mathbf{g}})$ of relaxed labelings $\boldsymbol{\alpha} \in \Lambda_{G,X}$ satisfying the complementarity constraints

$$[1 - \bar{g}_t(x)]\alpha_t(x) = 0, \quad [1 - \bar{g}_{tt'}(x,x')]\alpha_{tt'}(x,x') = 0. \tag{17}$$

Thus, $\bar{\Lambda}_{G,X}(\bar{\mathbf{g}})$ is the set of solutions to system (7)+(17). A CSP $(G, X, \bar{\mathbf{g}})$ is **relaxed-satisfiable** if $\bar{\Lambda}_{G,X}(\bar{\mathbf{g}}) \neq \emptyset$.

Further in this section, we let $\bar{\mathbf{g}}$ denote a function of $\mathbf{g}$ given by (16). In other words, $(G, X, \bar{\mathbf{g}})$ is not seen as an independent CSP, but it is composed of the maximal nodes and edges of the max-sum problem $(G, X, \mathbf{g})$. Complementary slackness now reads as follows:

Theorem 5. *The height of* $(G, X, \mathbf{g})$ *is minimal of all its equivalents if and only if* $(G, X, \bar{\mathbf{g}})$ *is relaxed-satisfiable. If it is so, then* $\Lambda_{G,X}(\mathbf{g}) = \bar{\Lambda}_{G,X}(\bar{\mathbf{g}})$.

5.1 Nonempty Kernel Necessary for Minimal Upper Bound

In Section 3, the concepts of arc consistency and the kernel have been shown to be useful for characterizing CSP satisfiability. They are useful also for characterizing relaxed satisfiability. To show that, we first generalize the result that taking the kernel preserves $\bar{L}_{G,X}(\bar{\mathbf{g}})$.

Theorem 6. *Let* $(G, X, \bar{\mathbf{g}}^*)$ *be the kernel of a CSP* $(G, X, \bar{\mathbf{g}})$*. Then,* $\bar{\Lambda}_{G,X}(\bar{\mathbf{g}}) = \bar{\Lambda}_{G,X}(\bar{\mathbf{g}}^*)$.

Proof. Obvious from the argument in Section 3. A formal proof in [42]. □

Thus, Theorem 2 can be extended to relaxed labelings.

Theorem 7. *A nonempty kernel of* $(G, X, \bar{\mathbf{g}})$ *is necessary for its relaxed satisfiability and, hence, for minimal height of* $(G, X, \mathbf{g})$.

Proof. An immediate corollary of Theorem 6. Alternatively, it is instructive to also consider the following dual proof:

We will denote the **height of pencil** (t, t', x) by $u_{tt'}(x) = \max_{x'} g_{tt'}(x, x')$ and call (t, t', x) a **maximal pencil** if it contains a maximal edge. Let us modify the arc consistency algorithm such that, rather than explicitly zeroing variables $\bar{\mathbf{g}}$ as in (4), nodes and edges of $(G, X, \bar{\mathbf{g}})$ are deleted by repeating the following equivalent transformations on $(G, X, \mathbf{g})$:

- Find a pencil (t, t', x) such that $u_{tt'}(x) < u_{tt'}$ and $g_t(x) = u_t$. Decrease node (t, x) by $\varphi_{tt'}(x) = \frac{1}{2}[u_{tt'} - u_{tt'}(x)]$. Increase all edges in pencil (t, t', x) by $\varphi_{tt'}(x)$.
- Find a pencil (t, t', x) such that $u_{tt'}(x) = u_{tt'}$ and $g_t(x) < u_t$. Increase node (t, x) by $\varphi_{tt'}(x) = \frac{1}{2}[u_t - g_t(x)]$. Decrease all edges in pencil (t, t', x) by $\varphi_{tt'}(x)$.

When no such pencil exists, the algorithm halts.

If the kernel of $(G, X, \bar{\mathbf{g}})$ was initially nonempty, the algorithm halts after the maximal nodes and edges that were not in the kernel are made nonmaximal. If the kernel was initially empty, the algorithm sooner or later decreases the height of some node or edge, hence, $U(\mathbf{g})$. □

The algorithm in the proof has only a theoretical value. In practice, it is useless due to its slow convergence.

5.2 Nonempty Kernel Insufficient for Minimal Upper Bound

One might hope that a nonempty kernel is not only necessary but also sufficient for relaxed satisfiability. Unfortunately, this is false, as was observed by Schlesinger [8] and, analogically in terms of convex combination of trees, by Kolmogorov [29], [31]. Figs. 5b, 5c, and 5d show counterexamples. We will justify these counterexamples first by giving a primal argument (i.e., by showing that $(G, X, \bar{\mathbf{g}})$ is not relaxed-satisfiable), then by giving a dual argument (i.e., by giving an equivalent transformation that decreases $U(\mathbf{g})$).

5.2.1 Primal Argument

Let $(G, X, \bar{\mathbf{g}}^*)$ denote the kernel of a CSP $(G, X, \bar{\mathbf{g}})$. Consider an edge $\{(t, x), (t', x')\}$. By Theorem 6, the existence of $\boldsymbol{\alpha} \in \bar{\Lambda}_{G,X}(\bar{\mathbf{g}})$ such that $\alpha_{tt'}(x, x') > 0$ implies $\bar{g}^*_{tt'}(x, x') = 1$. Less obviously, the opposite implication is false. In other words, the fact that an edge belongs to the kernel is necessary but not sufficient for some relaxed labeling to be nonzero on this edge. The same holds for nodes. Fig. 5a shows an example: It can be verified that system (7a)+(17) implies that $\alpha_{tt'}(x, x') = 0$ on the edge marked by the cross.

In Figs. 5b and 5c, the only solution to system (7a)+(17) is $\boldsymbol{\alpha} = \mathbf{0}$; therefore, $\bar{\mathbf{g}}$ is relaxed-unsatisfiable. Note that Fig. 5b contains Fig. 5a as its part.

5.2.2 Dual Argument

The analogical dual observation is that the kernel of $(G, X, \bar{\mathbf{g}})$ is not invariant to equivalent transformations of $(G, X, \mathbf{g})$. Consider the transformations in Fig. 5, depicted by nonzero values of $\varphi_{tt'}(x)$ written next to the line segments crossing edge pencils (t, t', x). In each subfigure, the shown transformation makes the edge marked by the cross nonmaximal and thus deletes it from the kernel. After this, the kernel in Fig. 5a still remains nonempty, while the kernels in Figs. 5b and 5c become empty, as is verified by doing the arc consistency algorithm by hand. Thus, in Figs. 5b and 5c, a nonempty kernel of $(G, X, \bar{\mathbf{g}})$ does not suffice for minimality of the height of $(G, X, \mathbf{g})$.

In Figs. 5b and 5c, system (7a)+(7b)+(17) has no solution, without even considering constraint (7c). Fig. 5d shows a more advanced counterexample, where system (7a)+(7b)+(17) has a (single) solution but this solution violates (7c).

5.3 Boolean Max-Sum Problems

For problems with Boolean variables ($|X| = 2$), Schlesinger observed [54] that a nonempty kernel is both necessary and sufficient for the minimal upper bound. Independently, the equivalent observation was made by Kolmogorov and Wainwright [19], [32], who showed that a weak tree agreement is sufficient for minimality of Wainwright et al.'s tree-based upper bound [25]. In addition, both noticed that, for Boolean variables, at least one relaxed labeling is half-integral; an analogical observation was made in pseudo-Boolean optimization [18], referring to [58].

