How to Compute Primal Solution from Dual One in LP Relaxation of MAP Inference in MRF?

Tomáš Werner
Center for Machine Perception, Czech Technical University, Prague

Abstract

In LP relaxation of MAP inference in Markov random fields (MRF), the primal LP maximizes the MAP objective over relaxed labelings (pseudomarginals) and the dual LP minimizes an upper bound on the true MAP solution by reparameterizations. Having solved the dual LP, we have no direct access to the corresponding primal solution. We propose a simple way to compute an optimal primal solution from an optimal dual solution. Precisely, we given an algorithm that either shows that the upper bound for a given problem can be further decreased by reparameterizations (i.e., it is not dual-optimal) or computes the corresponding optimal relaxed labeling. This is done by first removing inactive dual constraints and then solving the resulting feasibility problem by a very simple message-passing algorithm, sum-product diffusion.

MAP inference in undirected graphical models (Markov random fields, MRF) [16] leads to the following NP-hard combinatorial optimization problem: given a set of variables and a set of functions of (small) subsets of the variables, maximize the sum of the functions over all the variables. The problem has a natural LP relaxation, proposed independently in [15, 8, 16]. The primal task in this LP relaxation maximizes the MAP objective over relaxed labelings (pseudomarginals), the dual LP minimizes an upper bound on the true MAP solution by reparameterizations (equivalent transformations) of the original problem.

Currently, the only known algorithms able to compute the LP relaxation for large-scale instances are dual. Examples are algorithms based on averaging max-marginals [10, 6, 4, 5], the Augmenting DAG / VAC algorithm [9, 1], subgradient methods [7, 14], or smoothing approaches [20, 17, 5, 12].

Having an optimal dual solution, it is not easy to compute the corresponding primal solution (optimal relaxed labeling) for large-scale instances. However, this primal solution can be sometimes useful. In this paper, we present a simple algorithm to compute an optimal primal solution from an optimal dual solution in the LP relaxation. Precisely, suppose somebody gives us a problem and claims it has the minimal upper bound among all its reparameterizations. We want to either disprove this claim or compute the corresponding primal optimum.

Given a dual solution, we first remove inactive dual constraints, which means setting the corresponding primal variables (pseudomarginals) to zero. We are left with a feasibility task. This feasibility task is replaced with its a smooth optimization task such that the primal optimum of the smoothed task is a solution of the feasibility task. The optimum of the smoothed task can be computed with a very simple message passing algorithm, which we call *sum-product diffusion*. If the dual solution was not optimal, this is detected during sum-product diffusion and a decreasing direction for the upper bound is provided.

We build our paper around the particular form of LP relaxation proposed in [15] and the max-sum diffusion algorithm [10], which were reviewed in [18]. Since researchers are most familiar with problems with pairwise interactions, we present our algorithm on these – but it can be straightforwardly extended to LP relaxation of problems of arbitrary order (arity), as proposed in [19, 3].

1 MAP inference in MRF and its LP relaxation

Let V be a set of variables and $E \subseteq \binom{V}{2}$ a set of variable pairs, thus (V, E) is a graph. Variable $u \in V$ attains states x_u from a finite domain X_u. An assignment (or labeling) is a tuple $x \in X$, where X is the Cartesian product of the domains X_u for all $u \in V$. We want to maximize the function

$$F(x \,|\, \mathbf{f}) = \Big[\sum_{u \in V} f_u(x_u) + \sum_{uv \in E} f_{uv}(x_u, x_v) \Big] \tag{1}$$

over all $x \in X$, where the functions $f_u\colon X_u \to [-\infty, \infty)$ and $f_{uv}\colon X_u \times X_v \to [-\infty, \infty)$ are given. All numbers $f_u(x_u)$ and $f_{uv}(x_u, x_v)$ form a single vector $\mathbf{f} \in [-\infty, \infty)^T$, where

$$T = \{\, (u, x_u) \mid u \in V,\ x_u \in X_u \,\} \cup \{\, (uv, x_u x_v) \mid uv \in E,\ x_u \in X_u,\ x_v \in X_v \,\}$$

denotes the set of all states of all the functions. We understand E as an undirected graph, adopting that $f_{uv}(x_u, x_v) = f_{vu}(x_v, x_u)$. We will need also the directed version E^* of E, such that for any undirected edge $uv \in E$ we have arcs in both directions $uv \in E^*$ and $vu \in E^*$.

