Convergence of Some Convex Message Passing Algorithms to a Fixed Point

Václav Voráček [1] **Tomáš Werner** [2]

Abstract

A popular approach to the MAP inference problem in graphical models is to minimize an upper bound obtained from a dual linear programming or Lagrangian relaxation by (block-)coordinate descent. This is also known as convex/convergent message passing; examples are max-sum diffusion and sequential tree-reweighted message passing (TRW-S). Convergence properties of these methods are currently not fully understood. They have been proved to converge to the set characterized by local consistency of active constraints, with unknown convergence rate; however, it was not clear if the iterates converge at all (to any point). We prove a stronger result (conjectured before but never proved): the iterates converge to a fixed point of the method. Moreover, we show that the algorithm terminates within $\mathcal{O}(1/\varepsilon)$ iterations. We first prove this for a version of coordinate descent applied to a general piecewise-affine convex objective. Then we show that several convex message passing methods are special cases of this method. Finally, we show that a slightly different version of coordinate descent can cycle.

1. Introduction

Maximum aposterior (MAP) inference in undirected graphical models (Markov random fields) (Wainwright & Jordan, 2008) leads to an NP-hard combinatorial optimization problem, which is also known as discrete energy minimization (Szeliski et al., 2006; Kappes et al., 2015) or valued constraint satisfaction (Meseguer et al., 2006; Cooper et al., 2010). Graphical models find many applications even in various areas of deep learning (Chen et al., 2015; Tourani et al., 2018; Munda et al., 2017).

[1]Tübingen AI center, University of Tübingen [2]Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. Correspondence to: Václav Voráček <vaclav.voracek@uni-tuebingen.de>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

A popular approach to MAP inference is the class of methods known as convex (or convergent) message passing. These are in fact versions of (block-)coordinate descent (BCD) applied to a dual linear programming (LP) or Lagrangian relaxation/decomposition of the problem, which aims to minimize an upper bound on the true optimal value. The earliest (and arguably the simplest) such method is max-sum diffusion (Kovalevsky & Koval, 1975), revisited in (Werner, 2007; 2010). Other examples are sequential tree-reweighted message passing (TRW-S) (Kolmogorov, 2006), max-product linear programming (MPLP) (Globerson & Jaakkola, 2008), max-marginal averaging (Johnson et al., 2007), and sequential reweighted message passing (SRMP) (Kolmogorov, 2015). Message-passing methods for MAP inference have a rich history, reviewing which is beyond the scope of this paper, see also (Meltzer et al., 2009; Ruozzi & Tatikonda, 2013; Sontag et al., 2012; Tourani et al., 2020).

Besides MAP inference, convex message passing methods have been applied to other combinatorial optimization problems (Wedelin, 1995; Swoboda et al., 2017a;b; Swoboda & Andres, 2017) including the general 0-1 ILPs (Lange & Swoboda, 2021; Abbas & Swoboda, 2022), outperforming commercial ILP solvers on many large-scale instances.

The dual LP or Lagrangian relaxations of MAP inference are convex non-smooth and/or constrained problems. Coordinate descent applied to a convex function is known to converge to a global minimum if the function is smooth (or its non-smooth part is separable) and has unique coordinate-wise minimizers (Bertsekas, 1999; Tseng, 2001) but for non-smooth and/or constrained problems it can get stuck in local (w.r.t. coordinate moves) minima (Warga, 1963). To alleviate this drawback, a number of *globally* optimal large-scale methods have been adapted to solve the relaxations, such as subgradient methods (Komodakis et al., 2011), bundle methods (Savchynskyy, 2012), ADMM (Martins et al., 2011), or adaptive diminishing smoothing (Savchynskyy et al., 2012). It was however observed in the experimental study by (Kappes et al., 2015) that for large sparse instances from computer vision, dual BCD methods with tree-structured blocks (such as TRW-S) are consistently faster and the obtained local optima are usually very good.

Convergence properties of the convex message-passing methods are currently not fully understood. The objective

value cannot increase in any iteration by definition, but many iterations actually keep it unchanged. It is known that a necessary (but not sufficient) condition for fixed points of the methods is a local consistency (arc consistency for max-sum diffusion, weak tree consistency for TRW-S) of the active constraints. For TRW-S, (Kolmogorov, 2006) showed that any limit point of the sequence of the iterates satisfies weak tree consistency. For max-sum diffusion, (Schlesinger & Antoniuk, 2011) showed somewhat more: the iterates converge to the set defined by arc consistency (but not necessarily to any single point); this was reviewed by (Savchynskyy, 2019). This result was generalized by (Werner et al., 2020) to BCD applied to any convex optimization problem, assuming that the block-wise minimizer is always chosen from the relative interior of the set of block-minimizers.

1.1. Contributions

As our *first contribution* in this paper, we prove the long-open conjecture, formulated for max-sum difussion by (Schlesinger & Antoniuk, 2011; Werner, 2007) but dating back to (Kovalevsky & Koval, 1975), that the iterates converge to a fixed point of the method. Moreover, we show that for any precision $\varepsilon > 0$ this happens in $\mathcal{O}(1/\varepsilon)$ iterations – to the best of our knowledge, this is the first result on convergence rate for these methods.