Theorem 8. *Let a CSP* $(G, X, \bar{\mathbf{g}})$ *with* $|X| = 2$ *labels have a nonempty kernel. Then,* $\bar{\Lambda}_{G,X}(\bar{\mathbf{g}}) \cap \{0, \frac{1}{2}, 1\}^I \neq \emptyset$.

Proof. We will prove the theorem by constructing a relaxed labeling $\boldsymbol{\alpha} \in \bar{\Lambda}_{G,X}(\bar{\mathbf{g}}) \cap \{0, \frac{1}{2}, 1\}^I$.

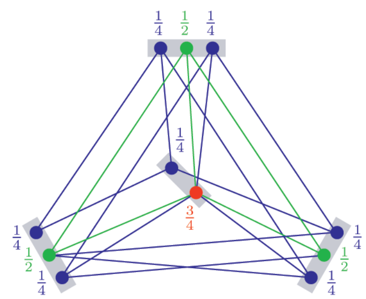
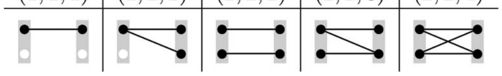


Fig. 6. A CSP for which $\bar{\Lambda}_{G,X}(\bar{\mathbf{g}})$ has a single element α that is not an integer multiple of $|X|^{-1}$. This can be verified by solving system (7a)+(7b)+(17).

Delete all nodes and edges not in the kernel. Denote the number of nodes in object t and the number of edges in pair $\{t,t'\}$ by $n_t = \sum_x \bar{g}_t(x)$ and $n_{tt'} = \sum_{x,x'} \bar{g}_{tt'}(x,x')$, respectively. All object pairs can be partitioned into five classes (up to swapping labels), indexed by triplets $(n_t, n_{t'}, n_{tt'})$:

$(1,1,1)$	$(1,2,2)$	$(2,2,2)$	$(2,2,3)$	$(2,2,4)$



Remove one edge in each pair of class (2, 2, 3) and two edges in each pair of class (2, 2, 4) such that they become (2, 2, 2). Now, there are only pairs of classes (1, 1, 1), (1, 2, 2), and (2, 2, 2). Let $\alpha_t(x) = \bar{g}_t(x)/n_t$ and $\alpha_{tt'}(x,x') = \bar{g}_{tt'}(x,x')/n_{tt'}$. Clearly, this $\boldsymbol{\alpha}$ belongs to $\bar{\Lambda}_{G,X}(\bar{\mathbf{g}})$. □

For $|X| > 2$, a relaxed labeling that is an integer multiple of $|X|^{-1}$ may not exist. A counterexample is in Fig. 6.

5.4 Summary: Three Kinds of Consistency

To summarize, we have met three kinds of "consistency," related by implications as follows:

$$\begin{matrix} \bar{\mathbf{g}} \text{ satisfiable} \\ \mathbf{g} \text{ trivial} \end{matrix} \Rightarrow \begin{matrix} \bar{\mathbf{g}} \text{ relaxed–satisfiable} \\ \text{height of } \mathbf{g} \text{ minimal} \end{matrix} \Rightarrow \text{kernel of } \bar{\mathbf{g}} \text{ nonempty.}$$

The opposite implications do not hold in general. Exceptions are problems with two labels, for which the nonempty kernel equals relaxed satisfiability, and the supermodular max-sum problems (lattice CSPs) and problems on trees for which a nonempty kernel equals satisfiability.

Testing for the first condition is NP-complete. Testing for the last condition is polynomial and simple, based on arc consistency. Testing for the middle condition is polynomial (solvable by linear programming), but we do not know any efficient algorithm to do this test for large instances. The difficulty seems to be in the fact that, while arc consistency can be tested by local operations, relaxed satisfiability is probably an inherently nonlocal property.

To the best of our knowledge, all known efficient algorithms for decreasing the height of max-sum problems use arc consistency or the nonempty kernel as their termination criterion. We will review two such algorithms in Sections 6 and 7. The existence of arc-consistent but relaxed-unsatisfiable configurations is unpleasant here because these algorithms need not find the minimal problem height. Analogical spurious minima also occur in the sequential tree-reweighted message passing (TRW-S) algorithm, as observed by Kolmogorov [28], [29], [30], [31]. Omitting a formal proof, we argue that they are of the same nature as arc-consistent relaxed-unsatisfiable states.

6 Max-Sum Diffusion

This section describes the max-sum diffusion algorithm [6], [7] to decrease the upper bound (13). It can be viewed as a coordinate descent method.

The diffusion is related to edge-based message passing by Wainwright et al. [25, algorithm 1], but, unlike the latter, it is conjectured to always converge. Also, it can be viewed as the sequential tree-reweighted message passing (TRW-S) by Kolmogorov [28], [31], with the trees being nodes and edges (we omit a detailed proof). The advantage of the diffusion is its simplicity: It is even simpler than belief propagation.

6.1 The Algorithm

The **node-pencil averaging** on pencil (t,t',x) is the equivalent transformation that makes $g_t(x)$ and $u_{tt'}(x)$ equal, i.e., which adds number $\frac{1}{2}[u_{tt'}(x) - g_t(x)]$ to $g_t(x)$ and subtracts the same number from the qualities of all edges in pencil (t,t',x). Recall that $u_{tt'}(x) = \max_{x'} g_{tt'}(x,x')$. In its simplest form, the max-sum diffusion algorithm repeats node-pencil averaging until convergence on all pencils in any order such that each pencil is visited "sufficiently often." The following code does it (with a deterministic order of pencils):

repeat
  for $(t,t',x) \in P$ **do**
    $\varphi_{tt'}(x) += \frac{1}{2}[\max_{x'} g^{\varphi}_{tt'}(x,x') - g^{\varphi}_t(x)]$;
  end for
until convergence
$\mathbf{g} := \mathbf{g}^{\varphi}$;

Remark 4. The algorithm can easily be made slightly more efficient. If a node (t,x) is fixed and node-pencil averaging is iterated on pencils $\{(t,t',x) \mid t' \in N_t\}$ till convergence, the heights of all of these pencils and $g_t(x)$ become equal. This can be done by a single equivalent transformation on node (t,x).

6.2 Monotonicity

When node-pencil averaging is done on a single pencil, the problem height can decrease, remain unchanged, or increase. For an example, when the height increases, consider a max-sum problem with $X = \{1,2\}$ such that, for some pair $\{t,t'\}$, we have $g_t(1) = g_t(2) = 1$ and $u_{tt'}(1) = u_{tt'}(2) = -1$. After the node-pencil averaging on $(t,t',1)$, $U(\mathbf{g})$ increases by 1.

Monotonic height decrease can be ensured by choosing an appropriate order of pencils, as given by Theorem 9. This shows that the diffusion is a coordinate descent method.