Here is the primal (left) and the dual (right) of the LP relaxation of the problem [15, 18]:

$$\max_{\boldsymbol{\mu} \in \Lambda} \langle \mathbf{f}, \boldsymbol{\mu} \rangle \;=\; \min_{\boldsymbol{\varphi}} U(\mathbf{f}^{\boldsymbol{\varphi}}) \,. \tag{2}$$

In the primal, $\Lambda \subseteq [0, 1]^T$ is the set of vectors $\boldsymbol{\mu} \in [0, 1]^T$ satisfying

$$\sum_{x_v} \mu_{uv}(x_u, x_v) = \mu_u(x_u) \,, \qquad uv \in E^*,\ x_u \in X_u \tag{3a}$$

$$\sum_{x_u} \mu_u(x_u) = 1 \,, \qquad u \in V \,. \tag{3b}$$

The set Λ is often referred to as the *local marginal polytope* and the components of $\boldsymbol{\mu}$ as *pseudomarginals* [16]. A vector $\boldsymbol{\mu} \in \Lambda$ is the collection of distributions μ_u and μ_{uv}, subject to marginalization constraints (3a) and normalization constraints (3b). In the scalar product $\langle \mathbf{f}, \boldsymbol{\mu} \rangle$ we define $-\infty \times 0 = 0$.

The meaning of the dual can be understood by combining two concepts, reparameterizations and an upper bound on (1), as shown next.

A *reparameterization* is a linear transformation of vector $\mathbf{f}$ that preserves $F(x \,|\, \mathbf{f})$ for all $x \in X$. The simplest reparameterization is done as follows: pick any $uv \in E^*$, subtract an arbitrary unary function (usually called a 'message') $\varphi_{uv}\colon X_u \to (-\infty, \infty)$ from function f_u and add it to function f_{uv}:

$$f_u(x_u) \leftarrow f_u(x_u) - \varphi_{uv}(x_u) \tag{4a}$$

$$f_{uv}(x_u, x_v) \leftarrow f_{uv}(x_u, x_v) + \varphi_{uv}(x_u) \,. \tag{4b}$$

This preserves $f_u + f_{uv}$ and hence $F(\cdot\,|\,\mathbf{f})$. Applying (4) to all $uv \in E^*$ yields

$$f^{\boldsymbol{\varphi}}_u(x_u) = f_u(x_u) - \sum_{v \in N_u} \varphi_{uv}(x_u) \tag{5a}$$

$$f^{\boldsymbol{\varphi}}_{uv}(x_u, x_v) = f_{uv}(x_u, x_v) + \varphi_{uv}(x_u) + \varphi_{vu}(x_v) \ , \tag{5b}$$

where $N_u = \{\, v \mid uv \in E \,\}$ is the set of neighbors of variable u. Vector $\boldsymbol{\varphi}$ is the collection of all the numbers $\varphi_{uv}(x_u)$, thus $\mathbf{f}^{\boldsymbol{\varphi}}$ denotes the reparameterization of vector $\mathbf{f}$ by messages $\boldsymbol{\varphi}$. Reparameterizations preserve not only $F(\cdot\,|\,\mathbf{f})$ but also the primal objective, i.e., for any $\boldsymbol{\mu} \in \Lambda$ we have $\langle \mathbf{f}, \boldsymbol{\mu} \rangle = \langle \mathbf{f}^{\boldsymbol{\varphi}}, \boldsymbol{\mu} \rangle$.

The function U in the dual is an upper bound on $F(\cdot\,|\,\mathbf{f})$,

$$U(\mathbf{f}) = \sum_{u \in V} \max_{x_u} f_u(x_u) + \sum_{uv \in E} \max_{x_u, x_v} f_{uv}(x_u, x_v) \geq \max_{x \in X} F(x\,|\,\mathbf{f}) \ . \tag{6}$$

The meaning of the dual is now evident: it minimizes the upper bound (6) by reparameterizations (5).

We will say that $\mathbf{f}$ is a *dual optimum* if it has the least upper bound among all its reparameterizations, i.e., $U(\mathbf{f}) = \min_{\boldsymbol{\varphi}} U(\mathbf{f}^{\boldsymbol{\varphi}})$. Note that this is a slight abuse of terminology because the decision variables in the dual in (2) are $\boldsymbol{\varphi}$ rather than $\mathbf{f}$.

