Revisiting the Linear Programming Relaxation Approach to Gibbs Energy Minimization and Weighted Constraint Satisfaction

Tomáš Werner

Abstract—We present a number of contributions to the LP relaxation approach to weighted constraint satisfaction (= Gibbs energy minimization). We link this approach to many works from constraint programming, which relation has so far been ignored in machine vision and learning. While the approach has been mostly considered only for binary constraints, we generalize it to n-ary constraints in a simple and natural way. This includes a simple algorithm to minimize the LP-based upper bound, n-ary max-sum diffusion – however, we consider using other bound-optimizing algorithms as well. The diffusion iteration is tractable for a certain class of high-arity constraints represented as a black-box, which is analogical to propagators for global constraints CSP. Diffusion exactly solves permuted n-ary supermodular problems. A hierarchy of gradually tighter LP relaxations is obtained simply by adding various zero constraints and coupling them in various ways to existing constraints. Zero constraints can be added incrementally, which leads to a cutting plane algorithm. The separation problem is formulated as finding an unsatisfiable subproblem of a CSP.

Index Terms—weighted constraint satisfaction, Gibbs distribution, graphical model, Markov random field, linear programming relaxation, marginal polytope, cut polytope, cutting plane algorithm, global constraint, supermodularity, tree-reweighted max-product

1 Introduction

The topic of this paper is the following problem: given a set of discrete variables and a set of functions each depending on a subset of the variables, maximize the sum of the functions over all the variables. This NP-hard combinatorial optimization problem is known as the weighted (valued, soft) constraint satisfaction problem (WCSP) [1], minimizing Gibbs energy, or finding the most probable configuration of a Markov random field. For Boolean (= two-state) variables, it becomes *pseudo-Boolean optimization* [2]. The WCSP is useful in many fields, such as AI or machine vision and learning.

One of the approaches to WCSP is the linear programming (LP) relaxation, first proposed by Schlesinger [3]. The WCSP is formulated as an integer LP in which the integrality constraint is then relaxed. The dual of the resulting LP minimizes an upper bound on the WCSP optimum by equivalent transformations (reparameterizations) of the problem. Schlesinger and colleagues proposed two algorithms to decrease the bound: *max-sum diffusion* [4], [5], which averages overlapping edge max-marginals until they all coincide, and the *augmenting DAG algorithm* [6]. In general, these algorithms do not find the global minimum of the bound but only a (good) local optimum. We surveyed works [3], [4], [6] in [7], [8].

This article is a continuation of our survey [8] of the approach by Schlesinger et al. and an improved version of our paper [9]. We present the following contributions:

• *The author is with the Department of Cybernetics, Czech Technical University, Karlovo náměstí 13, 12135 Praha, Czech Republic. Email: werner@cmp.felk.cvut.cz.*

Links to constraint programming: Minimizing Gibbs energy and WCSP are closely linked to the constraint satisfaction problem (CSP) and the related field of constraint programming [10] because: (i) the WCSP upper bound is tight iff the CSP formed by active constraint values has a solution [8], (ii) CSP is a special case of WCSP, (iii) WCSP itself is subject to research in the constraints community [1], [11], [12]. Though early seminal works on using constraints in image analysis reflected the rôle of crisp constraints [13] and their relation to soft constraints [14], [3], nowadays the relation to CSP is ignored in machine vision and learning, where people speak only about MRFs and graphical models. We relate the LP relaxation approach to many results from constraint programming. This links MRF inference to a lot of relevant literature.

N-ary generalization of the LP relaxation: The LP relaxation by Schlesinger and max-sum diffusion were originally formulated for binary WCSPs [3], [8]. We generalize them to constraints of any arity: while in the binary case nodes are coupled to edges, here we couple pairs of hyperedges. Which hyperedge pairs are actually coupled is specified by the *coupling scheme*. This allows to include non-binary constraints in a native way (= not by translation to binary constraints).

High-arity and global constraints: A high-arity constraint represented by a black box is feasible to handle by max-sum diffusion whenever max-marginals of its reparameterization are tractable to compute. This is very similar to how global constraints are commonly treated in CSP.

Supermodular n-ary problems: We show that for supermodular n-ary WCSPs, any local optimum of the bound solves the WCSP exactly; here, it suffices to couple hyperedges only to variables, i.e., to achieve only generalized arc consistency. By revisiting [15], [12], we generalize this result to permuted supermodular WCSPs.

Tighter relaxations: We show that once we have a natively n-ary LP relaxation, it can be tightened simply by adding zero constraints. Dually, this means that equivalent transformations change not only the constraint values but also the problem hypergraph. Adding various zero constraints and coupling them to existing constraints in various ways yields a hierarchy of gradually tighter relaxations, corresponding to a hierarchy of nested polyhedral outer bounds of the marginal polytope. This can be done incrementally, yielding a dual cutting plane algorithm. The separation problem can be posed purely in CSP terms, such as finding an unsatisfiable sub-CSP. We relate higher-order LP relaxations to stronger local consistencies (path consistency, k-consistency) in CSP.

1.1 Other Works

Non-binary constraints, tighter relaxations and cutting-plane strategies in LP relaxation approaches to WCSP have been addressed in a number of other works.

Most similar to ours is the *decomposition* approach. The original WCSP is expressed as a sum of subproblems on tractable hypergraphs. The sum of maxima of the subproblems is an upper bound on the maximum of their sum (= the true solution). The bound is minimized over constraint values of the subproblems, subject to that they sum to a reparameterization of the original WCSP.

The approach was proposed by Wainwright et al. [16], [17], [18] for tree-structured subproblems and improved by Kolmogorov [19]. Using hypertrees rather than trees allows for natively handling non-binary constraints and yields a hierarchy of progressively tighter relaxations [17, §VI], [18]. Johnson et al. [20] used general subproblems rather than (hyper)trees, also obtaining a hierarchy of relaxations. Komodakis et al. [21] pointed out that decomposition is a standard technique in optimization [22].

Our approach can be seen equivalent to the decomposition approach. In the one direction, we decompose the WCSP into the smallest possible subproblems, individual constraints. In the other direction, if each constraint in our approach is *itself* defined as a sum of several constraints, we obtain the decomposition approach. Our adding of zero constraints is similar to constructing an *augmented hypergraph* in [20], [18].

Weiss et al. [23] extended the LP relaxation to n-ary problems in a way similar to ours, with a small but crucial difference: they couple hyperedges only to variables (rather than other hyperedges), in which case adding zero constraints does not tighten the relaxation.

A global constraint in WCSP was used by Rother et al. [24]. Our aproach is different, relying on LP relaxation.

Tighter LP relaxations of WCSP and cutting plane strategies have recently appeared in many works. These approaches can be divided into primal [25], [26] and dual [20], [27], [28], [29]. Ours is dual. Primal approaches have a drawback that no algorithms to solve the primal LP are known that scale to large problems.

Koster et al. [25] proposed the same LP relaxation as [3] (without dual) and a primal cutting plane algorithm.

Sontag and Jaakkola [26] observed the relation of the marginal polytope and the cut polytope [30], for which many classes of cutting planes are known, and adapted the algorithm [31] to separate inconsistent cycles.

Kumar and Torr [27] and Komodakis and Paragios [28] add cycles to the LP relaxation.

Sontag et al. [29] incrementally tighten the relaxation by adding clusters of variables.

N-ary generalizations of the LP relaxation to WCSP, their higher-order versions, and cutting plane strategies proposed in the above works often use different formalisms which makes it difficult to compare them – however, one can conjecture that they all yield the same hierarchy of polyhedral relaxations (or at least some of its lower levels). We offer yet another formulation, which is very simple and general. Its main strength is in its close relation to constraint programming, which allows to formulate optimality conditions and the separation problem in CSP terms and straightforwardly extends to global and n-ary (permuted) supermodular constraints.

2 Notation and Problem Formulation

2^V resp. $\binom{V}{k}$ is the set of all resp. of k-element subsets of a set V. The value $[\![\omega]\!]$ is 1 if logical expression ω is true and 0 if ω is false. $\mathbb{R}$ denotes the reals and $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty\}$. The set of mappings from X to Y is Y^X. An ordered resp. unordered tuple is denoted $(\cdots)$ resp. $\{\cdots\}$.

Let V be a finite, totally ordered set of **variables**. To emphasize the variable ordering, when defining a subset of V by enumerating its elements we use $(\cdots)$ instead of $\{\cdots\}$. Each variable $v \in V$ is assigned a finite set X_v, its **domain**. An element of X_v is a **state** of variable v and is denoted x_v. The **joint domain** of variables $A \subseteq V$ is the Cartesian product $X_A = \times_{v \in A} X_v$, where the order of the factors is determined by the order on V. A tuple $x_A \in X_A$ is a **joint state** of variables A.

Example 1. Let $V = (1, 2, 3, 4)$, $X_1 = X_2 = X_3 = X_4 = \{\mathsf{a}, \mathsf{b}\}$. A joint state $x_{134} = (x_1, x_3, x_4) \in X_{134} = X_1 \times X_3 \times X_4$ of variables $A = (1, 3, 4) \subseteq V$ is e.g. $(\mathsf{a}, \mathsf{a}, \mathsf{b})$. □

We will use the following **implicit restriction convention**: for $B \subseteq A$, whenever symbols x_A and x_B appear in a single expression they do not denote independent joint states but x_B denotes the restriction of x_A to variables B.

A **constraint** with **scope** $A \subseteq V$ is a function $f_A \colon X_A \to \bar{\mathbb{R}}$. The **arity** of the constraint is the size of its scope, $|A|$.

Let $E \subseteq 2^V$ be a set of subsets of V, i.e., a hypergraph. Each hyperedge $A \in E$ is assigned a constraint f_A. All these constraints together are understood as a single mapping $f \colon T(E, X_V) \to \bar{\mathbb{R}}$, $(A, x_A) \mapsto f_A(x_A)$, where we denoted $T(E, X_V) = \{\, (A, x_A) \mid A \in E,\ x_A \in X_A \,\}$.

The topic of this article is the problem

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

$$\sum_{A\in E}\sum_{x_A} f_A(x_A)\,\mu_A(x_A) \to \max_{\mu} \qquad\qquad \sum_{A\in E}\psi_A \to \min_{\varphi,\psi} \tag{2a}$$

$$\sum_{x_{A\setminus B}} \mu_A(x_A) = \mu_B(x_B) \qquad\qquad \varphi_{AB}(x_B) \lessgtr 0 \qquad \forall (A,B)\in J,\ x_B\in X_B \tag{2b}$$

$$\sum_{x_A} \mu_A(x_A) = 1 \qquad\qquad \psi_A \lessgtr 0 \qquad \forall A\in E \tag{2c}$$

$$\mu_A(x_A) \ge 0 \qquad \sum_{B\mid(B,A)\in J}\varphi_{AB}(x_A) - \sum_{B\mid(A,B)\in J}\varphi_{BA}(x_B) + \psi_A \ge f_A(x_A) \qquad \forall A\in E,\ x_A\in X_A \tag{2d}$$

which we will refer to as the **weighted constraint satisfaction problem** (WCSP). The WCSP instance is defined by a tuple (V, E, X_V, f). When V, E and X_V are clear from context, sometimes we will refer to the instance just as f. The **arity of the instance** is $\max_{A\in E}|A|$.

Example 2. Let $V = (1, 2, 3, 4)$ and $E = \{(2, 3, 4), (1, 2), (3, 4), (3)\}$. Problem (1) means that we maximize the function $f_{234}(x_2, x_3, x_4)+f_{12}(x_1, x_2)+f_{34}(x_3, x_4)+f_3(x_3)$ over x_1, x_2, x_3, x_4. □

3 Linear Programming Relaxation

The LP relaxation approach developed by Schlesinger [3] was originally formulated for binary WCSP. Following our survey [8], we generalize it here to n-ary WCSPs.