Theorem 9. *After the equivalent transformation consisting of $|X|$ node-pencil averagings on pencils $\{(t,t',x) \mid x \in X\}$, the problem height does not increase.*

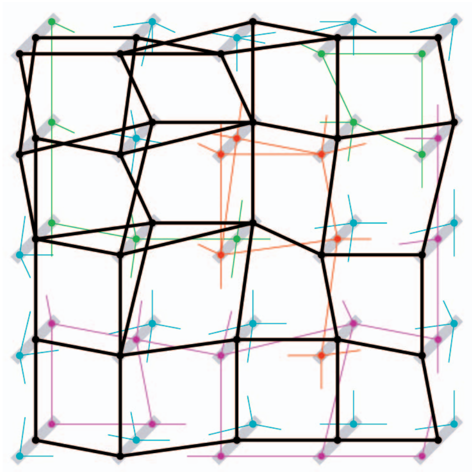


Fig. 7. A max-sum problem satisfying (18). A line segment starting from node (t,x) and aiming for but not reaching (t',x') denotes an edge satisfying $g_t(x) = g_{tt'}(x,x') < g_{t'}(x')$. If $g_t(x) = g_{tt'}(x,x') = g_{t'}(x')$, the line segment joins the nodes (t,x) and (t',x'). The gray levels help distinguish different layers; the highest layer is emphasized.

Proof. Before the transformation, the contribution of object t and pair $\{t,t'\}$ to $U(\mathbf{g})$ is $\max_x g_t(x) + \max_x u_{tt'}(x)$. After the transformation, this contribution is $\max_x[g_t(x) + u_{tt'}(x)]$. The first expression is not smaller than the second one because any two functions $f_1(x)$ and $f_2(x)$ satisfy

$$\max_x f_1(x) + \max_x f_2(x) \geq \max_x [f_1(x) + f_2(x)].$$

□

6.3 Properties of the Fixed Point

Based on numerous experiments, it was conjectured that the max-sum diffusion always converges. In addition, its fixed points can be characterized as follows:

Conjecture 1. *For any* $\mathbf{g} \in [-\infty,\infty)^I$, *the max-sum diffusion converges to a solution of the system*

$$\max_{x'} g_{tt'}(x,x') = g_t(x), \quad \{t,t'\} \in E,\ x \in X. \tag{18}$$

We are not aware of any proof of this conjecture.

Any solution to (18) has the following layered structure (see Fig. 7). A **layer** is a maximal connected subgraph of graph $(T \times X, E_X)$ such that each of its edges $\{(t,x),(t',x')\}$ satisfies $g_t(x) = g_{tt'}(x,x') = g_{t'}(x')$. By (18), all nodes and edges of a layer have the same quality, the **height of the layer**. The highest layer is formed by the maximal nodes and edges.

Property (18) implies arc consistency of the maximal nodes and edges, as given by Theorem 10. However, the converse is false: Not every max-sum problem with arc-consistent maximal nodes and edges satisfies (18).

Theorem 10. *If a max-sum problem satisfies (18), then its maximal nodes and edges form an arc-consistent CSP.*

Proof. Suppose (18) holds. By (3), we are to prove that a pencil (t,t',x) is maximal if and only if node (t,x) is maximal. If (t,x) is maximal, then $u_{tt'}(x) = g_t(x) \geq g_t(x') = u_{tt'}(x')$ for each x'; hence, (t,t',x) is maximal. If (t,x) is nonmaximal, then $u_{tt'}(x) = g_t(x) < g_t(x') = u_{tt'}(x')$ for some x'; hence, (t,t',x) is not maximal. □

Since the max-sum problems in Figs. 5b, 5c, and 5d satisfy (18), diffusion fixed points can have a nonminimal upper bound $U(\mathbf{g})$. This is a serious drawback of the algorithm.

More on max-sum diffusion can be found in recent papers [78], [79].

7 The Augmenting DAG Algorithm

This section describes the height-decreasing algorithm given in [4], [9]. Its main idea is as follows: Run the arc consistency algorithm on the maximal nodes and edges, storing the pointers to the causes of deletions. When all nodes in a single object are deleted, it is clear that the kernel is empty. Backtracking the pointers provides a directed acyclic graph (DAG), called the augmenting DAG, along which a height-decreasing equivalent transformation is done.

The algorithm has been proved to converge in a finite number of steps [1] if it is modified as follows: The maximality of nodes and edges is redefined using a threshold, ε. We will first explain the algorithm without this modification and return to it at the end of the section.

The iteration of the algorithm proceeds in three phases, described in subsequent sections. We use (15a), i.e., we look for φ that minimizes $U(\mathbf{g}^\varphi) = \sum_t \max_x g_t^\varphi(x)$ subject to the constraint that all edges are nonpositive, $g_{tt'}^\varphi(x,x') \leq 0$. Initially, all edges are assumed nonpositive.

7.1 Phase 1: Arc Consistency Algorithm

The arc consistency algorithm is run on the maximal nodes and edges. It is not done exactly as described by rules (4), but in a slightly modified way, as follows:

A variable $p_t(x) \in \{\texttt{ALIVE}, \texttt{NONMAX}\} \cup T$ is assigned to each node (t,x). Initially, we set $p_t(x) := \texttt{ALIVE}$ if (t,x) is maximal and $p_t(x) := \texttt{NONMAX}$ if (t,x) is nonmaximal.

If a pencil (t,t',x) is found satisfying $p_t(x) = \texttt{ALIVE}$ and violating condition

$$(\exists x') \begin{bmatrix} \text{edge } \{(t,x),(t',x')\} \text{ is maximal,} \\ p_{t'}(x') = \texttt{ALIVE} \end{bmatrix}, \tag{19}$$

node (t,x) is deleted by setting $p_t(x) := t'$. The object t' is called the **deletion cause** of node (t,x). This is repeated until either no such pencil exists or an object t^* is found with $p_{t^*}(x) \neq \texttt{ALIVE}$ for all $x \in X$. In the former case, the augmenting DAG algorithm halts. In the latter case, we proceed to the next phase.

After every iteration of this algorithm, the maximal edges and the variables $p_t(x)$ define a directed acyclic subgraph D of graph $(T \times X, E_X)$ as follows: The nodes of D are the end nodes of its edges; edge $((t,x),(t',x'))$ belongs to D if and only if it is maximal and $p_t(x) = t'$. Once t^* has been found, the **augmenting DAG** $D(t^*)$ is a subgraph of D reachable by a directed path in D from the maximal nodes of t^*.

Example. The example max-sum problem in Fig. 8 has $T = \{a,\dots f\}$ and the labels in each object are 1, 2, 3, numbered from bottom to top. Fig. 8a shows the maximal edges and the values of $p_t(x)$ after the first

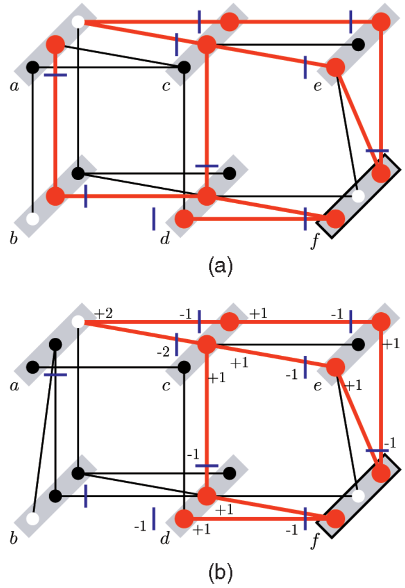


Fig. 8. (a) The augmenting DAG algorithm after phase 1; (b) after phase 2.

phase when 10 nodes have been deleted by applying rule (19) successively on pencils $(c, a, 2)$, $(c, a, 3)$, $(e, c, 1)$, $(e, c, 3)$, $(f, e, 3)$, $(d, c, 2)$, $(b, d, 2)$, $(a, b, 2)$, $(d, b, 1)$, and $(f, d, 1)$. The nonmaximal edges are not shown. The nodes with $p_t(x) = \texttt{ALIVE}$ are small filled, with $p_t(x) = \texttt{NONMAX}$ small unfilled, and with $p_t(x) \in T$ large filled. For the deleted nodes, the causes $p_t(x)$ are denoted by short segments across pencils $(t, p_t(x), x)$. The object $t^* = f$ has a black outline.