1.1 Relations between the primal and dual

The primal and dual are related by the well-known duality theorems:

- *Weak duality*: For any $\boldsymbol{\mu} \in \Lambda$ and any $\mathbf{f}$, we have $\langle \mathbf{f}, \boldsymbol{\mu} \rangle \leq U(\mathbf{f})$.
- *Strong duality*: For $\boldsymbol{\mu} \in \Lambda$, we have $\langle \mathbf{f}, \boldsymbol{\mu} \rangle = U(\mathbf{f})$ if and only if $\boldsymbol{\mu}$ is a primal optimum and $\mathbf{f}$ is a dual optimum.
- *Complementary slackness*: For $\boldsymbol{\mu} \in \Lambda$, we have $\langle \mathbf{f}, \boldsymbol{\mu} \rangle = U(\mathbf{f})$ if and only if

$$\big[\max_{y_u} f_u(y_u) - f_u(x_u)\big]\mu_u(x_u) = 0 \tag{7a}$$

$$\big[\max_{y_u, y_v} f_{uv}(y_u, y_v) - f_{uv}(x_u, x_v)\big]\mu_{uv}(x_{uv}) = 0 \ . \tag{7b}$$

If $\mathbf{f}$ is not a dual optimum then there exists no $\boldsymbol{\mu} \in \Lambda$ satisfying (7).

1.2 Computing primal solution as a feasibility problem

Our task in this paper is to compute a primal optimum from a dual optimum. More precisely, somebody gives us a vector $\mathbf{f}$ and claims it has the minimal upper bound among all its reparameterizations, $U(\mathbf{f}) = \min_{\boldsymbol{\varphi}} U(\mathbf{f}^{\boldsymbol{\varphi}})$. Our task is either to disprove this claim or to compute $\boldsymbol{\mu} \in \Lambda$ such that $\langle \mathbf{f}, \boldsymbol{\mu} \rangle = U(\mathbf{f})$.

Let us replace the given vector $\mathbf{f} \in [-\infty, \infty)^T$ with vector $\mathbf{g} \in \{-\infty, 0\}^T$ defined as

$$g_u(x_u) = \begin{cases} -\infty & \text{if } f_u(x_u) < \max_{y_u} f_u(y_u) \\ 0 & \text{if } f_u(x_u) = \max_{y_u} f_u(y_u) \end{cases} \tag{8a}$$

$$g_{uv}(x_u, x_v) = \begin{cases} -\infty & \text{if } f_{uv}(x_u, x_v) < \max_{y_u, y_v} f_{uv}(y_u, y_v) \\ 0 & \text{if } f_{uv}(x_u, x_v) = \max_{y_u, y_v} f_{uv}(y_u, y_v) \end{cases} . \tag{8b}$$

We assume that $U(\mathbf{f}) > -\infty$, hence $U(\mathbf{g}) = 0$. Replacing $\mathbf{f}$ with $\mathbf{g}$ can be seen as removing inactive dual constraints and setting the corresponding primal variables to zero. This does not change our task. Two cases can arise:

- $\mathbf{f}$ *is not a dual optimum*: Clearly, $\mathbf{f}$ is a dual optimum if and only if $\mathbf{g}$ is a dual optimum. Hence, there exists $\boldsymbol{\varphi}$ such that $U(\mathbf{g}^{\boldsymbol{\varphi}}) < 0$. This $\boldsymbol{\varphi}$ is at the same time a decreasing direction for $U(\mathbf{f})$, i.e., there exists $\lambda > 0$ such that $U(\mathbf{f}^{\lambda\boldsymbol{\varphi}}) < U(\mathbf{f})$.
- $\mathbf{f}$ *is a dual optimum*: Clearly, if $\mathbf{f}$ is a dual optimum then problems $\mathbf{f}$ and $\mathbf{g}$ have the same set of optimal primal solutions. Hence, there exists $\boldsymbol{\mu} \in \Lambda$ satisfying (7), which is the desired primal optimum. Condition (7) can be written in short as $\langle \mathbf{g}, \boldsymbol{\mu} \rangle = 0$, recalling that $-\infty \times 0 = 0$.

Now, our task in this paper reduces to the following feasibility problem:

Problem 1. *Given* $\mathbf{g} \in \{-\infty, 0\}^T$*, find* $\boldsymbol{\mu} \in \Lambda$ *satisfying* $\langle \mathbf{g}, \boldsymbol{\mu} \rangle = 0$*. If such* $\boldsymbol{\mu}$ *does not exist, find* $\boldsymbol{\varphi}$ *such that* $U(\mathbf{g}^{\boldsymbol{\varphi}}) < 0$*.*

2 Solving the feasibility problem

Here we describe a solution to Problem 1. In short, the idea is to replace task (2) with its smoothed version such that for $\mathbf{g} \in \{-\infty, 0\}^T$, the primal optimum of the smoothed task is a primal optimum of the original task. The (dual and primal) global optimum of the smoothed task can be computed with a very simple message passing algorithm.