We start by writing the relaxation as the pair of mutually dual linear programs (2). Here, $\lessgtr 0$ means that the variable is unconstrained. In matrix form, (2) reads

$$f^\top\mu \to \max_{\mu} \qquad\qquad \psi^\top 1 \to \min_{\varphi,\psi} \tag{3a}$$

$$M\mu = 0 \qquad\qquad \varphi \lessgtr 0 \tag{3b}$$

$$N\mu = 1 \qquad\qquad \psi \lessgtr 0 \tag{3c}$$

$$\mu \ge 0 \qquad \varphi^\top M + \psi^\top N \ge f^\top \tag{3d}$$

The pairs (2) and (3) are written such that a constraint and its Lagrange multiplier is always on the same line.

Besides E, X_V and f, the LP is described by a set

$$J \subseteq I(E) = \{\,(A,B) \mid A\in E,\ B\in E,\ B\subset A\,\} \tag{4}$$

where $I(E)$ denotes the (strict) inclusion relation on E. We refer to the set J as the **coupling scheme**.

In §3.1, §3.2 we explain the primal and dual in detail.

3.1 Primal Program

In the primal LP, each hyperedge $A \in E$ is assigned a function μ_A: $X_A \to [0, 1]$, where primal constraints (2c)+(2d) impose that μ_A is a probability distribution. All the distributions together are understood as a mapping μ: $T(E, X_V) \to [0, 1]$. While the constraints (2c)+(2d) affect each distribution separately, constraint (2b) couples some pairs of distributions, imposing that they have consistent marginals. The coupling scheme J determines which pairs of distributions are actually coupled.

Example 3. Let $A = (1, 2, 3)$, $B = (2, 3)$. Then (2b) reads: $\forall x_2 \in X_2, x_3 \in X_3$: $\sum_{x_1}\mu_{123}(x_1, x_2, x_3) = \mu_{23}(x_2, x_3)$. □

If μ is integral, μ: $T(E, X_V) \to \{0, 1\}$, then, on certain conditions on E and J, the primal LP is equivalent to WCSP. Theorem 1 shows that for this equivalence to hold, it suffices to couple hyperedges to variables [23].

Theorem 1. *Let $\binom{V}{1} \subseteq E$ and $I_{\mathrm{GAC}}(E) \subseteq J$, where*

$$I_{\mathrm{GAC}}(E) = \{\,(A,(v)) \mid A\in E,\ |A|>1,\ v\in A\,\} \tag{5}$$

Then the primal LP with integral μ is equivalent to (1).

Proof: Let μ be integral. Then μ_A represents a single joint state, x_A. Thus, $f_A(x_A) = \sum_{y_A} f_A(y_A)\mu_A(y_A)$. Equality (2b) means that the joint state represented by μ_B is the *restriction* of the joint state represented by μ_A on variables B. If $I_{\mathrm{GAC}}(E) \subseteq J$ then fixing μ_v for all $v \in V$ uniquely determines μ_A for all $A \in E$. □

If $I_{\mathrm{GAC}}(E) \not\subseteq J$ then the primal LP can have integral optimal solutions that are not solutions of (1).

To conclude, the primal LP is a relaxation of the WCSP. The relaxation is twofold: first, μ is allowed to be continuous rather than integral, second, only a subset, J, of possible marginalization constraints is imposed. Clearly, the primal optimum is an *upper bound* on (1).

3.1.1 *Alternative Forms of Marginal Consistency*

The marginal consistency condition (and the coupling scheme) could be formulated in several alternative ways, different from (2b). We state these alternative forms here.

First, (2b) can be stated in a symmetric form as

$$\sum_{x_{A\setminus C}} \mu_A(x_A) = \sum_{x_{B\setminus C}} \mu_B(x_B) \qquad \begin{cases} \forall (A,B,C)\in J \\ \forall x_C\in X_C \end{cases} \tag{6}$$

where $J \subseteq I(E) = \{\,(A,B,C) \mid A,B\in E,\ \emptyset \ne C \subseteq A\cap B\,\}$. Form (6) may appear more general than the asymmetric form (2b); e.g., if $A\cap B \ne \emptyset$ and $B \not\subseteq A$ then equality (2b) is vacuous and (6) is not. But this is not so because equality (6) applied on (A, B, C) is equivalent to two equalities (2b) applied on (A, C) and (B, C). This assumes that $C \in E$, which can be ensured by adding the zero constraint with scope C (see Example 6 in §9).

Second, while equality (2b) is imposed on all joint states $x_B \in X_B$, we could impose it only on their subset:

$$\sum_{x_{A\setminus B}} \mu_A(x_A) = \mu_B(x_B) \qquad \forall (A,B,x_B)\in J \tag{7}$$

where $J \subseteq I(E) = \{ (A, B, x_B) \mid A, B \in E,\ B \subset A,\ x_B \in X_B \}$. Form (6) can be refined similarly.

3.2 Dual Program

Definition 1. *Let $E \subseteq 2^V$, $E' \subseteq 2^V$, $f\colon T(E, X_V) \to \bar{\mathbb{R}}$, $f'\colon T(E', X_V) \to \bar{\mathbb{R}}$. WCSP instances (V, E, X_V, f) and (V, E', X_V, f') are* **equivalent** *iff*

$$\sum_{A \in E} f_A(x_A) = \sum_{A \in E'} f'_A(x_A) \qquad \forall x_V \in X_V$$

Unlike the previous definition of WCSP equivalence [3], [8], [19], Definition 1 does not require that equivalent WCSPs have the same hypergraph, it only requires that they have the same variables V and domains X_V. Thus, it allows to change not only the constraint values but also the hypergraph. We call a transformation taking a WCSP to its equivalent an **equivalent transformation**. Until §9, we will consider only equivalent transformations that preserve the hypergraph (i.e., $E = E'$ in the definition).

The simplest hypergraph-preserving equivalent transformation is applied to a single pair of constraints, f_A and f_B for $B \subset A$, by adding a function $\varphi_{AB}\colon X_B \to \mathbb{R}$ (a 'message') to f_A and subtracting it from[1] f_B, i.e.,

$$f_A(x_A) \leftarrow f_A(x_A) + \varphi_{AB}(x_B) \qquad \forall x_B \in X_B \tag{8a}$$

$$f_B(x_B) \leftarrow f_B(x_B) - \varphi_{AB}(x_B) \qquad \forall x_B \in X_B \tag{8b}$$

Let a function $\varphi_{AB}\colon X_B \to \mathbb{R}$ be assigned to each $(A, B) \in J$. The collection of these functions forms a single mapping φ. Let f^φ denote the WCSP obtained by applying (8) on f for all $(A, B) \in J$, i.e., f^φ is given by

$$f^\varphi_A(x_A) = f_A(x_A) - \sum_{B \mid (B,A) \in J} \varphi_{BA}(x_A) + \sum_{B \mid (A,B) \in J} \varphi_{AB}(x_B) \tag{9}$$

We refer to (9) as a **reparameterization**[2] of f.

In matrix form, (9) reads $f^\varphi = f - M^\top \varphi$. This shows clearly why reparameterizations preserve the primal objective: because $M\mu = 0$ implies $(f^\top - \varphi^\top M)\mu = f^\top \mu$.

Theorem 2. *For any $f\colon T(E, X_V) \to \bar{\mathbb{R}}$, we have*

$$\max_{x_V} \sum_{A \in E} f_A(x_A) \le \sum_{A \in E} \max_{x_A} f_A(x_A) \tag{10}$$

which holds with equality iff there exists a joint state $x_V \in X_V$ such that $f_A(x_A) = \max_{y_A} f_A(y_A)$ for all $A \in E$.

Proof: Clearly, $\max_i \sum_j a_{ij} \le \sum_j \max_i a_{ij}$ for any $a_{ij} \in \bar{\mathbb{R}}$, which holds with equality iff there exists i such that $a_{ij} = \max_k a_{kj}$ for all j. This is applied to (1). □

1. While (8) is clearly an equivalent transformation, it is far from obvious whether *any* hypergraph-preserving equivalent transformation is realizable as a composition of (8) for various $(A, B) \in I(E)$. In analogy with [8, Theorem 3] and [19, Lemma 6.3], we conjecture that this is so if E is closed to intersection ($A, B \in E$ implies $A \cap B \in E$).

2. There is a 'gauge freedom' in (9): $f = f^\varphi$ need not imply $\varphi = 0$. It is an open problem for a given f to describe the set $\{ \varphi \mid f = f^\varphi \}$. If some constraint values are $-\infty$, this seems to be difficult.

The right-hand expression in (10) is an upper bound on (1). By eliminating variables ψ_A, the dual LP reads

$$\min_\varphi \sum_{A \in E} \max_{x_A} f^\varphi_A(x_A) \tag{11}$$

which can be interpreted as *minimizing the upper bound by reparameterizations* permitted by J.

3.3 Hierarchy of LP Relaxations

We have shown, both by primal and dual arguments, that the optimum of the LP (2) is an upper bound on the true WCSP optimum (1). Sometimes, the bound is tight, i.e., equal to (1). For any non-trivially chosen J, this happens for a large and complex class of WCSPs.

Tightness of the relaxation depends on the coupling scheme J. An equality (2b) in the primal corresponds via duality to a variable $\varphi_{AB}(x_B)$ in the dual – thus, the larger J is, the more the primal is constrained and the larger is the set of permitted reparameterizations in the dual. LP relaxations for various $J \in I(E)$ form a *hierarchy*, partially ordered by the inclusion on $I(E)$.

Let $P(E, X_V, J) \subseteq [0, 1]^{T(E,X_V)}$ denote the polytope of mappings μ feasible to the primal LP. The hierarchy of relaxations is established by the obvious implication[3]

$$J_1 \supseteq J_2 \implies P(E, X_V, J_1) \subseteq P(E, X_V, J_2) \tag{12}$$

Imposing marginal consistency in form (7) rather than (2b) would yield a finer-grained hierarchy of relaxations. This would require to modify formula (9).

4 Constraint Satisfaction Problem

The *constraint satisfaction problem* (CSP) [33] is one of the classical NP-complete problems. Here we give background on the CSP which we will need later.

Let each hyperedge $A \in E$ be assigned a crisp constraint $\bar{f}_A\colon X_A \to \{0, 1\}$, understood as the characteristic function of an $|A|$-ary relation over variables A. A joint state x_A is **permitted** (**forbidden**) iff $A \in E$ and $\bar{f}_A(x_A)$ equals 1 (0). Let $\vee$ ($\wedge$) denote the logical disjunction (conjunction). The CSP asks whether there exists a joint state $x_V \in X_V$ satisfying all the relations, i.e., $\bar{f}_A(x_A) = 1$ for each $A \in E$. Such x_V is a **solution**. The CSP instance is defined by $(V, E, X_V, \bar{f})$, where $\bar{f}\colon T(E, X_V) \to \{0, 1\}$.

A CSP is **satisfiable** iff it possesses a solution. We call x_A a **satisfiable joint state** iff it can be extended to a solution. Note, the fact that the CSP is satisfiable and x_A is permitted does not imply that x_A is satisfiable. A joint state x_A is **locally consistent** iff $\bar{f}_B(x_B) = 1$ for every B such that $B \in E$ and $B \subseteq A$. In particular, x_V is a solution iff it is locally consistent.

For tractable subclasses of the CSP see [34], [35].

A powerful tool to solve CSPs is *constraint propagation* [36] (*filtering*, *relaxation labeling* [14]). The possibility to

3. Different coupling schemes may yield the same relaxation, i.e., $P(E, X_V, J_1) = P(E, X_V, J_2)$ need not imply $J_1 = J_2$. It is an open problem to characterize when exactly this happens.

propagate constraints is the distinguishing feature of the CSP: while in general a search cannot be avoided to solve a CSP, propagating constraints during the search prunes the search space such that instances of practical size can be solved. In a way, constraint propagation is a crisp analogy of 'message passing' in graphical models.

In constraint propagation, some obviously unsatisfiable joint states are iteratively deleted using a simple local rule, a *propagator*. This is often done until the CSP satisfies a state characterized by a *local consistency* – then we speak about *enforcing the local consistency*.

Many local consistencies have been proposed, see [36] for a survey and [37] for comparison of their strength for binary CSPs. The most well-known one is *arc consistency* (AC). It is defined for binary CSPs, while we need a local consistency defined for CSPs of any arity. Many such consistencies are known; of them, most relevant to our LP relaxation are *pairwise consistency* (PWC), *generalized arc consistency* (GAC), and *k-consistency* [36].