Fig. 8a shows D after t^* has been found and Fig. 8b shows $D(t^*)$. The edges of D and $D(t^*)$ are emphasized and the nodes, except $(a, 3)$, too.

7.2 Phase 2: Finding the Search Direction

The direction of height decrease is found in the space $\mathbb{R}^P$, i.e., a vector $\Delta\varphi$ is found such that $U(\mathbf{g}^{\varphi+\lambda\Delta\varphi}) < U(\mathbf{g}^{\varphi})$ for a small positive λ.

Denoting $\Delta\varphi_t(x) = \sum_{t' \in N_t} \Delta\varphi_{tt'}(x)$, the vector $\Delta\varphi$ has to satisfy

$$\begin{aligned} -\Delta\varphi_{t^*}(x) &= 1 \quad \text{if } p_{t^*}(x) \neq \texttt{NONMAX}, \\ \Delta\varphi_t(x) &\leq 0 \quad \text{if } p_t(x) \neq \texttt{NONMAX}, \\ \Delta\varphi_{tt'}(x) + \Delta\varphi_{t't}(x') &\geq 0 \quad \text{if } \{(t,x),(t',x')\} \text{ maximal}. \end{aligned}$$

We find the smallest vector $\Delta\varphi$ satisfying these. This is done by traversing $D(t^*)$ from roots to leaves, successively enforcing these constraints for all of its nodes and edges. The traversal is done in a linear order on $D(t^*)$, i.e., a node is not visited before the tails of all edges entering it have been visited. In Fig. 8b, the nonzero numbers $\Delta\varphi_{tt'}(x)$ are written near their pencils.

7.3 Phase 3: Finding the Search Step

The search step length λ is found such that no edge becomes positive, the height of no object is increased, and the height of t^* is minimized. These read, respectively,

$$\begin{aligned} g_{tt'}^{\varphi+\lambda\Delta\varphi}(x, x') &\leq 0, \\ g_t^{\varphi+\lambda\Delta\varphi}(x) &\leq \max_x g_t^{\varphi}(x), \\ g_{t^*}^{\varphi+\lambda\Delta\varphi}(x) &\leq \max_x g_{t^*}^{\varphi}(x) - \lambda. \end{aligned}$$

To justify the last inequality, see that each node of t^* with $p_{t^*}(x) \in T$ decreases by λ and each node with $p_{t^*}(x) = \texttt{NONMAX}$ increases by $\lambda\Delta\varphi_{t^*}(x)$. The latter is because $D(t^*)$ can have a leaf in t^*. To minimize the height of t^*, the nodes with $p_{t^*}(x) = \texttt{NONMAX}$ must not become higher than the nodes with $p_{t^*}(x) \in T$.

Solving the above three conditions for λ yields the system

$$\begin{aligned} \lambda &\leq \frac{g_{tt'}^{\varphi}(x, x')}{\Delta\varphi_{tt'}(x) + \Delta\varphi_{t't}(x')} \quad \text{if } \Delta\varphi_{tt'}(x) + \Delta\varphi_{t't}(x') < 0, \\ \lambda &\leq \frac{u_t - g_t^{\varphi}(x)}{[\![t = t^*]\!] + \Delta\phi_t(x)} \quad \text{if } [\![t = t^*]\!] + \Delta\varphi_t(x) > 0. \end{aligned}$$

We find the greatest λ satisfying these.

The iteration of the augmenting DAG algorithm is completed by the equivalent transformation $\varphi \mathrel{+}= \lambda\Delta\varphi$.

For implementation details, refer to [4], [42].

The algorithm sometimes spends a lot of iterations to minimize the height in a subgraph of G accurately [59]. This is wasteful because this accuracy is destroyed once the subgraph is left. This behavior, somewhat similar to the well-known inefficiency of the Ford-Fulkerson max-flow algorithm, can be reduced by redefining the maximality of nodes and edges using a threshold $\varepsilon > 0$ as follows [4]: Node (t, x) is maximal if and only if $-g_{tt'}^{\varphi}(x, x') \leq \varepsilon$ and edge $\{(t,x),(t',x')\}$ is maximal if and only if $-g_{tt'}^{\varphi}(x, x') \leq \varepsilon$. If ε is reasonably large, "nearly maximal" nodes and edges are considered maximal and, often, a larger λ results. With $\varepsilon > 0$, the algorithm terminates in a finite number of iterations [4]. A possible scheme is to run the algorithm several times, exponentially decreasing ε.

Since they are arc-consistent, the problems in Figs. 5b, 5c, and 5d are termination states of the algorithm. Thus, the algorithm can terminate with a nonminimal upper bound $U(\mathbf{g})$.

8 Supermodular Max-Sum Problems

(Super) submodularity, for bivariate functions also known as the *(inverse) Monge property* [60], is well-known to simplify many optimization tasks; in fact, it can be considered a discrete counterpart of convexity [61]. It has long been known that set supermodular max-sum problems can be translated to max-flow/min-cut [17], [62] and, therefore, are tractable. Some authors suggested this independently, e.g., Kolmogorov and Zabih [37]. Others showed translation to max-flow for other subclasses of the supermodular max-sum problem: Greig et al. [63] and Ishikawa and Geiger [64], [65] for the bivariate functions being convex univariate functions of differences of variable pairs, Cohen et al. for Max CSP [35]. Schlesinger and Flach [23] gave the translation to max-flow of the full class of supermodular max-sum problems; importantly, this is a special case of the more general result that a max-sum problem with any number of labels can be transformed into

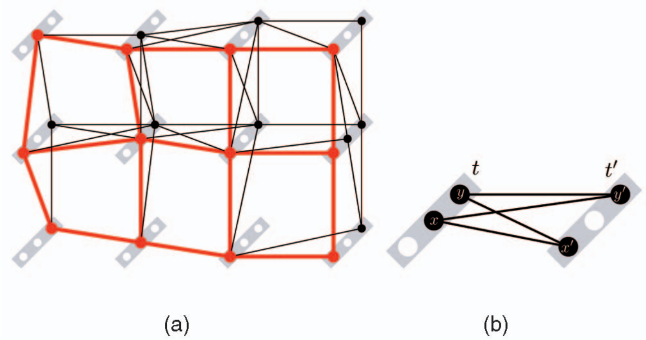


Fig. 9. (a) An arc-consistent lattice CSP is always satisfiable because a labeling on it can be found by picking the lowest label in each object separately (emphasized). (b) Supermodular max-sum problems satisfy $g_{tt'}(x,x') + g_{tt'}(y,y') \geq g_{tt'}(x,y') + g_{tt'}(y,x')$ for every $x \leq y$ and $x' \leq y'$. It follows that the poset $\bar{L}_{tt'} = \{(x,x') \mid g_{tt'}(x,x') = u_{tt'}\}$ is a lattice. In the pictures, the order $\leq$ is given by the vertical direction.

a problem with two labels [23]. In many of these works, especially in computer vision, connection with supermodularity was not noticed and the property was given ad hoc names.

The tractability of supermodular max-sum problems follows from a more general result. Their objective function is a special case of a supermodular function on a product of chains, which is, in turn, a special case of a supermodular function on a distributive lattice. Submodular functions with Boolean variables can be minimized in polynomial time [67], [68] and, for submodular functions on distributive lattices, even strongly polynomial algorithms exist [69], [70].