Although the smoothed version of (2) will be eventually applied to $\mathbf{g} \in \{-\infty, 0\}^T$, we formulate it for $\mathbf{g} \in [-\infty, 0)^T$. It reads

$$\max_{\boldsymbol{\mu} \in \Lambda} \langle \mathbf{g} - \log \boldsymbol{\mu}, \boldsymbol{\mu} \rangle \quad = \quad \min_{\boldsymbol{\varphi}} \tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}}) \; . \tag{9}$$

The primal (left) can be understood as a minimization of a convex free energy (here, maximization of negative energy). In the dual (right), we have

$$\tilde{U}(\mathbf{g}) = \sum_{u \in V} \bigoplus_{x_u} g_u(x_u) + \sum_{uv \in E} \bigoplus_{x_u, x_v} g_{uv}(x_u, x_v) \geq \bigoplus_{x \in X} F(x \,|\, \mathbf{g}) \; . \tag{10}$$

where $\bigoplus_i a_i = \log \sum_i \exp a_i$ denotes the *log-sum-exp operation*. The right-hand expression in (10) is the *log-partition function*, hence $\tilde{U}$ is an upper bound on the log-partition function. The bound is too loose to be useful for approximating (log-)partition function but this is not our task here. The dual is a differentiable convex task, it is at optimum if

$$\bigoplus_{x_v} g^{\boldsymbol{\varphi}}_{uv}(x_u, x_v) = g^{\boldsymbol{\varphi}}_u(x_u) \tag{11}$$

holds for all $uv \in E^*$ and x_u. The primal and dual optimum are related by

$$\mu_u(x_u) = \frac{\exp g^{\boldsymbol{\varphi}}_u(x_u)}{\sum_{x_u} \exp g^{\boldsymbol{\varphi}}_u(x_u)} \; , \quad \mu_{uv}(x_u, x_v) = \frac{\exp g^{\boldsymbol{\varphi}}_{uv}(x_u, x_v)}{\sum_{x_u, x_v} \exp g^{\boldsymbol{\varphi}}_{uv}(x_u, x_v)} \; . \tag{12}$$

We will not prove here the duality relation (9), the inequality in (10), and the optimality condition (11). This is easy and it can be found e.g. in [20]. Plugging (12) into (11) verifies the marginalization condition (3a), hence $\boldsymbol{\mu} \in \Lambda$.

We can see that to solve the dual task in (9), we need to reparameterize $\mathbf{g}$ to satisfy the stationary condition (11). Then the primal solution can be read off from (12).

Note, the approach has the desirable property that pseudomarginals $\boldsymbol{\mu}$ need not be explicitly stored in the memory, they are given implicitly by (12). Therefore we need to store only unary functions $\boldsymbol{\varphi}$ rather than unary and binary functions $\boldsymbol{\mu}$.

2.1 Enforcing arc consistency

It can happen that condition (11) is impossible to satisfy by any choice of $\boldsymbol{\varphi}$. This is because reparameterizations cannot change a finite weight to an infinite one or *vice versa* (note, messages cannot take infinite values) – in other words, $g_u^{\boldsymbol{\varphi}}(x_u) > -\infty$ if and only if $g_u(x_u) > -\infty$, and similarly for g_{uv}. Therefore, the finite part of $\mathbf{g}$ has to satisfy the property known as *arc consistency* [2]: for all $uv \in E^*$ and x_u we have

$$\big[\max_{x_v} g_{uv}(x_u, x_v) > -\infty \iff g_u(x_u) > -\infty\big] \tag{13}$$

Polynomial algorithms exist that recursively set some of the weights $\mathbf{g}$ to $-\infty$ until $\mathbf{g}$ becomes arc consistent. This is known as *relaxation labeling* [13] or, in constraint satisfaction, as *enforcing arc consistency* or *constraint propagation* [2]. It is outlined in Algorithm 1.

Algorithm 1 Enforcing arc consistency.

repeat
  Find $uv \in E^*$ and x_u violating (13).
  $g_u(x_u) \leftarrow -\infty$
  $g_{uv}(x_u, x_v) \leftarrow -\infty$ for all x_v
until no change is possible

It can be shown that enforcing arc consistency does not change the optimum of (9). In particular, if Algorithm 1 sets all weights to $-\infty$ ($\mathbf{g} = -\boldsymbol{\infty}$), Problem 1 is infeasible.

2.2 Message passing algorithm

The dual in (9) can be solved by coordinate descent, which leads to a message passing algorithm. The algorithm repeats the following iteration until convergence:

Pick any $uv \in E^*$ and enforce equality (11) by reparameterization (4).

This strictly monotonically decreases $\tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}})$. On convergence, (11) holds globally.