4.1 J-consistency

To fit our form of coupling, we introduce a modification of PWC, *J-consistency*. While PWC enforces consistency of all pairs of relations, J-consistency enforces consistency of relations $\bar{f}_A$ and $\bar{f}_B$ only if $(A, B) \in J$.

Definition 2. *For $B \subset A$, relations $\bar{f}_A\colon X_A \to \{0,1\}$ and $\bar{f}_B\colon X_B \to \{0,1\}$ are* **pairwise consistent** *iff*

$$\bigvee_{x_{A\setminus B}} \bar{f}_A(x_A) = \bar{f}_B(x_B) \qquad \forall x_B \in X_B \tag{13}$$

*A CSP $(V, E, X_V, \bar{f})$ is J-***consistent** *iff relations $\bar{f}_A$ and $\bar{f}_B$ are pairwise consistent for every $(A, B) \in J$.*

Note that the set of equalities (13) has the following meaning: a joint state x_B is permitted by relation $\bar{f}_B$ iff x_B can be extended to a joint state x_A satisfying $\bar{f}_A$.

PWC and GAC are special cases of J-consistency. PWC is obtained if E is closed to intersection (i.e., $A, B \in E$ implies $A \cap B \in E$) and $J = I(E)$. GAC is obtained[4] if $\binom{V}{1} \subseteq E$ and $J = I_{\text{GAC}}(E)$. For binary CSPs with $\binom{V}{1} \subseteq E$, PWC and GAC become AC.

To enforce J-consistency, a generalization of well-known algorithms to enforce (G)AC can be used. Algorithm 1 deletes unsatisfiable joint states until the CSP becomes J-consistent, while preserving the solution set, i.e., the relation $\bigwedge_{A\in E} \bar{f}_A(x_A)$.

Obviously, if the algorithm makes $\bar{f}$ empty (i.e., $\bar{f} = 0$) then the initial CSP was unsatisfiable. Note that if any relation $\bar{f}_A$ becomes empty during the algorithm, it is already clear that $\bar{f}$ will eventually become empty.

We give the algorithm also in the parameterized form as Algorithm 2, which does not change the relations $\bar{f}$ (thus, they can be represented intensionally, §6.3). Each $(A, B) \in J$ is assigned a function $\bar{\varphi}_{AB}\colon X_B \to \{0,1\}$, where all these functions together form a mapping $\bar{\varphi}$. Initially we set $\bar{\varphi} = 1$. Analogically to (9), we define transformation $\bar{f}^{\bar{\varphi}}$ of $\bar{f}$ by

$$\bar{f}^{\bar{\varphi}}_A(x_A) = \bar{f}_A(x_A) \wedge \bigwedge_{B\mid(B,A)\in J} \bar{\varphi}_{BA}(x_A) \;\wedge \bigwedge_{B\mid(A,B)\in J} \bar{\varphi}_{AB}(x_B)$$

Algorithm 1 (enforcing J-consistency of CSP)

repeat
  Find $(A, B) \in J$, $x_B \in X_B$ s.t. $\bigvee_{x_{A\setminus B}} \bar{f}_A(x_A) \neq \bar{f}_B(x_B)$
  for $x_{A\setminus B} \in X_{A\setminus B}$ **do** $\bar{f}_A(x_A) \leftarrow 0$ **end for**
  $\bar{f}_B(x_B) \leftarrow 0$
until $\bar{f}$ is J-consistent

Algorithm 2 (enforcing J-consistency, parameterized)

repeat
  Find $(A, B) \in J$, $x_B \in X_B$ s.t. $\bigvee_{x_{A\setminus B}} \bar{f}^{\bar{\varphi}}_A(x_A) \neq \bar{f}^{\bar{\varphi}}_B(x_B)$
  $\bar{\varphi}_{AB}(x_B) \leftarrow 0$
until $\bar{f}^{\bar{\varphi}}$ is J-consistent

The *closure* of a CSP with respect to a local consistency is the maximal subset of its permitted joint states that still achieves the local consistency [36, §3]. To formalize this, we define **inclusion** $\leq$ and **join** $\vee$ on CSPs by:

$$\begin{aligned} \bar{f} \leq \bar{f}' &\iff \forall A, x_A\colon \bar{f}_A(x_A) \leq \bar{f}'_A(x_A) \\ \bar{f} = \bar{f}' \vee \bar{f}'' &\iff \forall A, x_A\colon \bar{f}_A(x_A) = \bar{f}'_A(x_A) \vee \bar{f}''_A(x_A) \end{aligned}$$

Definition 3. *The* J**-consistency closure** *of a CSP $(V, E, X_V, \bar{f})$ is the CSP $(V, E, X_V, \bar{f}^*)$ where*

$$\bar{f}^* = \bigvee \{\, \bar{f}' \mid \bar{f}' \leq \bar{f},\;\; \bar{f}' \textit{ is } J\textit{-consistent} \,\} \tag{14}$$

It is easy to verify that the join of J-consistent CSPs is J-consistent (in other words, J-consistent CSPs form a join-semilattice). Hence the closure (14) is J-consistent.

It is not true in general that an algorithm to enforce a local consistency produces the closure of that local consistency [36]. However, it is true for J-consistency.

Theorem 3. *Algorithm 1 or 2 finds the J-consistency closure.*

4.2 k-consistency

There exist stronger local consistencies than (G)AC and PWC. Most important of them is *k-consistency* [36].

Definition 4. *A CSP is k-***consistent** *iff for every locally consistent joint state x_A such that $|A| = k - 1$ and every variable $v \in V$ there exists a state x_v such that $x_{A\cup(v)}$ is locally consistent (i.e., x_A can be extended to variable v).*

Strong k-consistency is k'-consistency for all $k' \leq k$. Solvability by strong k-consistency characterizes an important class of tractable relation languages [34, §8.4.2] (e.g., binary CSPs with Boolean variables are solved by

4. We remark that the concept of GAC allows us to explain Theorem 1 in §3.1 in CSP terms as follows. An integer primal-feasible μ can be seen as a CSP in which, due to (2c), each relation has a single permitted joint state. Clearly, such a CSP is satisfiable iff it is GAC.

strong 3-consistency). Strong k-consistency solves CSPs with (hyper)graph of treewidth less than k [38], [35, §3.4].

Unlike J-consistency, enforcing (strong) k-consistency in general requires adding new relations to the CSP. Let $\bar{f}_A = 1$ denote the *universal relation* (i.e., identically true) on A. An inefficient way to enforce k-consistency is to add all possible universal relations of arity $k-1$ and k (such that $\binom{V}{k-1} \cup \binom{V}{k} \subseteq E$), then enforce PWC, and then remove all the previously added k-ary relations [39, §8].

In a more efficient algorithm, only *some* of the missing $(k-1)$-ary relations can be added. It achieves strong k-consistency by enforcing k'-consistency for $k' = 2, \dots, k$ in turn. A $(k-1)$-consistent CSP is made k-consistent as follows. We iteratively set $\bar{f}_A(x_A) \leftarrow 0$ whenever $|A| = k-1$ and x_A cannot be extended to some variable v. Here, if A was not already in E, we first add A to E and set $\bar{f}_A \leftarrow 1$. Simultaneously, PWC is enforced.

2-consistency is the same as arc consistency.

4.2.1 3-consistency and Path Consistency

3-consistency in a binary CSP is also known as *path consistency* for the following reason. A sequence $(u, \dots, v)$ of variables is a *path* if $\{u, v\}$ and all edges along the sequence are in E (we also allow $u = v$ which yields a cycle). The path is *consistent* iff any state pair (x_u, x_u) satisfying relation $\bar{f}_{uv}$ can be extended to all intermediate relations along the path. A graph is chordal (= triangulated) iff every cycle of length 4 or more has a chord.

Theorem 4. *In a chordal graph, every path of length 3 (i.e., with 3 variables) is consistent iff every path is consistent.*

Proof: For complete (hence chordal) graphs, this is a classical results by Montanari [13], [36]. It was extended to chordal graphs by Bliek and Sam-Haroud [40]. □

By definition, 3-consistency means that any locally consistent state pair (x_u, x_v) can be extended to any third variable w. In other words, after filling-in the CSP to the complete graph with universal binary relations, all paths of length 3 are consistent – hence, all paths are consistent.

4.3 CSP with a Relation over All Variables

Let $\bigcup E = \bigcup_{A \in E} A$ (typically but not necessarily we have $\bigcup E = V$). Consider a CSP containing a relation over hyperedge $\bigcup E$ (i.e., $\bigcup E \in E$) and the coupling scheme

$$I_{\mathrm{SAT}}(E) = \{ (\textstyle\bigcup E, A) \mid A \in E,\ A \neq \bigcup E \} \tag{15}$$

which couples $\bigcup E$ to all other hyperedges. Propositions 5, 6, 7 give properties of $I_{\mathrm{SAT}}(E)$-consistency we will need later. Proofs are easy, from Definitions 2, 3.

Proposition 5. *A CSP with $\bigcup E \in E$ has a non-empty $I_{\mathrm{SAT}}(E)$-consistency closure iff it is satisfiable.*

Proposition 6. *A CSP with $\bigcup E \in E$ is $I_{\mathrm{SAT}}(E)$-consistent iff every joint state x_A permitted by $\bar{f}_A$ is satisfiable.*

Proposition 7. *If a CSP with $\bigcup E = V \in E$ is $I_{\mathrm{SAT}}(E)$-consistent then*

$$\bar{f}_V(x_V) \leq \bigwedge_{A \in E} \bar{f}_A(x_A) \qquad \forall x_V \in X_V \tag{16}$$

Note, equality in (16) for all x_V means that the relation $\bar{f}_V$ is realizable as the conjunction of the relations $\bar{f}_A$.

Example 4. Let $V = (1, 2, 3)$, $X_V = \{0, 1\}^V$, and $E = \{ (1, 2), (1, 3), (2, 3), (1, 2, 3) \}$. Let $\bar{f}$ be defined by

$$\bar{f}_{123}(x_1, x_2, x_3) = x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \tag{17a}$$
$$\bar{f}_{12}(x_1, x_2) = \bar{x}_1 \bar{x}_2 \vee x_1 \bar{x}_2 \vee \bar{x}_1 x_2 \tag{17b}$$

and $\bar{f}_{13} = \bar{f}_{23} = \bar{f}_{12}$, where we denoted $\bar{x}_u = 1 - x_u$ and $x_u x_v = x_u \wedge x_v$. The CSP $(V, E, X_V, \bar{f})$ is $I_{\mathrm{SAT}}(E)$-consistent but the inequality in (16) is strict for $x_V = (x_1, x_2, x_3) = (0, 0, 0)$. Clearly, $\bar{f}_{123}$ is not realizable as a conjunction of *any* binary relations. □

5 Optimality of the LP Relaxation

Given feasible primal and dual variables, we want to recognize whether they are optimal to the LP pair and whether this optimum is tight. As shown in [3], [8] for binary WCSPs, the answers to these questions depend only on the properties of the *CSP formed by the active joint states*. This is significant because it moves reasoning about optimality of the LP relaxation to the realm of a well-known and long-studied problem.

Here we extend these results for WCSPs of any arity. Theorems 8, 9, 10 characterize three levels of optimality of the LP relaxation: whether the upper bound is tight (i.e., equal to (1)), minimal, or locally minimal.

Definition 5. *Given a function $f_A \colon X_A \to \bar{\mathbb{R}}$, we define a relation $\lceil f_A \rceil \colon X_A \to \{0, 1\}$ by*

$$\lceil f_A \rceil(x_A) = \begin{cases} 1 & \text{if } f_A(x_A) = \max_{y_A} f_A(y_A) \\ 0 & \text{if } f_A(x_A) < \max_{y_A} f_A(y_A) \end{cases} \tag{18}$$

A joint state x_A of constraint f_A is **active** *iff $\lceil f_A \rceil(x_A) = 1$.*

Given a mapping $f \colon T(E, X_V) \to \bar{\mathbb{R}}$, we define a mapping $\lceil f \rceil \colon T(E, X_V) \to \{0, 1\}$ by $\lceil f \rceil_A(x_A) = \lceil f_A \rceil(x_A)$.