Linear programming relaxation (11) was shown to be tight for supermodular problems by Schlesinger and Flach [11] and, independently, for Boolean supermodular problems by Kolmogorov and Wainwright [19], [32] using a convex combination of trees [15], [25]. Further in this section, we prove this, following [11].

In particular, we will prove that, if the function $F(\mathbf{x} \mid \mathbf{g})$ (or, equivalently, the functions $g_{tt'}(\bullet,\bullet)$) is supermodular, then the max-sum problem has a trivial equivalent and finding an optimal labeling is tractable. We will proceed in two steps: First, we will show that a certain subclass of CSP is tractable and, moreover, satisfiable if its kernel is nonempty; second, we will show that the maximal nodes and edges of a supermodular problem always form a CSP in this subclass.

We assume that the label set X is endowed with a (known) total order $\leq$, i.e., the poset $(X, \leq)$ is a chain. The product $(X^n, \leq)$ of n of these chains is a distributive lattice, with the new partial order given componentwise and with meet $\wedge$ (join $\vee$) being componentwise minimum (maximum). In this section, $\wedge$ and $\vee$ denote meet and join rather than logical conjunction and disjunction. See [42], [55], [71] for background on lattices and supermodularity.

Let $\bar{L}_{tt'} = \{(x,x') \mid \bar{g}_{tt'}(x,x') = 1\}$. We call $(G,X,\bar{\mathbf{g}})$ a **lattice CSP** if the poset $(\bar{L}_{tt'}, \leq)$ is a lattice (i.e., is closed under meet and join) for every $\{t,t'\} \in E$. Note that it easily follows that, for a lattice CSP, $\bar{L}_{G,X}(\bar{\mathbf{g}})$ is also a lattice. Theorem 11 shows that lattice CSPs are tractable.

Theorem 11. *Any arc-consistent lattice CSP $(G,X,\bar{\mathbf{g}})$ is satisfiable. The "lowest" labeling $\mathbf{x} = \bigwedge \bar{L}_{G,X}(\bar{\mathbf{g}})$ is given by $x_t = \min\{x \in X \mid \bar{g}_t(x) = 1\}$ (x_t are the components of $\mathbf{x}$).*

Proof. It is obvious from Fig. 9a that the "lowest" nodes and edges form a labeling. Here is a formal proof.

Let $x_t = \min\{x \in X \mid \bar{g}_t(x) = 1\}$. We will show that $\bar{g}_{tt'}(x_t, x_{t'}) = 1$ for $\{t,t'\} \in E$. Pick $\{t,t'\} \in E$. By (3), pencil (t,t',x_t) contains at least one edge, while pencils $\{(t,t',x) \mid x < x_t\}$ are empty. Similarly for pencils $(t',t,x_{t'})$ and $\{(t',t,x') \mid x' < x_{t'}\}$. Since $(\bar{L}_{tt'}, \leq)$ is a lattice, the meeting of the edges in pair $\{t,t'\}$ is $\{(t,x_t),(t',x_{t'})\}$. □

Recall that a function $f : A \to \mathbb{R}$ on a lattice $(A, \leq)$ is **supermodular** if all $a, b \in A$ satisfy

$$f(a \wedge b) + f(a \vee b) \geq f(a) + f(b). \tag{20}$$

In particular, a bivariate function f (i.e., $(A, \leq)$ is a product of two chains, $(X^2, \leq)$) is supermodular if and only if $x \leq y$ and $x' \leq y'$ implies $f(x,x') + f(y,y') \geq f(x,y') + f(y,x')$.

We say $(G,X,\mathbf{g})$ is a **supermodular max-sum problem** if all of the functions $g_{tt'}(\bullet,\bullet)$ are supermodular on $(X^2, \leq)$. The following theorem shows that this is equivalent to supermodularity of the function $F(\bullet|\mathbf{g})$.

Theorem 12. *The function $F(\bullet|\mathbf{g})$ is supermodular if and only if all of the bivariate functions $g_{tt'}(\bullet,\bullet)$ are supermodular.*

Proof. The *if* part is true because, by (20), a sum of supermodular functions is supermodular.

The *only if* part. Pick a pair $\{t,t'\}$. Let two labelings $\mathbf{x},\mathbf{y} \in X^T$ be equal in all objects except t and t', where they satisfy $x_t \leq x_{t'}$ and $y_t \geq y_{t'}$. If $F(\bullet|\mathbf{g})$ is supermodular, by (20) it is $F(\mathbf{x} \wedge \mathbf{y} \mid \mathbf{g}) + F(\mathbf{x} \vee \mathbf{y} \mid \mathbf{g}) \geq F(\mathbf{x} \mid \mathbf{g}) + F(\mathbf{y} \mid \mathbf{g})$. After substitution from (5) and some manipulations, we are left with

$$g_{tt'}(x_t, y_{t'}) + g_{tt'}(y_t, x_{t'}) \geq g_{tt'}(x_t, x_{t'}) + g_{tt'}(y_t, y_{t'}).$$

□

Function $F(\bullet \mid \mathbf{g})$ is invariant to equivalent transformations. Theorem 12 implies that the supermodularity of $g_{tt'}(\bullet,\bullet)$ also is. This is also seen from the fact that an equivalent transformation means adding a zero problem, which is modular, and supermodularity is preserved by adding a modular function.

The following theorem shows that the maximal nodes and edges of a supermodular problem form a lattice CSP.

Theorem 13 [55]. *The set A^* of maximizers of a supermodular function f on a lattice A is a sublattice of A.*

Proof. Let $a, b \in A^*$. Denote $p = f(a) = f(b)$, $q = f(a \wedge b)$, and $r = f(a \vee b)$. The maximality of p implies $p \geq q$ and $p \geq r$. The supermodularity condition $q + r \geq 2p$ yields $p = q = r$. □

The theorem can be applied to function f being either $g_{tt'}(\bullet,\bullet)$ or $F(\bullet \mid \mathbf{g})$. This completes the proof that every supermodular max-sum problem has a trivial equivalent and is tractable.

9 Application to Structural Image Analysis

Even if this paper primarily focuses on theory, we present an example of applying the approach to structural image

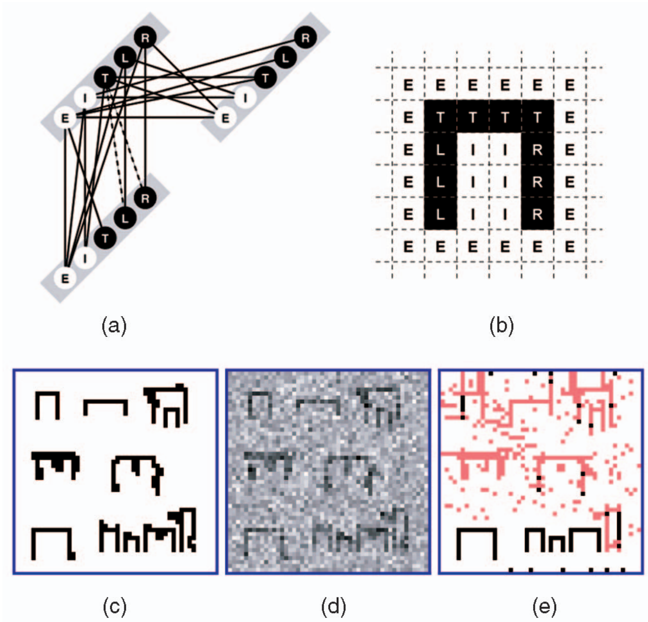


Fig. 10. The "Letters Π" example. (a) The vertical and horizontal pixel pair defining the problem. (b) A labeled image feasible to this definition. The input image in (d) is the image in (c) plus independent Gaussian noise. (e) The output image. Image size 50×50 pixels.