Algorithm 2 The $(\oplus, +)$-diffusion algorithm.

repeat
  for $uv \in E^*$ and $x_u \in X_u$ such that $g_u(x_u) > -\infty$ **do**
    $\varphi_{uv}(x_u) \leftarrow \varphi_{uv}(x_u) + \frac{1}{2}\Big[g_u^{\boldsymbol{\varphi}}(x_u) - \bigoplus_{x_v} g_{uv}^{\boldsymbol{\varphi}}(x_u, x_v)\Big]$
  end for
until convergence

This is summarized in Algorithm 2. It is precisely analogical to max-sum diffusion [10, 18], only the function max was replaced with the log-sum-exp function $\oplus$. The algorithm assumes that $\mathbf{g}$ is arc consistent – then, the condition $g_u(x_u) > -\infty$ ensures that $\varphi_{uv}(x_u)$ is never set to $-\infty$ or ∞.

2.3 Solving the feasibility problem

Let us put things together and see how to solve Problem 1. For $\mathbf{g} \in \{-\infty, 0\}^T$, any $\boldsymbol{\mu} \in \Lambda$ satisfying $\langle \mathbf{g} - \log \boldsymbol{\mu}, \boldsymbol{\mu} \rangle > -\infty$ satisfies also $\langle \mathbf{g}, \boldsymbol{\mu} \rangle = 0$. Hence, any $\boldsymbol{\mu}$ feasible to the primal in (9) solves Problem 1.

For any $\mathbf{g}$ we have $\tilde{U}(\mathbf{g}) \geq U(\mathbf{g})$ because the log-sum-exp function upper bounds the maximum, $\bigoplus_i a_i \geq \max_i a_i$, Hence, if during Algorithm 2 we get $\tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}}) < 0$, it implies $U(\mathbf{g}^{\boldsymbol{\varphi}}) < 0$.

Thus, Problem 1 is solved as follows:

1. Enforce arc consistency, Algorithm 1. If $\mathbf{g} = -\boldsymbol{\infty}$, stop (Problem 1 is infeasible).
2. Run Algorithm 2. If $\tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}}) < 0$ any time during the algorithm, stop (Problem 1 is infeasible).
3. Otherwise, let Algorithm 2 converge and then compute a solution $\boldsymbol{\mu}$ of Problem 1 from (12).

Before convergence of Algorithm 2, $\boldsymbol{\mu}$ given by (12) satisfies $\langle \mathbf{g}, \boldsymbol{\mu} \rangle = 0$ but violates the marginalization constraint (3a). On convergence, $\boldsymbol{\mu}$ satisfies (3a), hence $\boldsymbol{\mu} \in \Lambda$.

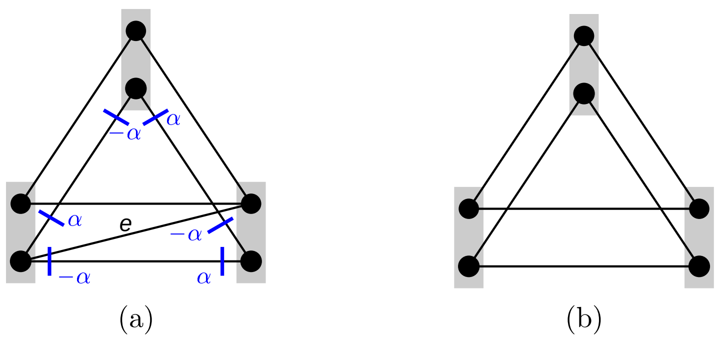

Figure 1: The depicted reparameterization decreases edge e to an arbitrarily small value α.

2.4 Unbounded messages

The algorithm has an interesting but undesired property: even if problem (9) is bounded, its optimum may be attained for $\|\boldsymbol{\varphi}\| \to \infty$. In that case, during Algorithm 2 some of the messages $\varphi_{uv}(x_u)$ will diverge to $-\infty$ or ∞.

Figure 1a shows an example of this phenomenon. It shows a problem with $V = 3$ variables, the complete graph E, and $|X_u| = 2$ labels. All the nodes have weights $g_u(x_u) = 0$. The shown edges have weights $g_{uv}(x_u, x_v) = 0$, the remaining (not shown) edges have weights $g_{uv}(x_u, x_v) = -\infty$.