Theorem 8. *Inequality (10) holds with equality iff the CSP $(V, E, X_V, \lceil f \rceil)$ is satisfiable*[5]*. The solutions of this CSP are in one-to-one correspondence with the maximizers of (1).*

Proof: By restating the second part of Theorem 2. □

Theorem 9. *Let $\mu \colon T(E, X_V) \to [0, 1]$ be feasible to the primal LP. The primal and dual LP are jointly optimal iff*

$$[1 - \lceil f_A^{\varphi} \rceil(x_A)]\, \mu_A(x_A) = 0 \qquad \forall A \in E,\ x_A \in X_A \tag{19}$$

Proof: Apply complementary slackness to (2b). □

Theorem 9 characterizes WCSPs for which the bound is dual optimal, i.e., cannot be improved by changing φ:

5. Note a subtlety: since finding a solution to a CSP is NP-complete even if we know that the CSP is satisfiable, finding an optimizer to a WCSP is NP-complete even if we know that the upper bound is tight.

it is when a primal feasible μ exists such that $\mu_A(x_A) = 0$ whenever joint state x_A is inactive. In fact, given any single dual optimal solution, all primal optimal solutions are uniquely determined by the active joint states by (19).

Theorem 10. *If, for any J, φ is optimal to the dual LP then the J-consistency closure of $\lceil f^\varphi \rceil$ is not empty.*

Proof: Let $\bar{f}'_A(x_A) = [\![\mu_A(x_A) > 0]\!]$. For any non-negative μ_A, obviously $\sum_{x_{A\setminus B}} \mu_A(x_A) = \mu_B(x_B)$ implies $\bigvee_{x_{A\setminus B}} \bar{f}'_A(x_A) = \bar{f}'_B(x_B)$. Hence, the CSP $\bar{f}'$ is J-consistent. Clearly, (19) can be rewritten as $\bar{f}' \leq \lceil f^\varphi \rceil$. By (14), $\lceil f^\varphi \rceil$ has a non-empty J-consistency closure. □

As shown in [41], [8], [19], a non-empty closure of $\lceil f^\varphi \rceil$ is only necessary but not sufficient for dual optimality. Thus, Theorem 10 characterizes *local minima* of the upper bound (10). These local minima naturally appear in several algorithms to solve the dual LP[6].

The only known WCSP classes for which all local minima are global are the supermodular ones and those with binary constraints and Boolean variables [8], [42].

6 Optimizing the Bound

Here we focus on algorithms to solve the LP (2).

It is better to solve the dual LP than the primal LP. This is because no algorithm is known to solve (or find a good suboptimum of) the primal for large instances; only general LP solvers (simplex) have been used [25], [26]. Moreover, the number of primal variables is exponential in the arity of the instance, thus for large arities the primal cannot be solved explicitly at all, whereas the corresponding exponential number of dual constraints sometimes can be handled implicitly (§6.3, §9.3).

6.1 Existing Algorithms for Binary Problems

The dual LP in the form (11) is an unconstrained minimization of a convex piecewise linear (hence nonsmooth) function. To scale to large instances, it is reasonable to require that an algorithm to solve (11) have space complexity *linear in the number of dual variables* $\varphi_{AB}(x_B)$. This rules out e.g. the simplex and interior point algorithms. For binary WCSPs, known algorithms with this property can be divided into two groups:

1) Local algorithms find a local minimum of the upper bound characterized by arc consistency of the active joint states. The found local minima are usually very good or even global. Two types of such algorithms are known:

a) Algorithms based on averaging max-marginals: *max-sum diffusion* [4], [5], [8], *TRW-S* [19] and [43], [20]. They can be roughly seen as a (block) coordinate descent. Existence of local minima follows from the fact that coordinate descent need not find the global minimum of a convex nonsmooth function [44, §7.3].

b) The *augmenting DAG* algorithm [6], [7], [8] and the *virtual arc consistency (VAC)* algorithm [12]. They explicitly try to enforce AC of $\lceil f^\varphi \rceil$. If all the states of any variable are deleted, the bound can be improved by back-tracking the causes of deletions.

2) Global algorithms find the global minimum of the upper bound. Two types of such algorithms are known:

a) Subgradient descent [45], [21] is a well-known method to minimize nonsmooth functions. These approaches rely on decomposing the WCSP as a sum of tractable subproblems (§1.1). To achieve good convergence rate, the subproblems must be well chosen (large enough).

b) Smoothing algorithms [44, §7.4], [23], [20], [46] use a sequence of smooth convex approximations of our nonsmooth convex objective function. Each such function can be minimized by coordinate descent globally.

Unlike the global algorithms, the local algorithms improve the bound monotonically.

In principle, any of the above algorithms can be generalized to n-ary WCSPs, still keeping its space complexity linear in the number of variables $\varphi_{AB}(x_B)$. This is easiest for max-sum diffusion, which we show in §6.2.

6.2 Max-sum Diffusion

The max-sum diffusion iteration is the reparameterization (8) on a single $(A, B) \in J$ that averages f_B and the max-marginals of f_A, i.e., makes satisfied the equalities

$$\max_{x_{A\setminus B}} f_A(x_A) = f_B(x_B) \qquad \forall x_B \in X_B \tag{20}$$

If $f_B(x_B) > -\infty$, $\max_{x_{A\setminus B}} f_A(x_A) > -\infty$, this is done by setting $\varphi_{AB}(x_B) = [f_B(x_B) - \max_{x_{A\setminus B}} f_A(x_A)]/2$ in (8).

Theorem 11. *The iteration does not increase the upper bound.*

Proof: Let us denote $a(x_B) = \max_{x_{A\setminus B}} f_A(x_A)$, $b(x_B) = f_B(x_B)$, $c(x_B) = [b(x_B) - a(x_B)]/2 = \varphi_{AB}(x_B)$. Before the iteration, the contribution of f_A and f_B to the upper bound (10) is

$$\max_{x_A} f_A(x_A) + \max_{x_B} f_B(x_B) = \max_{x_B} a(x_B) + \max_{x_B} b(x_B) \tag{21}$$

After the iteration, this contribution is

$$\begin{aligned} &\max_{x_B}[a(x_B) + c(x_B)] + \max_{x_B}[b(x_B) - c(x_B)] \\ &= \max_{x_B}[a(x_B) + b(x_B)] \end{aligned} \tag{22}$$

Clearly, expression (22) is not greater than (21). □

Using parameterization (9), we obtain Algorithm 3. To correctly handle infinite weights, it assumes that the CSP $\bar{f}^{\rm fin}$ defined by $\bar{f}^{\rm fin}_A(x_A) = [\![f_A(x_A) > -\infty]\!]$ is J-consistent. Optionally, any time a constant can be added to a constraint and subtracted from another constraint.

Next we give important properties of the algorithm.

Theorem 12. *In any fixed point φ of Algorithm 3, $\lceil f^\varphi \rceil$ is J-consistent.*

Proof: Show that $\max_{x_{A\setminus B}} f_A(x_A) = f_B(x_B)$ implies $\bigvee_{x_{A\setminus B}} \lceil f_A \rceil(x_A) = \lceil f_B \rceil(x_B)$, which is easy. □

6. In particular, in any fixed point of the TRW-S algorithm [19], the states and state pairs whose max-marginals are maximal in trees form an arc consistent CSP. This is called *weak tree agreement* in [19].

Algorithm 3 (max-sum diffusion, parameterized)

loop
  for $(A, B) \in J$, $x_B \in X_B$ s.t. $f_B(x_B) > -\infty$ **do**
    $\varphi_{AB}(x_B) \leftarrow \varphi_{AB}(x_B) + [f_B^\varphi(x_B) - \max_{x_{A\setminus B}} f_A^\varphi(x_A)]/2$
  end for
end loop

Theorem 13. *If the J-consistency closure of $\lceil f^\varphi \rceil$ is initially empty then after a finite number of iterations of Algorithm 3, the upper bound strictly decreases.*

Theorem 14. *If the J-consistency closure of $\lceil f^\varphi \rceil$ is initially non-empty then:*

- *after any number of iterations of Algorithm 3, the upper bound does not change;*
- *after a finite number of iterations of Algorithm 3, $\lceil f^\varphi \rceil$ becomes the J-consistency closure of the initial $\lceil f^\varphi \rceil$.*

Proof: Theorems 13 and 14 can be proved by noting that what diffusion does to the active joint states is precisely what Algorithm 2 does to the permitted joint states. See [8, Theorem 7], cf. [19, Theorem 3.4]. □

Max-sum diffusion is not yet fully understood, in particular its convergence theory is missing. For binary WCSPs, it has been conjectured [4], [5] that diffusion converges to a fixed point, when (20) holds for all $(A, B) \in J$. Though firmly believed true, this conjecture has been never proved. We state it as follows.

Conjecture 15. *In Algorithm 3, the sequence of numbers $f_B^\varphi(x_B) - \max_{x_{A\setminus B}} f_A^\varphi(x_A)$ converges to zero.*

6.3 Handling High-arity and Global Constraints

A constraint f_A can be represented either by explicitly storing the values $f_A(x_A)$ for all $x_A \in X_A$ or by a black-box function. In constraint programming, this is known as *extensional* and *intensional* representation, respectively. For high-arity constraints, only intensional representation is possible because the set X_A is intractably large. Intensionally represented constraints of a non-fixed arity (not necessarily depending on all the variables) are referred to as *global constraints* [47], [36], [48].

The propagator of a local consistency that is trivial to execute for a low-arity constraint may be intractable for a high-arity intensional constraint. A lot of research has been done to find polynomial-time propagators for global constraints [47]. Usually, the strongest local consistency for which such a propagator is found is GAC.

Analogically, max-sum diffusion can handle an intensionally represented constraint f_A of an arbitrarily high arity if a polynomial algorithm exists to compute $\max_{x_{A\setminus B}} f_A^\varphi(x_A)$. Recall from §4 that the iteration of Algorithm 1 or 2 is the propagator for J-consistency. In this sense, the max-sum diffusion iteration can be called a *soft propagator* (for the augmenting DAG / VAC algorithm, we would need a slightly different soft propagator). Thus, soft high-arity and global constraints can be handled in the way similar to how crisp global constraints are commonly handled in CSP.

Example 5. Let $E = \binom{V}{1} \cup E' \cup (V)$ where $E' \subseteq \binom{V}{2}$. Let $J = I_{\text{GAC}}(E)$. Algorithm 3 does two kinds of updates: between binary and unary constraints, and between the global and unary constraints. For the latter, we need to compute $\max_{x_{V\setminus(u)}} f_V^\varphi(x_V)$ for every $u \in V$ and $x_u \in X_u$. From (9) we have

$$f_V^\varphi(x_V) = f_V(x_V) + \sum_{v\in V} \varphi_{Vv}(x_v) \tag{23}$$

Note that (23) is an objective function of a WCSP with the global and unary constraints. Depending on f_V, computing $\max_{x_{V\setminus(u)}} f_V^\varphi(x_V)$ may or may not be tractable.

As a tractable example, let $X_V = \{0, 1\}^V$ and

$$f_V(x_V) = \begin{cases} 0 & \text{if } \sum_{v\in V} x_v = n \\ -\infty & \text{otherwise} \end{cases} \tag{24}$$

be the *cardinality constraint*[7], which enforces the number of variables with state 1 to be n. Instead of the max-marginal $\max_{x_{V\setminus(u)}} f_V^\varphi(x_V)$, for simplicity we will only show how to compute $\max_{x_V} f_V^\varphi(x_V)$. It can be rewritten as a constrained maximization,

$$\max_{x_V} f_V^\varphi(x_V) = \max\Big\{ \sum_{v\in V} \varphi_{Vv}(x_v) \;\Big|\; x_V \in X_V,\ \sum_{v\in V} x_v = n \Big\}$$

One verifies that this equals $\beta + \sum_{v\in V} \varphi_{Vv}(0)$ where β is the sum of n greatest numbers from $\{\, \varphi_{Vv}(1) - \varphi_{Vv}(0) \mid v \in V \,\}$ [49]. This can be done efficiently using a dynamically updated sorted list.