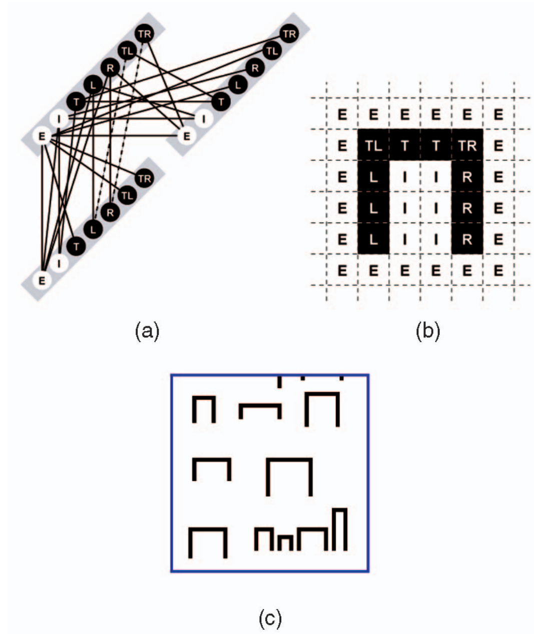


Fig. 11. The "Letters Π 2" example, an alternative "better" definition of "Letters Π." (a) Definition, (b) a feasible labeled image, (c) output. The input was Fig. 10d.

analysis. It is motivated by those in [1], [9] and we give more such examples in [42]. The task is different from nonsupermodular problems of the Potts type and arising from stereo reconstruction, experimentally examined in [19], [29], [30], [32], [72], [73], in the fact that a lot of edge qualities are $-\infty$. In that, our example is closer to CSP. In the sense of [1], [9], it can be interpreted as finding the "nearest" image belonging to the language generated by a given 2D grammar (in full generality, 2D grammars also include hidden variables). If qualities are viewed as log-likelihoods, the task corresponds to finding the maximum of a Gibbs distribution.

Let the following be given: Let G represent a four-connected image grid. Each pixel $t \in T$ has a label from $X = \{\mathrm{E}, \mathrm{I}, \mathrm{T}, \mathrm{L}, \mathrm{R}\}$. Numbers $g_{tt'}(x, x')$ are given by Fig. 10a, which shows three pixels forming one horizontal and one vertical pair, as follows: The solid edges have quality 0, the dashed edges $-\frac{1}{2}$, and the edges not shown $-\infty$. The functions $g_{tt'}(\bullet, \bullet)$ for all vertical pairs are equal, as well as for all horizontal pairs.

Numbers $f(\mathrm{E}) = f(\mathrm{I}) = 1$ and $f(\mathrm{T}) = f(\mathrm{L}) = f(\mathrm{R}) = 0$ assign an intensity to each label. Thus, $\mathbf{f}(\mathbf{x}) = \{f(x_t) \mid t \in T\}$ is the black-and-white image corresponding to labeling $\mathbf{x}$.

First, assume that $g_t(x) = 0$ for all t and x. The set $\{\mathbf{f}(\mathbf{x}) \mid F(\mathbf{x} \mid \mathbf{g}) > -\infty\}$ contains images feasible to the 2D grammar $(G, X, \mathbf{g})$, here, images of multiple nonoverlapping black "free-form" characters "Π" on a white background. An example of such an image with labels denoted is in Fig. 10b. The number of characters in the image is $-F(\mathbf{x}|\mathbf{g})$.

Let an input image $\{f_t \mid t \in T\}$ be given. The numbers $g_t(x) = -c[f_t - f(x)]^2$ quantify similarity between the input image and the intensities of the labels; we set $c = \frac{1}{6}$. Setting the dashed edges in Fig. 10a to a nonzero value discourages images with a large number of small characters, which can be viewed as a regularization.

For the input in Fig. 10d, we minimized the height of the max-sum problem $(G, X, \mathbf{g})$ by the augmenting DAG algorithm and then computed the kernel of the CSP formed by the maximal nodes and edges. To get a partial and suboptimal solution to the CSP, we used the unique label condition from Theorem 2. The result is in Fig. 10e. A pixel t with a unique maximal node (t, x) is black or white; as given by $f(x)$, a pixel with multiple maximal nodes is gray. Unfortunately, there are many ambiguous pixels.

It turns out that, if X and $\mathbf{g}$ are redefined by adding two more labels, as shown in Fig. 11, a unique label in each pixel is obtained. We observed this repeatedly: Of several formulations of the max-sum problem defining the same feasible set $\{\mathbf{f}(\mathbf{x}) \mid F(\mathbf{x}|\mathbf{g}) > -\infty\}$, some (usually not the simplest ones) provide tight upper bounds more often.

For Fig. 10, the runtime of the augmenting DAG algorithm (the implementation [42]) was 1.6 s on a 1.2-GHz laptop PC and the max-sum diffusion achieved the state with arc-consistent maximal nodes and edges in almost 8 min (maximality threshold 10^{-6}, double arithmetic). For Fig. 11, the augmenting DAG algorithm took 0.3 s and the diffusion 20 s.

10 CONCLUSION

We have reviewed the approach to the max-sum problem by Schlesinger et al. in a unified and self-contained framework.

The fact that, due to nonoptimal fixed points, no efficient algorithm to minimize the upper bound $U(\mathbf{g})$ is known is the *most serious open question*. This is not only a gap in theory, but also relevant in applications because the difference between the true and a spurious minimum can be arbitrarily large.

To present the approach by Schlesinger et al. in a single paper, we had to omit some issues for lack of space. We have omitted a detailed formal comparison with the work by Wainwright et al. [19], [25], [31]. We have not discussed the relation to other continuous relaxations [20], [21], [22], [23], to α-expansions and $\alpha\beta$-swaps [74], and to primal-dual schema [75]. We have not done an experimental comparison of the max-sum diffusion and the augmenting DAG algorithms with other approximative algorithms for the max-sum problem [73], [76], [77]. We have not discussed persistency (partial optimality) results by Kolmogorov and Wainwright [19] for Boolean variables and by Kovtun [40], [41] for the (NP-hard) Potts model.

ACKNOWLEDGMENTS

The author's work has been supported by the the European Union, grant IST-2004-71567. The paper could not have been written without Václav Hlaváč, who established the cooperation of the author's group with the Kiev and Dresden groups in 1996 and who has been supporting it since then, and without the author's personal communication with Mikhail I. Schlesinger. Christoph Schnör, Alexander Shekhovtsov, Václav Hlaváč, Mirko Navara, Tomáš Pajdla, Jiri Matas, and Vojtech Franc provided the author with valuable comments.

REFERENCES

[1] M.I. Schlesinger, "Sintaksicheskiy analiz dvumernykh zritelnikh signalov v usloviyakh pomekh (Syntactic Analysis of Two-Dimensional Visual Signals in Noisy Conditions)," *Kibernetika,* vol. 4, pp. 113-130, 1976, in Russian.

[2] M.L. Minsky and S.A. Papert, *Perceptrons: An Introduction to Computational Geometry,* second ed. MIT Press, 1988.