Consider the reparameterization $\boldsymbol{\varphi}$ given by setting $\varphi_{uv}(x_u) = \alpha$ for the six messages depicted by blue line segments, and $\varphi_{uv}(x_u) = 0$ for the remaining six messages. For any α, this reparameterization changes the weight of edge e to $g^{\boldsymbol{\varphi}}(e) = -\alpha$ and leaves the remaining weights unchanged. For any $\alpha \geq 0$, we have $U(\mathbf{g}^{\boldsymbol{\varphi}}) = U(\mathbf{g}) = 0$. By complementary slackness, it necessarily follows that $\mu(e) = 0$.

Although the bound $U(\mathbf{g}^{\boldsymbol{\varphi}})$ is the same for any $\alpha \geq 0$, the smoothed upper bound $\tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}})$ decreases with increasing α. Since Algorithm 2 minimizes $\tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}})$, it will keep increasing α (and hence decreasing $g^{\boldsymbol{\varphi}}(e)$) without any limits. This must be so because by (12), for $\mu(e) = 0$ we need $g^{\boldsymbol{\varphi}}(e) = -\infty$.

This behavior can lead to numerical problems because in (5) we can get expressions like $g^{\boldsymbol{\varphi}}_{uv}(x_u, x_v) = g_{uv}(x_u, x_v) + \alpha - \alpha$ where α is a very large number. This can be alleviated by doing reparameterizations (4) 'in-place' by directly changing $\mathbf{g}$ (Algorithm 3) rather than by storing messages – but in that case we may instead have problems with error accumulation. Moreover, modifying binary functions $g_{uv}(\cdot,\cdot)$ typically needs more memory than storing only unary functions $\varphi_{uv}(\cdot)$.

Algorithm 3 The $(\oplus, +)$-diffusion algorithm, message-free version.

---

repeat
    for $uv \in E^*$ and $x_u \in X_u$ such that $g_u(x_u) > -\infty$ **do**
        $\varphi \leftarrow \frac{1}{2}\Big[g_u(x_u) - \bigoplus_{x_v} g_{uv}(x_u, x_v)\Big]$
        $g_u(x_u) \leftarrow g_u(x_u) - \varphi$
        $g_{uv}(x_u, x_v) \leftarrow g_{uv}(x_u, x_v) + \varphi$ for all x_v
    end for
until convergence

The phenomenon can be further clarified as follows. Let two vectors $\mathbf{g}, \mathbf{g}' \in [-\infty, \infty)^T$ be called *equivalent* if they define the same function $F(\cdot \,|\, \mathbf{g})$. An *equivalent transformation* is a change of vector $\mathbf{g}$ to its equivalent. Now, three classes of equivalent transformations can be distinguished [20]:

1. *Transformations that are compositions of a finite number of local reparameterizations (4).* These are precisely the linear transformations (5).
2. *Transformations that are compositions of an infinite number of local reparameterizations (4).* The resulting transformations are not all covered by (5) because (5) does not allow to change a finite weight to $-\infty$ or *vice versa*.
3. *Transformations that are not compositions of any number of local reparameterizations (4).* E.g., any unsatisfiable CSP is equivalent to the empty CSP.

The problems in Figure 1a and 1b are equivalent in the second sense. Hence, there exists no $\boldsymbol{\varphi}$ that would reparameterize Figure 1a to Figure 1b or *vice versa*.

2.5 Sum-product version of the algorithm

Algorithms 2 and 3 require time-consuming evaluation of the log-sum-exp function. By applying the exponential function to all involved quantities, these algorithms (and the theory in §2) can be translated from the semiring $([-\infty, \infty), \oplus, +)$ into the sum-product semiring $([0, \infty), +, \times)$. Then, the algorithms will use only addition, multiplication, division, and square root. We refer to the resulting algorithm as *sum-product diffusion*. Its message-free version is Algorithm 4. Its input is a vector $\mathbf{g} \in [0, \infty)^T$ that is assumed to be arc consistent, i.e., $\max_{x_v} g_{uv}(x_u, x_v) > 0$ if and only if $g_u(x_u) > 0$.

Algorithm 4 The sum-product diffusion algorithm, message-free version.

```
1: repeat
2:     for uv ∈ E* and x_u ∈ X_u such that g_u(x_u) > 0 do
3:         φ ← [ g_u(x_u) / Σ_{x_v} g_uv(x_u, x_v) ]^{1/2}
4:         g_u(x_u) ← g_u(x_u)/φ
5:         g_uv(x_u, x_v) ← g_uv(x_u, x_v) φ   for all x_v
6:     end for
7: until convergence
```