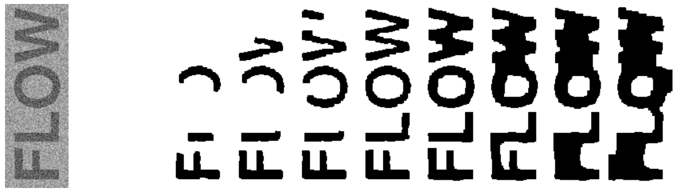
As an evidence that the approach yields plausible approximations, we present a toy experiment with image segmentation. The first image in Figure 1 is the input binary image corrupted with additive Gaussian noise. We set $f_{uv}(x_u, x_v) = [\![x_u = x_v]\!]$ and $f_v(x_v) = -[\theta(x_v) - g_v]^2$ where $\theta(x)$ is the expected intensity of a pixel with label x and g_v is the actual intensity of pixel v. We ran diffusion until the greatest residual was 10^{-8} and then we obtained x_V by taking the active state in each variable (this means, the constraint $\sum_v x_v = n$ may be satisfied only approximately). The binary images in Figure 1 show the results for different n. □

Note that e.g. all algorithms in [49] can be used as soft GAC-propagators. We anticipate that in the future, more global constraints with tractable soft propagators and useful in applications will be discovered.

7 Supermodular Problems

Supermodular constraints form the only known interesting tractable class of weighted constraints languages [11]. For binary supermodular WCSPs, it is known that

7. Binary supermodular WCSPs with cardinality constraint (24) (and its soft versions) can be well approximated by a more efficient algorithm using parametric max-flow. In detail, we observed experimentally (but did not prove) that constraining the variables $\varphi_{Vv}(x_v)$ to be equal for all $v \in V$ does not change the least upper bound. However, this of course may not hold for other global constraints.



$n =$	2000	3000	4000	5000	6000	7000	8000	9000
$\sum_v x_v =$	2008	3004	4011	5006	6004	7024	7982	9032

Fig. 1. Image segmentation with cardinality constraint.

the LP relaxation [3] is tight [50], [8]. This has been generalized to n-ary supermodular WCSPs by Werner [9] and Cooper et al. [12]. Moreover, [9], [12] show that to solve a supermodular WCSP it suffices to find any *local* optimum of the bound such that $\lceil f^\varphi \rceil$ is (G)AC.

D. Schlesinger [15] showed that binary supermodular WCSPs can be solved in polynomial time even after an unknown permutation of states in each variable. As pointed out in [15], [12], this can be done also for n-ary supermodular WCSPs.

Revisiting [50], [8], [9], [12], [15], we show in this section how to solve permuted n-ary supermodular WCSPs.

Let each domain X_v be endowed with a total order $\leq_v$. A function f_A: $X_A \to \bar{\mathbb{R}}$ is **supermodular** iff

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

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

Suppose that max-sum diffusion (or the augmenting DAG / VAC algorithm, §6.1) with $J = I_{\mathrm{GAC}}(E)$ found φ such that $\lceil f^\varphi \rceil$ is GAC. It can be verified that supermodularity of f_A is preserved by reparameterizations (8) on pairs[8] $(A,(v))$, hence constraints f_A^φ are supermodular too. It remains to prove the following theorem.

Theorem 16. *Let a WCSP be such that its constraints are supermodular and the CSP formed by its active joint states is GAC. Then (10) holds with equality and a maximizer of (1) can be found in polynomial time, without taking into account the orders $\leq_v$ of variable states.*

Proof: A relation $\bar{f}_A$: $X_A \to \{0,1\}$ is a **lattice** iff

$$\bar{f}_A(x_A \wedge y_A) \wedge \bar{f}_A(x_A \vee y_A) \geq \bar{f}_A(x_A) \wedge \bar{f}_A(y_A)$$

for any $x_A, y_A \in X_A$, i.e., iff $\bar{f}_A(x_A) = \bar{f}_A(y_A) = 1$ implies $\bar{f}_A(x_A \wedge y_A) = \bar{f}_A(x_A \vee y_A) = 1$. A CSP in which each relation is a lattice is a **lattice CSP**. The lattice CSP is both max-closed and min-closed [34, §8.4.2], hence tractable. Its instance is satisfiable iff its GAC closure is not empty.

The maximizers of a supermodular function on a distributive lattice form a sublattice of this lattice [51]. Hence, $\lceil f^\varphi \rceil$ is a lattice CSP. Because $\lceil f^\varphi \rceil$ is GAC and non-empty, it is satisfiable and its solutions are in one-to-one correspondence with the maximizers of (1).

We will show how to find a solution to a lattice CSP that is GAC and non-empty. If the orders $\leq_v$ are known, a solution x_V is formed simply by the lowest (w.r.t. $\leq_v$) permitted state x_v in each variable [34], [50], [8].

If the orders $\leq_v$ are unknown, we give an algorithm to find a solution independently on them. It is easy to prove that the GAC closure of a lattice CSP is again a lattice CSP. Let us pick any (v, x_v) and set $\bar{f}_v(x_v) \leftarrow 0$. This can be seen as adding a unary relation to the CSP. Since any unary relation is trivially a lattice, if we now enforce GAC we again obtain a lattice CSP. This CSP is either empty or non-empty. If it is empty (i.e., x_v was the last satisfiable state in variable v), we undo the enforcing of GAC and pick a different (v, x_v). If it is non-empty, we have a lattice CSP with fewer permitted states that is non-empty and GAC, hence satisfiable. We can pick another (v, x_v) and repeat the iteration, until each variable has a single permitted state. □

8 Incrementally Tightening Relaxation

We have seen in §3.3 that choosing different coupling schemes $J \subseteq I(E)$ yields a hierarchy of LP relaxations. Here we show that the relaxation can be tightened *incrementally*, by progressively enlarging J.

Any time during max-sum diffusion, we can extend the current J by any $J' \subseteq I(E)$ (i.e., we set $J \leftarrow J \cup J'$). This means, we add dual variables φ_{AB} for $(A,B) \in J'$ and set them to zero. Clearly, this does not change the current upper bound. By Theorem 11, the future diffusion iterations either preserve or improve the bound. If the bound does not improve, all we have lost is the memory occupied by the added dual variables.

Alternatively, this can be imagined as if the dual variables φ_{AB} were initially present for all $(A,B) \in I(E)$ but were 'locked' to zero except for those given by J. Extending J 'unlocks' some dual variables.

This scheme can be run also with other bound-optimizing algorithms (§6.1). If the algorithm is monotonic, the resulting incremental scheme is monotonic too.

The incremental scheme can be seen as a *cutting plane algorithm* because an extension of J that leads to a better bound corresponds to adding linear inequalities that separate the solution μ optimal in the current feasible polytope $P(E, X_V, J)$ from the polytope $P(E, X_V, I(E))$. Finding cutting plane(s) that separate current μ from $P(E, X_V, I(E))$ is known as the *separation problem*[9].

Note that our algorithm runs in the dual rather than primal space and that many rather than one cutting plane are added at a time: extending J by already a single $(A,B) \in I(E)$ may result in several planes intersecting $P(E, X_V, J)$, induced by the primal constraints.

8. I thank Martin Cooper for pointing out that transformation (8) preserves supermodularity only if $|B| = 1$. It is not clear whether diffusion solves supermodular WCSPs if $J \supset I_{\mathrm{GAC}}(E)$.

9. The term *separation problem* is not fully justified here because it will be applied also to the case when extending J gets the bound out of a *local* optimum (see Theorem 10), as shown later in Example 11.

8.1 Separation Test

Let us ask whether adding a given $J' \subseteq I(E)$ to current J would lead a bound improvement. We refer to this as the *separation test*. Of course, this test must be simpler and provide more insight than actually adding J' and running the bound-optimizing algorithm.

One easily invents a *sufficient* separation test: if running diffusion such that only pairs $(A, B) \in J'$ are visited improves the bound (where the amount of improvement is a good heuristic to assess the usefulness of J' [29]) then running diffusion on pairs $(A, B) \in J \cup J'$ would obviously improve the bound too. Unfortunately, this test is not *necessary*, by Example 11 given later in §9.5.

Theorems 13+14 yield a sufficient and necessary test:

Proposition 17. *Extending J by J' leads to a bound improvement iff the $(J \cup J')$-consistency closure of $\lceil f^\varphi \rceil$ is empty.*

By Proposition 17, to find out whether extending J and running Algorithm 3 improves the bound, we can extend J and run (simpler and faster) Algorithm 2.

9 Adding Zero Constraints

So far, we have considered only equivalent transformations that preserve the hypergraph (i.e., $E = E'$ in Definition 1). Let us turn to equivalent transformations that change the hypergraph. The simplest such transformation is obtained by *adding a zero constraint*, i.e., by adding a hyperedge $A \notin E$ to E, setting $f_A = 0$ (where 0 denotes the zero function), and extending J to couple A to (some or all of) the existing incident hyperedges. More complex such transformations are obtained as the composition of reparameterizations and adding zero constraints. Since adding zero constraints enables previously impossible reparameterizations, it may improve the relaxation.

All the results obtained in §3–§8 of course apply also to zero constraints. Further in §9 we discuss some specific properties of WCSPs containing zero constraints.

9.1 Complete Hierarchy of LP Relaxations

Given a WCSP with hypergraph E, its hypergraph can be completed to the complete hypergraph 2^V by adding zero constraints with scopes $2^V \setminus E$. Now, the relaxation is determined by the coupling scheme $J \subseteq I(2^V)$ alone. The zero constraints not present in any pair $(A, B) \in J$ are only virtual, they have no effect and can be ignored.

As in §3.3, relaxations for various $J \subseteq I(2^V)$ form a hierarchy, partially ordered by inclusion on $I(2^V)$. For the lowest element of the hierarchy, $J = \emptyset$, formula (9) permits no reparameterizations at all and the optimum of the LP is simply the upper bound (10). The highest element of the hierarchy, $J = I(2^V)$, yields the exact solution; however, by Proposition 5 the exact solution is obtained already for $J = I_{\mathrm{SAT}}(E)$. In between, there is a range of intermediate relaxations, including $J = I(E)$.

9.2 Adding Zero Constraints = Lifting + Projection

Let zero constraints with scopes $F \subseteq 2^V \setminus E$ be added to a WCSP (V, E, X_V, f) and let $J \subseteq I(E \cup F)$. Since zero constraints do not affect the objective function of the primal LP, the primal LP can be written as

$$\max \{ f^\top \mu \mid \mu \in \pi_{T(E,X_V)} P(E \cup F, X_V, J) \} \tag{25}$$

where $\pi_{D'} Y \subseteq \mathbb{R}^{D'}$ denotes the **projection** of a set $Y \subseteq \mathbb{R}^D$ onto dimensions $D' \subseteq D$ (i.e., $\pi_{D'}$ deletes components $D \setminus D'$ of every element of Y). Thus, zero constraints manifest themselves as a projection of the primal feasible polytope onto the space of non-zero constraints. In turn, adding zero constraints with scopes F then means *lifting* the primal feasible set from dimensions $T(E, X_V)$ to dimensions $T(E \cup F, X_V)$, imposing new primal constraints (2b)+(2c)+(2d) in the lifted space, and projecting back onto dimensions $T(E, X_V)$.

Suppose zero constraints with scopes $2^V \setminus E$ have been added. Similarly to (12), for any $J_1, J_2 \subseteq I(2^V)$ we have[10]

$$\begin{aligned} & J_1 \supseteq J_2 \implies \\ & \pi_{T(E,X_V)} P(2^V, X_V, J_1) \subseteq \pi_{T(E,X_V)} P(2^V, X_V, J_2) \end{aligned}$$

In [16], [17], [18], Wainwright et al. introduced the *marginal polytope*, formed by collections (associated with E and X_V) of marginals of some global distribution μ_V. Of fundamental importance is the marginal polytope of the complete hypergraph 2^V, given by $P(2^V, X_V, I(2^V))$. The marginal polytope of a hypergraph $E \subseteq 2^V$ is then $\pi_{T(E,X_V)} P(2^V, X_V, I(2^V))$. Therefore, for any $J \subseteq I(2^V)$, polytope $\pi_{T(E,X_V)} P(2^V, X_V, J)$ is a polyhedral *outer bound* of the marginal polytope associated with (E, X_V).