[3] A. Mackworth, "Constraint Satisfaction," *Encyclopedia of Artificial Intelligence,* S. Shapiro, ed., pp. 285-292, Wiley, 1991.

[4] V.K. Koval and M.I. Schlesinger, "Dvumernoe programmirovanie v zadachakh analiza izobrazheniy (Two-Dimensional Programming in Image Analysis Problems)," *USSR Academy of Science, Automatics, and Telemechanics,* vol. 8, pp. 149-168, 1976, in Russian.

[5] V.A. Kovalevsky, M.I. Schlesinger, and V.K. Koval, "Ustrojstvo dlya analiza seti," Patent No. 576843, USSR, priority of 4 January 1976, 1977, in Russian.

[6] V.A. Kovalevsky and V.K. Koval, "A Diffusion Algorithm for Decreasing Energy of Max-Sum Labeling Problem," Glushkov Inst. of Cybernetics, Kiev, USSR, unpublished, 1975.

[7] B. Flach, "A Diffusion Algorithm for Decreasing Energy of Max-Sum Labeling Problem," Fakultät Informatik, Technische Universität Dresden, Germany, unpublished, 1998.

[8] M.I. Schlesinger, "False Minima of the Algorithm for Minimizing Energy of Max-Sum Labeling Problem," Glushkov Inst. of Cybernetics, Kiev, USSR, unpublished, 1976.

[9] M.I. Schlesinger, *Matematicheskie sredstva obrabotki izobrazheniy (Mathematical Tools of Image Processing),* Naukova Dumka, 1989, in Russian.

[10] M.I. Schlesinger and V. Hlaváč, *Ten Lectures on Statistical and Structural Pattern Recognition,* M.A. Viergever, ed., Kluwer Academic, 2002.

[11] M.I. Schlesinger and B. Flach, "Some Solvable Subclasses of Structural Recognition Problems," *Proc. Czech Pattern Recognition Workshop,* 2000.

[12] A. Koster, C.P.M. van Hoesel, and A.W.J. Kolen, "The Partial Constraint Satisfaction Problem: Facets and Lifting Theorems," *Operations Research Letters,* vol. 23, nos. 3-5, pp. 89-97, 1998.

[13] A. Koster, "Frequency Assignment—Models and Algorithms," PhD dissertation, Universiteit Maastricht, Maastricht, The Netherlands, 1999.

[14] C. Chekuri, S. Khanna, J. Naor, and L. Zosin, "Approximation Algorithms for the Metric Labeling Problem via a New Linear Programming Formulation," *Proc. Symp. Discrete Algorithms,* pp. 109-118, 2001.

[15] M. Wainwright, T. Jaakkola, and A. Willsky, "MAP Estimation via Agreement on (Hyper)Trees: Message Passing and Linear Programming Approaches," *Proc. Allerton Conf. Comm., Control, and Computing,* 2002.

[16] C.L. Kingsford, B. Chazelle, and M. Singh, "Solving and Analyzing Side-Chain Positioning Problems Using Linear and Integer Programming," *Bioinformatics,* vol. 21, no. 7, pp. 1028-1039, 2005.

[17] E. Boros and P.L. Hammer, "Pseudo-Boolean Optimization," *Discrete Applied Math.,* vol. 123, nos. 1-3, pp. 155-225, 2002.

[18] P.L. Hammer, P. Hansen, and B. Simeone, "Roof Duality, Complementation and Persistency in Quadratic 0-1 Optimization," *Math. Programming,* vol. 28, pp. 121-155, 1984.

[19] V.N. Kolmogorov and M.J. Wainwright, "On the Optimality of Tree-Reweighted Max-Product Message-Passing," *Proc. Conf. Uncertainty in Artificial Intelligence (UAI),* 2005.

[20] T. Wierschin and S. Fuchs, "Quadratic Minimization for Labeling Problems," technical report, Technical Univ. Dresden, Germany, 2002.

[21] P. Ravikumar and J. Lafferty, "Quadratic Programming Relaxations for Metric Labeling and Markov Random Field MAP Estimation," *Proc. Int'l Conf. Machine Learning (ICML),* 2006.

[22] M.J. Wainwright and M.I. Jordan, "Semidefinite Relaxations for Approximate Inference on Graphs with Cycles," *Proc. Conf. Neural Information Processing Systems (NIPS),* 2003.

[23] D. Schlesinger and B. Flach, "Transforming an Arbitary MinSum Problem into a Binary One," Technical Report TUD-FI06-01, Dresden Univ. of Technology, Germany, Apr. 2006.

[24] M.J. Wainwright, T. Jaakkola, and A.S. Willsky, "A New Class of Upper Bounds on the Log Partition Function," *IEEE Trans. Information Theory,* vol. 51, no. 7, pp. 2313-2335, 2005.

[25] M. Wainwright, T. Jaakkola, and A. Willsky, "MAP Estimation via Agreement on (Hyper)Trees: Message Passing and Linear Programming Approaches," *IEEE Trans. Information Theory,* vol. 51, no. 11, pp. 3697-3717, 2005.

[26] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree-Based Reparameterization Framework for Analysis of Sum-Product and Related Algorithms," *IEEE Trans. Information Theory,* vol. 49, no. 5, pp. 1120-1146, 2003.

[27] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree Consistency and Bounds on the Performance of the Max-Product Algorithm and Its Generalizations," *Statistics and Computing,* vol. 14, pp. 143-166, 2004.

[28] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," Technical Report MSR-TR-2004-90, Microsoft Research, 2004.

[29] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," Technical Report MSR-TR-2005-38, Microsoft Research, 2005.

[30] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," *Proc. Int'l Workshop Artificial Intelligence and Statistics (AISTATS),* 2005.

[31] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1568-1583, Oct. 2006.

[32] V. Kolmogorov and M. Wainwright, "On the Optimality of Tree-Reweighted Max-Product Message-Passing," Technical Report MSR-TR-2004-37, Microsoft Research, 2005.

[33] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[34] J.S. Yedidia, W.T. Freeman, and Y. Weiss, "Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms," *IEEE Trans. Information Theory,* vol. 51, no. 7, pp. 2282-2312, 2005.

[35] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin, "Supermodular Functions and the Complexity of Max CSP," *Discrete Applied Math.,* vol. 149, nos. 1-3, pp. 53-72, 2005.

[36] S. Bistarelli, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, and H. Fargier, "Semiring-Based CSPs and Valued CSPs: Frameworks, Properties, and Comparison," *Constraints,* vol. 4, no. 3, pp. 199-240, 1999.

[37] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *Proc. European Conf. Computer Vision (ECCV),* pp. 65-81, 2002.

[38] D. Schlesinger, "Strukturelle Ansätze für die Stereorekonstruktion," PhD dissertation, Technische Universität Dresden, Fakultät Informatik, Institut für Künstliche Intelligenz, July 2005, in German.

[39] B. Flach, "Strukturelle Bilderkennung," technical report, Fakultät Informatik, Technische Universität Dresden, Germany, 2002 habilitation thesis, in German.

[40] I. Kovtun, "Partial Optimal Labelling Search for a NP-Hard Subclass of (max,+) Problems," *Proc. Conf. German Assoc. for Pattern Recognition (DAGM),* pp. 402-409, 2003.