In the sum-product form, the core idea of our paper is especially obvious. Inequality (10) reads

$$\tilde{U}(\mathbf{g}) = \Big[\prod_{u\in V}\sum_{x_u} g_u(x_u)\Big]\Big[\prod_{uv\in E}\sum_{x_u,x_v} g_{uv}(x_u,x_v)\Big] \geq \sum_{x\in X} F(x\,|\,\mathbf{g}) \;, \tag{14}$$

where

$$F(x\,|\,\mathbf{g}) = \prod_{u\in V} g_u(x_u) \prod_{uv\in E} g_{uv}(x_u,x_v) \;. \tag{15}$$

The stationary condition (11) reads

$$\sum_{x_v} g_{uv}(x_u,x_v) = g_u(x_u) \;. \tag{16}$$

Now, Problem 1 is solved simply by applying sum-product diffusion to $\mathbf{g} \in \{0,1\}^T$. It is obvious that after convergence, the optimal $\boldsymbol{\mu}$ coincides with $\mathbf{g}$ (up to normalization) – because (16) is the marginalization condition (3a) we want to impose.

Moreover, the test for infeasibility of Problem 1 has an interesting interpretation in terms of the *constraint satisfaction problem* (CSP) [11]. A vector $\mathbf{g} \in \{0,1\}^T$ can be understood as a (pairwise) CSP, the functions $g_u\colon X_u \to \{0,1\}$ and $g_{uv}\colon X_u \times X_v \to \{0,1\}$ representing unary and binary relations. A *solution* of the CSP is an assignment $x \in X$ satisfying all the relations, i.e., $F(x\,|\,\mathbf{g}) = 1$. The CSP is *satisfiable* if it has at least one solution.

For $\mathbf{g} \in \{0,1\}^T$, the right-hand side of inequality (14) is the *number of solutions* of the CSP. This is known as the *counting constraint satisfaction problem* (#CSP). Thus, $\tilde{U}(\mathbf{g})$ is an upper bound on the number of solutions of the CSP. This bound is too loose to be useful, except in one situation: $\tilde{U}(\mathbf{g}) < 1$ implies that the CSP is unsatisfiable. Algorithm 4 minimizes $\tilde{U}(\mathbf{g})$ by reparameterizing $\mathbf{g}$, hence preserving $F(\cdot\,|\,\mathbf{g})$. If any time during the algorithm we get $\tilde{U}(\mathbf{g}) < 1$, we know that the CSP is unsatisfiable. Therefore, sum-product diffusion provides a *test to disprove satisfiability of a CSP*. This test is dissimilar to all tests based on local consistencies [2], used in constraint programming. However, note that it is not apparent at the first sight that this test is equivalent to satisfiability of Problem 1 – to show this, we needed the duality relation (9).

3 What about non-optimal dual solutions?

Many algorithms to tackle the dual in (2) yield only suboptimal dual solutions $\mathbf{f}$ – most importantly, the algorithms based on averaging max-marginals, such as max-sum diffusion [10, 18] or TRW-S [6]. An example of a problem that is a fixed point of these algorithms but is not a dual

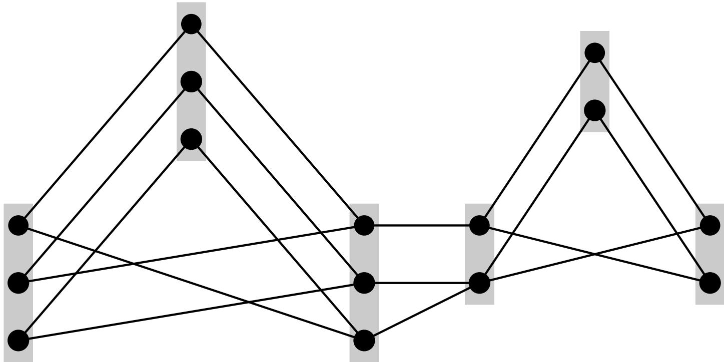

Figure 2: An example of a fixed point of max-sum diffusion (or TRW-S) that is not a dual optimum (given by Schlesinger [18]).

optimum is in Figure 2. In that case, Algorithm 2 does not converge and achieves $\tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}}) < 0$. Therefore, it would be useful to compute a *suboptimal* primal solution from a *suboptimal* dual solution. We do not know how to modify our algorithm in a principled way to achieve this. But one can think of several heuristics.