It is not hard to show [9] that the marginal polytope is the *WCSP integral hull*, i.e., the convex hull of (integral) points feasible to the integer LP given by Theorem 1.

9.3 Handling Zero Constraints in Max-sum Diffusion

In the sense of §6.3, a zero constraint can be understood as a trivial intensionally represented constraint and the max-sum diffusion iteration as its soft propagator. Let us see how zero constraints can be handled in diffusion.

Suppose $f_A = 0$. Then reparameterization (9) reads

$$f_A^\varphi(x_A) = -\sum_{B|(B,A)\in J} \varphi_{BA}(x_A) + \sum_{B|(A,B)\in J} \varphi_{AB}(x_B) \tag{26}$$

Example 6. Let $E = \{(2), (1, 2), (2, 3)\}$, $f_2 = 0$, $J = I(E)$. Then $f_2^\varphi(x_2) = -\varphi_{12,2}(x_2) - \varphi_{23,2}(x_2)$. □

More interesting is the case when there is no B such that $(B, A) \in J$. Then the first sum in (26) is vacuous and f_A^φ is the objective function of a WCSP with variables A, hypergraph $E_A = \{ B \mid (A, B) \in J \}$, and constraints φ_{AB}. Computing $\max_{x_{A \setminus B}} f_A^\varphi(x_A)$ means *solving a WCSP*

10. Recall (Footnote 3) that $J_1 \neq J_2$ may yield the same relaxation. If all the constraints are non-zero, this happens iff $P(2^V, X_V, J_1) = P(2^V, X_V, J_2)$. If constraints with scopes $2^V \setminus E$ are zero, this happens iff $\pi_{T(E,X_V)} P(2^V, X_V, J_1) = \pi_{T(E,X_V)} P(2^V, X_V, J_2)$. Note that the former condition implies the latter one but not *vice versa*.

on a smaller hypergraph[11]. Adding a zero constraint f_A makes sense only if WCSPs on E_A are easier to solve than the WCSP on E. Note that no function of arity $|A|$ needed to be explicitly stored.

Example 7. Let $V = (1,2,3,4)$, $E = \{\,(1), (2), (3), (4), (1,2), (2,3), (3,4), (1,4), (1,3), V\,\}$, $f_V = 0$, and $J = \{\,(V,(1,2)), (V,(2,3)), (V,(3,4)), (V,(1,4))\,\}$. Then E_A is a cycle of length 4 and $f_V^\varphi(x_1,x_2,x_3,x_4) = \varphi_{V,12}(x_1,x_2) + \varphi_{V,23}(x_2,x_3) + \varphi_{V,34}(x_3,x_4) + \varphi_{V,14}(x_1,x_4)$. □

Example 8. In this example, we show how adding short cycles improves relaxations of binary WCSPs.

We tested two types of graphs:

- $E = \binom{V}{1} \cup E'$ where $E' \subseteq \binom{V}{2}$ is the 2-dimensional 4-connected $m \times m$ grid.
- Complete graph, $E = \binom{V}{1} \cup \binom{V}{2}$ where $|V| = m$.

For the grid graph, we tested two relaxations: $J_1 = I(E)$ and $J_2 = I(E \cup F)$ where $F \subseteq \binom{V}{4}$ contains all hyperedges A such that $E \cap 2^A$ is a cycle of length 4 (as in Example 7). For the complete graph, we tested two relaxations: $J_1 = I(E)$ and $J_2 = I\big(E \cup \binom{V}{3}\big)$, i.e., the relaxation J_2 was obtained by adding all 3-cycles.

Each variable had the same number of states, $|X_v|$. We tested five types of constraints f:

- `random`: all weights $f_v(x_v)$ and $f_{uv}(x_u,x_v)$ were i.i.d. drawn from the normal distribution $\mathcal{N}[0;1]$.
- `attract`: $f_{uv}(x_u,x_v) = [\![x_u = x_v]\!]$ and $f_v(x_v)$ were drawn from $\mathcal{N}[0;1.6]$. We chose variance 1.6 because it yielded (by trial) the hardest instances.
- `repulse`: $f_{uv}(x_u,x_v) = [\![x_u \neq x_v]\!]$ and $f_v(x_v)$ were drawn from $\mathcal{N}[0;0.1]$.

In the other types (apply only to grid graphs), $f_v(x_v)$ were drawn from $\mathcal{N}[0;1]$ and the binary constraints were crisp, $f_{uv}(x_u,x_v) \in \{-\infty, 0\}$. They were taken from [7]:

- `lines`: $f_{uv}(x_u,x_v)$ were as in [7, Figure 19a].
- `curve`: $f_{uv}(x_u,x_v)$ were as in [7, Figure 15a].

On a number of WCSP instances, we counted how many instances were solved to optimality. Once diffusion converged, the instance was marked as solved if there was a unique active state in each variable. Table 1 shows the results, where r_1 resp. r_2 is the proportion of instances solved to optimality by relaxation J_1 resp. J_2. There were 100 trials for each line; in each trial, we randomly drew instances from the instance type and computed relaxation J_1 until it was not tight, and then we computed relaxation J_2. Runtime for `random` or `attract` on 100×100 grid and $|X_v| = 4$ was several minutes (for a non-optimized Matlab+C code).

For `random` and `attract` on both graphs and for `repulse` on grids, relaxation J_2 was much tighter and was often exact even for large graphs. For crisp binary constraints (on grids), relaxation J_2 clearly beat J_1, but for $m \geq 25$ `lines` and `curve` were unsolvable. This is not too surprising because `lines` and `curve` are much harder than instances typical in low-level vision (such as the benchmarks in [52]). In more detail [7], they are easy if the data terms $f_v(x_v)$ are 'close to a feasible image' but this is not at all the case if $f_v(x_v)$ are random.

Despite it is known that densely connected instances are hard [53], it is surprising that `repulse` was never solved even on very small graphs. Note, `repulse` encourages neighboring variables to have different states, thus it is close to the difficult graph coloring problem. □

graph	constraints	m	$\lvert X_v \rvert$	r_1	r_2
grid	`random`	15	5	0.01	1.00
grid	`random`	25	3	0.00	0.98
grid	`random`	100	3	0.00	0.72
grid	`attract`	15	5	0.79	0.99
grid	`attract`	25	5	0.48	0.98
grid	`attract`	100	5	0.00	0.81
grid	`repulse`	10	3	0.18	1.00
grid	`repulse`	20	3	0.00	0.98
grid	`repulse`	50	3	0.00	0.57
grid	`lines`	10	4	0.71	0.85
grid	`lines`	15	4	0.40	0.54
grid	`lines`	25	4	0.00	0.05
grid	`curve`	10	9	0.17	0.65
grid	`curve`	15	9	0.00	0.24
grid	`curve`	25	9	0.00	0.00
complete	`random`	10	3	0.01	1.00
complete	`random`	15	3	0.00	0.89
complete	`random`	20	3	0.00	0.40
complete	`random`	25	2	0.00	0.87
complete	`repulse`	4	2	0.00	0.98
complete	`repulse`	5	2	0.00	0.00
complete	`repulse`	4	3	0.00	0.00

TABLE 1
Tightening the LP relaxation by adding short cycles.

9.4 Optimality under Presence of Zero Constraints

As shown in §5, optimality of the upper bound (10) depends on the CSP formed by active joint states. Since $f_A = 0$ implies $\lceil f_A \rceil = 1$ (where 1 denotes the universal relation), adding a zero constraint to the WCSP means adding a universal relation to this CSP. After reparameterization (9), a zero constraint $f_A = 0$ becomes f_A^φ which is no longer zero and a universal relation $\lceil f_A \rceil = 1$ becomes $\lceil f_A^\varphi \rceil$ which is no longer universal.

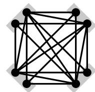
By Theorem 8, the relaxation is tight iff $\lceil f^\varphi \rceil$ is satisfiable. It can happen that the CSP formed only by relations $\lceil f_A^\varphi \rceil$ with $f_A \neq 0$ is satisfiable but the whole CSP $\lceil f^\varphi \rceil$ is unsatisfiable. Example 9 shows this is indeed possible. Thus, *we must not ignore zero constraints* when testing for bound optimality and recovering an optimizer.

Example 9. First, we give an unsatisfiable ternary CSP $(V, E, X_V, \bar f)$ whose binary part is satisfiable. Let $V = (1,2,3,4)$, $X_V = \{0,1\}^V$, $E = \binom{V}{2} \cup \binom{V}{3}$. Let $\bar f$ be defined by $\bar f_{12} = \bar f_{13} = \bar f_{14} = \bar f_{23} = \bar f_{24} = \bar f_{34}$ and $\bar f_{123} = \bar f_{124} = \bar f_{134} = \bar f_{234}$ where relations $\bar f_{123}$ and $\bar f_{12}$ are given by (17). Thus, the CSP consists of four copies of the CSP from Example 4 glued together[12]. Its six binary relations are

11. These sub-WCSPs roughly correspond to the subproblems in the decomposition approach [17], [19], [20] (see §1.1) and to the 'slave' problems in the dual decomposition formulation [21].

12. This CSP is the ternary generalization of the well-known binary unsatisfiable CSP on three variables (the 'frustrated cycle'), given by $\bar f_{12} = \bar f_{13} = \bar f_{23}$ where $\bar f_{12}(x_1,x_2) = \bar x_1 x_2 \vee x_1 \bar x_2$.

shown below (the ternary relations are not visualized):



For $J = I(E)$, check that $\bar{f}$ is J-consistent. Any $x_{1234} \in \{(1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1),(0,0,0,0)\}$ satisfies all the six binary relations but none of them satisfies all the ternary relations. Hence, $\bar{f}$ is unsatisfiable.

Second, we give a WCSP (V,E,X_V,f) with the ternary constraints being zero and we give a diffusion fixed point φ such that $\bar{f} = \lceil f^\varphi \rceil$. This may look trivial because any CSP is indeed realizable as $\lceil f^\varphi \rceil$ for some f and φ. But we must not forget about the constraint that the ternary constraints are zero. E.g., $f_{123} = 0$ and hence, by (9), $f^\varphi_{123} = \varphi_{123,12} + \varphi_{123,13} + \varphi_{123,23}$, i.e., f^φ_{123} must be a sum of binary functions. We show such f and φ exist.

Let f and φ be defined by $f_{123} = f_{124} = f_{134} = f_{234} = 0$, $f_{12} = f_{13} = f_{14} = f_{23} = f_{24} = f_{34}$ and $\varphi_{123,12} = \varphi_{123,13} = \varphi_{123,23} = \varphi_{124,12} = \varphi_{124,14} = \varphi_{124,24} = \varphi_{134,13} = \varphi_{134,14} = \varphi_{134,34} = \varphi_{234,23} = \varphi_{234,24} = \varphi_{234,34}$, where

$$f_{12}(x_1,x_2) = 4\bar{x}_1\bar{x}_2 + 9(\bar{x}_1 x_2 + x_1\bar{x}_2) + 7x_1x_2$$
$$\varphi_{123,12}(x_1,x_2) = \bar{x}_1\bar{x}_2 + 2(\bar{x}_1 x_2 + x_1\bar{x}_2)$$

From (9) we get that f^φ is given by $f^\varphi_{123} = f^\varphi_{124} = f^\varphi_{134} = f^\varphi_{234} = \varphi_{123,12} + \varphi_{123,13} + \varphi_{123,23}$ and $f^\varphi_{12} = f^\varphi_{13} = f^\varphi_{14} = f^\varphi_{23} = f^\varphi_{24} = f^\varphi_{34} = f_{12} - \varphi_{123,12} - \varphi_{124,12}$ where

$$\begin{aligned} f^\varphi_{123}(x_1,x_2,x_3) &= 3\bar{x}_1\bar{x}_2\bar{x}_3 + 5(x_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3) \\ &\quad + 4(x_1x_2\bar{x}_3 + x_1\bar{x}_2x_3 + \bar{x}_1x_2x_3) \\ f^\varphi_{12}(x_1,x_2) &= 5(\bar{x}_1\bar{x}_2 + \bar{x}_1x_2 + x_1\bar{x}_2) + 4x_1x_2 \end{aligned}$$

Check that $f^\varphi_B(x_B) = \max_{x_{A \setminus B}} f^\varphi_A(x_A)$ for each $(A,B) \in J$, i.e., φ is a diffusion fixed point. Check that $\bar{f} = \lceil f^\varphi \rceil$.