[41] I. Kovtun, "Segmentaciya zobrazhen na usnovi dostatnikh umov optimalnosti v NP-povnikh klasakh zadach strukturnoi rozmitki (Image Segmentation Based on Sufficient Conditions of Optimality in NP-Complete Classes of Structural Labeling Problems)," PhD dissertation, IRTC ITS Nat'l Academy of Science Ukraine, Kiev, 2004, in Ukrainian.

[42] T. Werner, "A Linear Programming Approach to Max-Sum Problem: A Review," Technical Report CTU-CMP-2005-25, Center for Machine Perception, Czech Technical Univ., Dec. 2005.

[43] S. Verdú and H.V. Poor, "Abstract Dynamic Programming Models under Commutativity Conditions," *SIAM J. Control and Optimization,* vol. 25, no. 4, pp. 990-1006, July 1987.

[44] S. Bistarelli, U. Montanari, and F. Rossi, "Semiring-Based Constraint Satisfaction and Optimization," *J. ACM,* vol. 44, no. 2, pp. 201-236, 1997.

[45] S. Gaubert, "Methods and Applications of (max,+) Linear Algebra," Technical Report 3088, Institut National de Recherche en Informatique et en Automatique (INRIA), 1997.

[46] S.M. Aji and R.J. McEliece, "The Generalized Distributive Law," *IEEE Trans. Information Theory,* vol. 46, no. 2, pp. 325-343, 2000.

[47] D.L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," technical report, Massachusetts Inst. of Technology, Cambridge, Mass., 1972.

[48] U. Montanari, "Networks of Constraints: Fundamental Properties and Application to Picture Processing," *Information Science,* vol. 7, pp. 95-132, 1974.

[49] A. Rosenfeld, R.A. Hummel, and S.W. Zucker, "Scene Labeling by Relaxation Operations," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 6, no. 6, pp. 420-433, June 1976.

[50] A.K. Mackworth, "Consistency in Networks of Relations," *Artificial Intelligence,* vol. 8, no. 1, pp. 65-73, 1977.

[51] R.M. Haralick and L.G. Shapiro, "The Consistent Labeling Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, pp. 173-184, 1979.

[52] A. Bulatov, P. Jeavons, and A. Krokhin, "Classifying the Complexity of Constraints Using Finite Algebras," *Computing,* vol. 34, no. 3, pp. 720-742, 2005.

[53] R. Debruyne and C. Bessière, "Domain Filtering Consistencies," *J. Artificial Intelligence Research,* no. 14, pp. 205-230, May 2001.

[54] M.I. Schlesinger, "Lectures on Labeling Problems Attended by the Author of this Article in Kiev, Prague, Dresden," 1996-2006.

[55] D.M. Topkis, "Minimizing a Submodular Function on a Lattice," *Operations Research,* vol. 26, no. 2, pp. 305-321, 1978.

[56] M.I. Schlesinger and V. Kovalevsky, "A Hydraulic Model of a Linear Programming Relaxation of Max-Sum Labeling Problem," Glushkov Inst. of Cybernetics, Kiev, USSR, unpublished, 1978.

[57] R.J. Vanderbei, *Linear Programming: Foundations and Extensions.* Kluwer Academic, 1996.

[58] M.L. Balinski, "Integer Programming: Methods, Uses, Computation," *Management Science,* vol. 12, no. 3, pp. 253-313, 1965.

[59] M.I. Schlesinger, personal communication, Int'l Research and Training Centre, Kiev, Ukraine, 2000-2005.

[60] R.E. Burkard, B. Klinz, and R. Rudolf, "Perspectives of Monge Properties in Optimization," *Discrete Applied Math.,* vol. 70, no. 2, pp. 95-161, 1996.

[61] L. Lovász, "Submodular Functions and Convexity," *Math. Programming—The State of the Art,* A. Bachem, M. Grötschel, and B. Korte, eds., pp. 235-257, Springer-Verlag, 1983.

[62] P.L. Hammer, "Some Network Flow Problems Solved with Pseudo-Boolean Programming," *Operations Research,* vol. 13, pp. 388-399, 1965.

[63] D. Greig, B. Porteous, and A. Seheult, "Exact Maximum A Posteriori Estimation for Binary Images," *J. Royal Statistical Soc. B,* no. 51, pp. 271-279, 1989.

[64] H. Ishikawa and D. Geiger, "Segmentation by Grouping Junctions," *Proc. IEEE Computer Vision and Pattern Recognition Conf. (CVPR),* pp. 125-131, 1998.

[65] H. Ishikawa, "Exact Optimization for Markov Random Fields with Convex Priors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 10, pp. 1333-1336, Oct. 2003.

[66] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin, "A Maximal Tractable Class of Soft Constraints," *J. Artificial Intelligence Research,* vol. 22, pp. 1-22, 2004.

[67] M. Grötschel, L. Lovász, and A. Schrijver, "The Ellipsoid Method and Its Consequences in Combinatorial Optimization," *Combinatorica,* vol. 1, no. 2, pp. 169-197, 1981.

[68] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization,* second ed. Springer Verlag, 1993.

[69] A. Schrijver, "A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time," *Combinatorial Theory, Series B,* vol. 80, no. 2, pp. 346-355, 2000.

[70] S. Iwata, L. Fleischer, and S. Fujishige, "A Combinatorial Strongly Polynomial-Time Algorithm for Minimizing Submodular Functions," *J. ACM,* vol. 48, pp. 761-777, 2001.

[71] B.A. Davey and H.A. Priestley, *Introduction to Lattices and Order.* Cambridge Univ. Press, 1990.

[72] T. Meltzer, C. Yanover, and Y. Weiss, "Globally Optimal Solutions for Energy Minimization in Stereo Vision Using Reweighted Belief Propagation," *Proc. Int'l Conf. Computer Vision (ICCV),* pp. 428-435, June 2005.

[73] 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," *Proc. European Conf. Computer Vision (ECCV),* pp. II:16-29, 2006.

[74] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

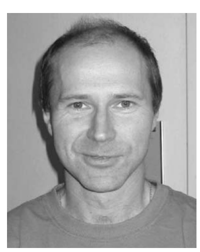
[75] N. Komodakis and G. Tziritas, "A New Framework for Approximate Labeling via Graph Cuts," *Proc. Int'l Conf. Computer Vision (ICCV),* pp. 1018-1025, 2005.

[76] V. Kolmogorov and C. Rother, "Comparison of Energy Minimization Algorithms for Highly Connected Graphs," *Proc. European Conf. Computer Vision (ECCV),* pp. II:1-15, 2006.

[77] C. Yanover, T. Meltzer, and Y. Weiss, "Linear Programming Relaxations and Belief Propagation: An Empirical Study," *J. Machine Learning Research,* vol. 7, pp. 1887-1907, Sept. 2006.

[78] T. Werner and A. Shekhovtsov, "Unified Framework for Semiring-Based Arc Consistency and Relaxation Labeling," *Proc. 12th Computer Vision Winter Workshop,* M. Grabner and H. Grabner, eds., pp. 27-34, Feb. 2007.

[79] T. Werner, "What Is Decreased by the Max-Sum Arc Consistency Algorithm?" *Proc. Int'l Conf. Machine Learning,* June 2007.



Tomáš Werner has spent most of his career at the Center for Machine Perception, a research group at the Czech Technical University in Prague. In 1999, he defended his PhD thesis on image-based rendering and became a research fellow there. In 2001-2002, he spent a year and a half in the Visual Geometry Group at Oxford University, United Kingdom, and then returned to Prague. His main interests are multiple view geometry and, recently, optimization and Markov random fields. He is a member of the IEEE Computer Society.