Of the duality relations (§1.1), one can sacrifice either primal feasibility or zero duality gap. One option is to minimize primal infeasibility (violation of (3a)) subject to zero duality gap – but we do not know how to do this for large instances. However, our algorithm can be modified to find a feasible primal solution with possibly small (but in general not minimal) duality gap. Instead of (8), let $\mathbf{g}$ be defined by

$$g_u(x_u) = \begin{cases} -\delta & \text{if } f_u(x_u) < \max_{y_u} f_u(y_u) - \varepsilon \\ 0 & \text{if } f_u(x_u) \geq \max_{y_u} f_u(y_u) - \varepsilon \end{cases} \tag{17a}$$

$$g_{uv}(x_u, x_v) = \begin{cases} -\delta & \text{if } f_{uv}(x_u, x_v) < \max_{y_u, y_v} f_{uv}(y_u, y_v) - \varepsilon \\ 0 & \text{if } f_{uv}(x_u, x_v) \geq \max_{y_u, y_v} f_{uv}(y_u, y_v) - \varepsilon \end{cases}. \tag{17b}$$

For any $\varepsilon \geq 0$ and $\delta > 0$, Algorithm 2 will converge (with either sign of $\tilde{U}(\mathbf{g}^{\boldsymbol{\varphi}})$) and (12) will be a feasible primal solution $\boldsymbol{\mu} \in \Lambda$. With a small ε and large δ, one can expect a reasonably small duality gap $U(\mathbf{f}) - \langle \mathbf{f}, \boldsymbol{\mu} \rangle$. Unfortunately, this cannot be guaranteed based on solely ε and δ because for unbounded weights $\mathbf{g}$ the gap can be unbounded in general. Currently, we cannot provide a bound on the duality gap based on ε, δ and $\mathbf{g}$.

4 Conclusion

We have proposed an algorithm to compute a primal optimal solution from an optimal dual solution in the LP relaxation of MAP-MRF inference. If the dual solution is not optimal, we provide a decreasing direction as a certificate of non-optimality. The main idea of our approach is extremely simple, in fact being summarized by §2.5. We have discussed an interesting but undesirable behavior of the algorithm, unbounded messages.

In this paper, we do not provide enough empirical evidence that our approach is useful in practice. We tested the algorithm on instances with a sparse (grid) graph E and random weights $\mathbf{f}$ drawn i.i.d. from the normal distribution. Dual solutions were obtained by max-sum diffusion. For instances with $V = 20^2$ variables, the dual solution was optimal in approx. 50% cases, for instances with $V = 50^2$ variables almost never. For optimal dual solutions, optimal primal solutions could always be found up to a small primal infeasibility. Unbounded messages never caused a serious problem.

We formulated the algorithm for one particular form of the LP dual [15, 18], closely related to max-sum diffusion [10, 18]. However, it can be easily extended to algorithms with tree-based updates, both using max-marginal averaging [6] and subgradients [7, 14]. In that case, the problem $\mathbf{g}$ would have to be defined in a different way than by (8).

Acknowledgment

This research has been supported by the European Community project FP7-ICT-247022 (MASH) and by the Czech government grant MSM6840770038. The author thanks Sebastian Nowozin for useful discussions.

References

[1] M. C. Cooper, S. de Givry, M. Sanchez, T. Schiex, M. Zytnicki, and T. Werner. Soft arc consistency revisited. *Artificial Intelligence*, 174(7-8):449–478, May 2010.

[2] Romuald Debruyne and Christian Bessière. Domain filtering consistencies. *Journal of Artificial Intelligence Research*, (14):205–230, 2001.

[3] V. Franc, S. Sonnenburg, and T. Werner. Cutting plane methods in machine learning. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011. To appear.

[4] Amir Globerson and Tommi Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Neural Information Processing Systems (NIPS)*, pages 553–560, 2008.

[5] Jason K. Johnson, Dmitry M. Malioutov, and Alan S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *Allerton Conf. Communication, Control and Computing*, 2007.

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

[7] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *Intl. Conf. Computer Vision (ICCV)*, 2007.

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

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

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

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

[12] Pradeep Ravikumar, Alekh Agarwal, and Martin J. Wainwright. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *Journal of Machine Learning Research*, 11:1043–1080, 2010.

[13] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. on Systems, Man, and Cybernetics*, 6(6):420–433, June 1976.

[14] Michail 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*, 1 and 2, 2007. In Russian, English translation available on www.

[15] M. I. Shlezinger. Syntactic analysis of two-dimensional visual signals in noisy conditions. *Cybernetics and Systems Analysis*, 12(4):612–628, 1976. Translation from Russian.

[16] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[17] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP estimation, linear programming and belief propagation with convex free energies. In *Conf. Uncertainty in Artificial Intelligence (UAI)*, 2007.

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

[19] Tomáš Werner. Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, August 2010. To appear in August 2010.

[20] Tomáš Werner and Alexander Shekhovtsov. Unified framework for semiring-based arc consistency and relaxation labeling. In *12th Computer Vision Winter Workshop, St. Lambrecht, Austria*, pages 27–34. Graz University of Technology, February 2007.