Note a surprising property of f^φ_{123}: function f^φ_{123} is a sum of binary functions but (see Example 4) relation $\lceil f^\varphi_{123} \rceil$ is not a conjunction of any binary relations[13]. □

9.5 Adding Zero Constraints Incrementally

We have shown in §8 how the relaxation can be tightened incrementally by extending J. When combined with adding zero constraints[14], this can be seen as a cutting plane algorithm which adds sets of linear inequalities separating μ optimal in $\pi_{T(E,X_V)} P(E,X_V,J)$ from the marginal polytope $\pi_{T(E,X_V)} P(2^V, X_V, I(2^V))$. Here we focus mainly on the separation problem.

The separation test (§8.1) asks whether extending J by $J' \subseteq I(2^V)$ would lead to bound improvement. In general, this is answered by Proposition 17. However, if J' has a special form, this can be formulated in terms of *satisfiability of a sub-CSP* of $\lceil f^\varphi \rceil$. For a CSP with hypergraph E, we define its **restriction** to $F \subseteq E$ to be the CSP with hypergraph F and the relations inherited from the original CSP.

13. This suggests an interesting problem: Given $\bar{f}_V \colon X_V \to \{0,1\}$ and $E \subseteq 2^V$, find $f \colon T(E,X_V) \to \mathbb{R}$ such that $\bar{f}_V = \lceil \sum_{A \in E} f_A \rceil$ or show that no such f exists. For given (V,E,X_V), characterize the class of relations $\bar{f}_V$ realizable in this way.

14. The incremental scheme from §8 is not restricted to zero constraints, it can be used also with non-zero constraints. E.g., given a WCSP with hypergraph $E \cup F$ where constraints E are 'easy' (unary, binary) and constraints F are 'hard' (high arity), we can first solve constraints E and then incrementally extend J to include constraints F. In both cases, Proposition 17 applies.

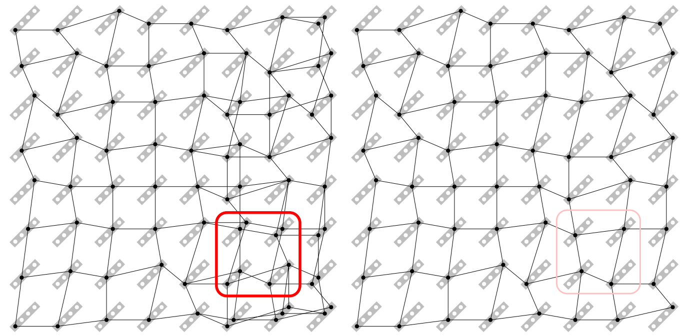


Fig. 2. Incrementally adding zero constraints.

Proposition 18. *Let (V,E,X_V,f) be a WCSP. Let $F \subseteq E$. Let us ask whether adding the zero constraint with scope $\bigcup F$, extending J by $J' = I_{\mathrm{SAT}}(F)$, and running max-sum diffusion will improve the bound.*

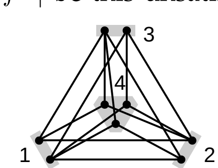
- *If the restriction of $\lceil f^\varphi \rceil$ to F is not satisfiable then the answer is 'yes'.*
- *If, in the restriction of $\lceil f^\varphi \rceil$ to F, every permitted joint state is satisfiable then the answer is 'no'.*

Proof: In Proposition 17, apply Propositions 5 and 6 on the restriction of $\lceil f^\varphi \rceil$ on hypergraph F. □

Example 10. Let us return to Example 8. Figure 2 (left) shows the CSP $\lceil f^\varphi \rceil$ after convergence of max-sum diffusion for relaxation J_1 of an instance `random` with size $m = 8$ and $|X_v| = 4$. The upper bound is not optimal because of the depicted unsatisfiable sub-CSP. Let A denote the four depicted variables. After adding the zero constraint with scope A, diffusion yielded Figure 2 (right) with an improved bound – here, the exact solution. Of course, many such steps are typically needed.

The inconsistent sub-CSP is supported only by 2 (rather than 4) states of each variable $v \in A$. Thus, instead of adding a zero constraint with 4 states, we could add a constraint with 2 states. For variables with large domains, this could drastically reduce the computational effort (see experiments in [29]). This would mean using the fine-grained hierarchy of coupling schemes[15] (7). □

Example 11. Let $\lceil f^\varphi \rceil$ be this unsatisfiable CSP:



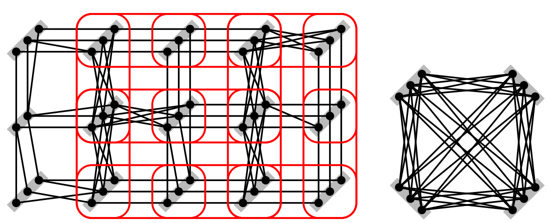
15. Using the fine-grained hierarchy of coupling schemes (i.e., using (7) rather than (2b)) would require adapting Algorithm 3 and the theorems in §6.2 because e.g. Theorem 11 does not hold. This would require some more research, for which paper [54] might be relevant.

The sub-CSP on variables $(1, 2, 4)$ is satisfiable but has unsatisfiable permitted joint states. Let us add constraint $f_{124} = 0$. This makes the PWC closure of the whole CSP empty. Thus, running diffusion on the sub-WCSP on variables $(1, 2, 4)$ will not improve the bound but running diffusion on the whole WCSP will.

As shown in [8, Figure 5b], the CSP $\lceil f^{\varphi} \rceil$ in the figure corresponds to a local optimum of φ that is not global. Thus, adding zero constraints sometimes can get the bound out of a local optimum. □

For some WCSP instances, it can be useful to add more complex subproblems than cycles.

Example 12. Consider the binary CSP $\lceil f^{\varphi} \rceil$ in figure (a), whose hypergraph E has 15 variables (i.e., 1-element hyperedges) and 22 2-element hyperedges:



(a) (b)

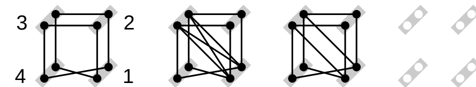
Let $J = I(E)$. The CSP is J-consistent, thus diffusion cannot improve the bound. It contains no unsatisfiable cycles but it is unsatisfiable because of the sub-CSP with hypergraph $F \subset E$ marked in red: F has $|\bigcup F| = 12$ variables and 14 2-element hyperedges. The sub-CSP is unsatisfiable because it can be reduced (by 'contracting' the identity relations) to the CSP in figure (b), which encodes the (impossible) task of 3-coloring the graph K_4.

Adding the zero constraint with scope $\bigcup F$ and extending J by $J' = I_{\mathrm{SAT}}(F) = \{ (\bigcup F, A) \mid A \in F \}$ makes the J-consistency closure of $\lceil f^{\varphi} \rceil$ empty (verify by Algorithm 2). Hence, diffusion will now improve the bound. Computing $\max_{x_{A \setminus B}} f^{\varphi}_A(x_A)$ for $A = \bigcup F$ needs more effort than if F was a cycle, but is still feasible. □

Given a family $\mathcal{J}$ of tentative subsets of $I(2^V)$, the separation problem consists in finding a 'small' $J' \in \mathcal{J}$ that will improve the bound. If $\mathcal{J}$ is small, J' can be searched exhaustively. If $\mathcal{J}$ has an intractable size, Proposition 18 translates the separation problem to finding an unsatisfiable sub-CSP of $\lceil f^{\varphi} \rceil$. It is subject to future research to discover polynomial-time algorithms to find an unsatisfiable sub-CSP of a CSP, where the sub-CSP are from some combinatorially large class (such as cycles). Finding *minimal* unsatisfiable sub-CSPs (though not in polynomial time) has been addressed in [55], [54].

Finally, note that extending J by elements of $\mathcal{J}$ one by one has a theoretical problem: it can happen that adding any single element of $\mathcal{J}$ does not improve the bound but adding the union of several elements of $\mathcal{J}$ does.

Example 13. Consider the CSP with $E = \{(1), (2), (3), (4), (1, 2), (2, 3), (3, 4), (1, 4)\}$ in figure (a):



(a) (b) (c) (d)

Adding simultaneously zero constraints with scopes $(1, 3)$, $(1, 2, 3)$, $(1, 3, 4)$ makes the PWC closure empty (figures b,c,d). However, adding any of these constraints separately does not make the PWC closure empty. □

9.6 k-consistency of Active Joint States

If E is closed to intersection and $J = I(E)$, $\lceil f^{\varphi} \rceil$ can always be made PWC by changing φ. PWC is the strongest local consistency of $\lceil f^{\varphi} \rceil$ achievable without adding zero constraints. By adding zero constraints, stronger local consistencies of $\lceil f^{\varphi} \rceil$ can be achieved.

By §4.2, adding all possible zero constraints of arity k and $k-1$ and running max-sum diffusion with $J = I(E)$ makes $\lceil f^{\varphi} \rceil$ k-consistent. Unlike in CSP, the previously added k-ary constraints *cannot* be removed after this [39, §8]. Thus, there is a difference in the rôle of k-consistency in CSP and WCSP. Enforcing strong k-consistency of a CSP requires adding only relations of arity less than k. For a WCSP, enforcing strong k-consistency of $\lceil f^{\varphi} \rceil$ requires adding constraints of arity less *or equal* to k. E.g., a binary CSP can be made 3-consistent and remain binary; for a binary WCSP, $\lceil f^{\varphi} \rceil$ can be made 3-consistent but only at the expense of making the WCSP ternary[16].

Similarly as in CSP (§4.2), strong k-consistency of $\lceil f^{\varphi} \rceil$ can be enforced in a more efficient way, by incrementally adding only *some* of all missing constraints of arity k or less. Supposing $\lceil f^{\varphi} \rceil$ is $(k-1)$-consistent, it is made k-consistent as follows. Whenever $|A| = k - 1$ and x_A cannot be extended to some variable v, we add constraint $f_{A \cup (v)} = 0$, and if $A \notin E$ we also add constraint $f_A = 0$. Then we set $J = I(E)$ and re-optimize the bound.

As making $\lceil f^{\varphi} \rceil$ 3-consistent requires adding new ($O(|V|^3)$ at worst) binary and ternary constraints, it is practical only for mid-size instances. Otherwise, partial forms of 3-consistency can be considered. One such form is suggested by Theorem 4: add only edges to E that make E chordal[17] (rather than complete). Since this can be still prohibitive, even fewer edges can be added.

Example 14. Let E be the $m \times m$ grid graph. We did a 'partial chordal completion' of E as follows: of all edges necessary to complete E to a chordal graph, we added only those edges (u, v) for which the Manhattan distance between nodes u and v in the original graph was not greater than d (this can be done by a simple modification of known algorithms for chordal completion). Then we triangulated the graph, added the resulting triangles and

16. Note, otherwise we'd get a paradox. A binary CSP with Boolean variables is tractable: it is satisfiable iff enforcing 3-consistency does not make it empty [34, §8.4.2]. If the active joint states of any binary WCSP with Boolean variables could be made 3-consistent without adding ternary constraints, we would have a polynomial algorithm to solve any binary WCSP with Boolean variables, which is intractable.

17. It is well-known that chordal completion is done also before the junction tree algorithm [32]. This is unrelated to its purpose here.

ran max-sum diffusion. The table shows the proportions of instances drawn from type `lines` (see Example 8) that were solved to optimality for various m and d. The number of added triangles is stated in parentheses.

	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$
m=10	0.71 (0)	0.85 (162)	0.88 (313)	0.92 (592)	0.95 (703)
m=15	0.40 (0)	0.54 (392)	0.58 (785)	0.77 (1469)	0.88 (1906)
m=20	0.11 (0)	0.26 (722)	0.28 (1483)	0.38 (2773)	0.57 (3167)
m=25	0.00 (0)	0.05 (1152)	0.06 (2392)	0.16 (4459)	0.33 (5819)

Comparing to Table 1 shows that the added triangles significantly tightened the relaxation. We remark that using the above described more efficient incremental algorithm would results in fewer added triangles. □

9.6.1 3-consistency and Cycle Inequalities

By Theorem 4, in a 3-consistent binary CSP every cycle is consistent[18]. This closely resembles the algorithm by Barahona and Mahjoub [31] (applied to WCSP in [26]) to separate cycle inequalities in the cut polytope [30]. While the algorithm [31] is primal and works only for Boolean variables, enforcing 3-consistency of $\lceil f^\varphi \rceil$ works in the dual space and for variables with any number of states. The precise relationship between the algorithm [31] and enforcing 3-consistency of $\lceil f^\varphi \rceil$ has yet to be clarified.

In particular, it is known that the *planar max-cut problem* is tractable, solved by a linear program over the *semimetric polytope* defined by the cycle inequalities [30, §27.3], cf. [56], [57]. The planar max-cut problem is equivalent to the WCSP (V, X_V, E, f) where $X_V = \{0,1\}^V$, $E \subseteq \binom{V}{2}$ is a planar graph, and constraints f have the form $f_{uv}(x_u, x_v) = c_{uv}[\![x_u = x_v]\!]$ with $c_{uv} \in \mathbb{R}$ (i.e., c_{uv} have arbitrary signs). It is an open problem whether this WCSP is solved by enforcing 3-consistency of $\lceil f^\varphi \rceil$.

10 Conclusion

We have tried to pave the way to WCSP solvers that would natively handle non-binary constraints (possibly of high-arity and represented by a black-box function) and use the cutting plane strategy.

Though we have considered only max-sum diffusion, most of the theory applies to the other bound-optimizing algorithms from §6.1, most notably to the augmenting DAG / VAC algorithm. Choosing which bound-optimizing algorithm to use is important and each algorithm has pros and cons:

- Max-sum diffusion is extremely simple and very flexible. Its drawback is that it is rather slow – for images, several times than the closely related and slightly more complex TRW-S.
- The augmenting DAG / VAC algorithm is complex and painful to implement efficiently [7] but has a unique advantage in its *incrementality*: if we run it to convergence and make a 'small' change to the WCSP, it typically needs a 'small' number of iterations to re-converge. This makes it suitable for cutting plane schemes (as observed in [28]), branch&cut search, and for incremental fixing of undecided variables. All the other bound-optimizing algorithms from §6.1 need a large number of iterations to re-converge, and this does not seem possible to avoid by any clever scheduling of iterations.
- In the light of the possibility to add zero constraints, the globally optimal algorithms (such as subgradient and smoothing methods) lose something of their attractivity. It is always a question whether to use these algorithms (which are slower than the local algorithms, especially when the LP relaxation is not tight) or to add zero constraints.

We considered only obtaining upper bounds on WCSP and we have not discussed rounding schemes to round undecided variables or using the LP relaxation as part of a search, such as branch&bound [12] or branch&cut.

18. Note, this shows that if all cycles of length 3 are added to a WCSP on a complete graph in Example 8 (relaxation J_2), the relaxation cannot be further improved by adding any cycles of greater lengths.

Acknowledgments

This research was supported by the European Commission grant 215078 and the Czech government grant MSM6840770038. I thank Mikhail I. Schlesinger for open discussions on his unpublished work. Martin Cooper and Thomas Schiex provided useful remarks on WCSP.

References

[1] P. Meseguer, F. Rossi, and T. Schiex, "Soft constraints," ch. 9, in [10].

[2] 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.

[3] M. I. Shlezinger, "Syntactic analysis of two-dimensional visual signals in noisy conditions," *Cybernetics and Systems Analysis*, vol. 12, no. 4, pp. 612–628, 1976, translation from Russian: Sintaksicheskiy analiz dvumernykh zritelnikh signalov v usloviyakh pomekh, Kibernetika, vol. 12, no. 4, pp. 113-130, 1976.

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

[5] "Personal communication of the author with Mikhail I. Schlesinger," 2000-2005.

[6] 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.

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

[8] ——, "A linear programming approach to max-sum problem: A review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1165–1179, July 2007.

[9] ——, "High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF)," in *Computer Vision and Pattern Recognition (CVPR) Conf., Anchorage, USA*, June 2008.

[10] F. Rossi, P. van Beek, and T. Walsh, *Handbook of Constraint Programming*. Elsevier, 2006.

[11] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin, "The complexity of soft constraint satisfaction," *Artificial Intelligence*, vol. 170, pp. 983–1016, 2006.

[12] M. C. Cooper, S. de Givry, M. Sánchez, T. Schiex, and M. Zytnicki, "Virtual arc consistency for weighted CSP," in *Conf. on Artificial Intelligence (AAAI)*, July 2008, pp. 253–258.

[13] U. Montanari, "Networks of constraints: Fundamental properties and application to picture processing," *Information Science*, vol. 7, pp. 95–132, 1974.
[14] A. Rosenfeld, R. A. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 420–433, June 1976.
[15] D. Schlesinger, "Exact solution of permuted submodular MinSum problems," in *Conf. Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), Ezhou, China*. Springer, 2007, pp. 28–38.
[16] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," UC Berkeley, Dept. of Statistics, Tech. Rep. 649, 2003.
[17] 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.
[18] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.
[19] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.
[20] J. K. Johnson, D. M. Malioutov, and A. S. Willsky, "Lagrangian relaxation for MAP estimation in graphical models," in *Allerton Conf. Communication, Control and Computing*, 2007.
[21] N. Komodakis, N. Paragios, and G. Tziritas, "MRF optimization via dual decomposition: Message-passing revisited," in *Intl. Conf. Computer Vision (ICCV)*, 2007.
[22] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.
[23] Y. Weiss, C. Yanover, and T. Meltzer, "MAP estimation, linear programming and belief propagation with convex free energies," in *Conf. Uncertainty in Artificial Intelligence (UAI)*, 2007.
[24] 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.
[25] 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, no. 3–5, pp. 89–97, 1998.
[26] D. Sontag and T. Jaakkola, "New outer bounds on the marginal polytope," in *Neural Information Processing Systems (NIPS)*, 2007.
[27] M. P. Kumar and P. H. S. Torr, "Efficiently solving convex relaxations for MAP estimation," in *Intl. Conf. on Machine Learning (ICML)*. ACM, 2008, pp. 680–687.
[28] N. Komodakis and N. Paragios, "Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles," in *European Conf. on Computer Vision (ECCV)*, 2008.
[29] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss, "Tightening LP relaxations for MAP using message passing," in *Conf. Uncertainty in Artificial Intelligence (UAI)*, 2008.
[30] M. M. Deza and M. Laurent, *Geometry of Cuts and Metrics*. Springer, Berlin, 1997.
[31] F. Barahona and A. R. Mahjoub, "On the cut polytope," *Mathematical Programming*, vol. 36, no. 2, pp. 157–173, 1986.
[32] S. Lauritzen, *Graphical Models*. Oxford University Press, 1996.
[33] A. Mackworth, "Constraint satisfaction," in *Encyclopaedia of Artificial Intelligence*. Wiley, 1991, pp. 285–292.
[34] D. Cohen and P. Jeavons, "The complexity of constraint languages," ch. 8, in [10].
[35] J.K.Pearson and P.G.Jeavons, "A survey of tractable constraint satisfaction problems," Royal Holloway, University of London, Tech. Rep. CSD-TR-97-15, July 1997.
[36] C. Bessiere, "Constraint propagation," ch. 3, in [10].
[37] R. Debruyne and C. Bessière, "Domain filtering consistencies," *Journal of Artificial Intelligence Research*, no. 14, pp. 205–230, 2001.
[38] E. C. Freuder, "A sufficient condition for backtrack-free search," *J. ACM*, vol. 29, no. 1, pp. 24–32, 1982.
[39] T. Werner, "Marginal consistency: Unifying constraint propagation on commutative semirings," in *Intl. Workshop on Preferences and Soft Constraints (co-located with Conf. on Principles and Practice of Constraint Programming)*, September 2008, pp. 43–57.
[40] C. Bliek and D. Sam-Haroud, "Path consistency on triangulated constraint graphs," in *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, 1999, pp. 456–461.
[41] M. I. Schlesinger, "False minima of the algorithm for minimizing energy of max-sum labeling problem," 1976, Glushkov Institute of Cybernetics, Kiev, USSR. Unpublished.
[42] V. N. Kolmogorov and M. J. Wainwright, "On the optimality of tree-reweighted max-product message-passing," in *Conf. Uncertainty in Artificial Intelligence (UAI)*, 2005.
[43] A. Globerson and T. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations," in *Neural Information Processing Systems (NIPS)*, 2008, pp. 553–560.
[44] T. Werner and A. Shekhovtsov, "Unified framework for semiring-based arc consistency and relaxation labeling," in *12th Computer Vision Winter Workshop, St. Lambrecht, Austria*. Graz University of Technology, February 2007, pp. 27–34.
[45] M. I. Schlesinger and V. V. Giginjak, "Solving (max,+) problems of structural pattern recognition using equivalent transformations," *Upravlyayushchie Sistemy i Mashiny (Control Systems and Machines), Kiev, Naukova Dumka*, vol. 1 and 2, 2007, in Russian, English translation available on www.
[46] P. Ravikumar, A. Agarwal, and M. J. Wainwright, "Message-passing for graph-structured linear programs: proximal projections, convergence and rounding schemes," in *Intl. Conf. on Machine Learning (ICML)*. ACM, 2008, pp. 800–807.
[47] W.-J. van Hoeve and I. Katriel, "Global constraints," ch. 7, in [10].
[48] C. Bessière and P. V. Hentenryck, "To be or not to be ... a global constraint," in *Conf. on Principles and Practice of Constraint Programming (CP)*, 2003, pp. 789–794.
[49] R. Gupta, A. A. Diwan, and S. Sarawagi, "Efficient inference with cardinality-based clique potentials," in *Intl. Conf. on Machine Learning (ICML)*, 2007, pp. 329–336.
[50] M. I. Schlesinger and B. Flach, "Some solvable subclasses of structural recognition problems," in *Czech Pattern Recognition Workshop*. Czech Pattern Recognition Society, 2000.
[51] D. M. Topkis, *Supermodularity and Complementarity*, ser. Frontiers of Economic Research. Princeton University Press, 1998.
[52] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwal, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for Markov random fields," in *Eur. Conf. Computer Vision (ECCV)*, 2006, pp. II: 16–29.
[53] V. Kolmogorov and C. Rother, "Comparison of energy minimization algorithms for highly connected graphs," in *European Conf. Computer Vision (ECCV)*, 2006, pp. II: 1–15.
[54] E. Gregoir, B. Mazure, and C. Piette, "MUST: Provide a finer-grained explanation of unsatisfiability," in *Conf. on Principles and Practice of Constraint Programming (CP)*, 2007, pp. 317–331.
[55] F. Hemery, C. Lecoutre, L. Sais, and F. Boussemart, "Extracting MUCs from constraint networks," in *Europ. Conf. on Artificial Intelligence (ECAI)*, Trento, Italy, 2006, pp. 113–117.
[56] A. Globerson and T. Jaakkola, "Approximate inference using planar graph decomposition," in *Neural Information Processing Systems (NIPS)*, 2006, pp. 473–480.
[57] N. N. Schraudolph and D. Kamenetsky, "Efficient exact inference in planar Ising models," in *Neural Information Processing Systems (NIPS)*, 2008, pp. 1417–1424.

Tomáš Werner spent most of his time at the Center for Machine Perception, a research group at the Czech Technical University in Prague. In 1999, he defended there his PhD thesis on multiple view geometry in computer vision. Since 2000, he spent 1.5 years in the Visual Geometry Group at the Oxford University, U.K., and then returned to Prague. Until 2002 his main interest was multiple view geometry and since then, optimization and Markov random fields.