To that end, we first study (in §2) a seemingly simple iterative method, coordinate descent applied to minimization of the pointwise maximum of affine functions. This method was studied already in (Werner, 2017). Here, the objective function is non-smooth and can have non-unique coordinate-wise minimizers. Therefore, one must decide in each iteration which minimizer to choose. A natural way to resolve this ambiguity is to ignore those affine functions that do not depend on the current variable.

Then (in §2.1), we introduce a novel energy function that strictly decreases with any non-trivial update. It follows that the method cannot cycle. Assuming boundedness of this energy, we proceed to prove convergence to the fixed point and obtain an assymptotic upper bound on convergence rate.

Finally we show (in §3) that max-sum diffusion and max-marginal averaging in Lagrangian decomposition are special cases of the above algorithm, which allows us to transfer the above convergence results to them. Note that the result for max-marginal averaging is very general, since Lagrangian decomposition is applicable to many hard combinatorial optimization problems beyond MAP inference.

In our *second contribution*, we consider a slight modification of the above method, in which in every update the coordinate-wise minimizer is chosen to be the mid-point of the interval of all coordinate-wise minimizers. We show that in this case, the method can cycle.

2. Minimizing Maximum of Affine Functions

In this section, we consider coordinate descent applied to unconstrained minimization of a convex piecewise-affine function. Such a function can be always expressed as the pointwise maximum of affine functions,

$$f(x) = \max_{i\in[m]}(a_i^T x + b_i) = \max_{i\in[m]}(Ax+b)_i, \qquad (1)$$

where a_i^T are the rows of matrix $A = [a_{ij}] \in \mathbb{R}^{m\times n}$ and $b = (b_1, \dots, b_m) \in \mathbb{R}^m$. We aim at applications where A is large, sparse and its non-zero entries are small integers, often just $\{-1, 0, +1\}$. We will refer to $a_{i1}, \dots, a_{in}$ as the *coefficients* and to b_i as the *offset* of the ith affine function $a_i^T x + b_i$.

Starting from an initial point $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, in each iteration of coordinate descent we pick some $j \in [n]$ and minimize f over variable x_j while keeping the remaining variables $x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n$ fixed, i.e., we minimize the univariate function $x_j \mapsto f(x)$.

Here we assume that that the function $x_j \mapsto f(x)$ always has a minimum. This is ensured by the following condition, which we assume throughout the paper:

Assumption 2.1. The matrix A satisfies

$$\forall j : ((\exists i : a_{ij} < 0) \wedge (\exists i : a_{ij} > 0)). \qquad (2)$$

It says that for each variable x_j there is at least one affine function increasing in x_j and at least one affine function decreasing in x_j. If this is not the case for some j, the rows with non-zero elements in column j can be deleted from A without affecting the infimum of f, because the corresponding affine functions can be decreased arbitrarily without changing the other affine functions. This can be repeated until (2) becomes satisfied (if this makes A empty, then f is unbounded from below or constant). Thus, assumption (2) is purely technical and does not limit the following results in any way. We remark that (2) is a form of local consistency, called *sign consistency* in (Werner, 2017).

Example 2.2. The function $f(x_1, x_2) = \max\{x_1, -x_1, x_1 + x_2\}$ does not satisfy (2) because of variable x_2. But the affine function $x_1 + x_2$ can be omitted without affecting the minimum of f. The new function $f(x_1, x_2) = \max\{x_1, -x_1\}$ satisfies (2).

So we have ensured that the univariate function $x_j \mapsto f(x)$ always has at least one minimizer. But it may have more than one minimizer (an interval) because some of the affine function can be constant in x_j. Therefore, we need to introduce a rule to choose a single minimizer. A natural[1] such

[1]Another natural rule, see is to choose a mid-point of the interval of minimizers. We show in §4 that such a rule may lead to oscillation of iterates.

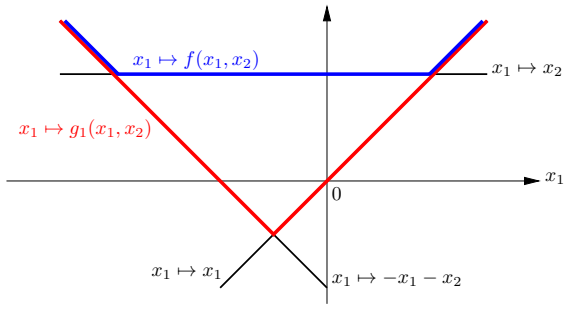

Figure 1. Plots of the functions $x_j \mapsto f(x)$ (in blue) and $x_j \mapsto g_j(x)$ (red) for the first update in Example 2.3 (so that $x_2 = 1$). Also shown are the three constituent affine functions (black).

rule, considered in (Werner, 2017), is to *ignore* the affine functions that are constant in x_j, i.e., to replace f with the function

$$g_j(x) = \max_{i:\, a_{ij} \neq 0} (a_i^T x + b_i). \tag{3}$$

Assuming (2), for every $x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n \in \mathbb{R}$ the univariate function $x_j \mapsto g_j(x)$ has exactly one minimizer, x_j^*, which is the unique solution to the equation

$$\max_{i:\, a_{ij} < 0} (a_i^T x + b_i) = \max_{i:\, a_{ij} > 0} (a_i^T x + b_i). \tag{4}$$

For the special case $A \in \{-1, 0, 1\}^{m \times n}$, (4) has the closed-form solution

$$x_j^* = \frac{1}{2}\Big(\max_{i:\, a_{ij} < 0} (a_i^T x^{-j} + b_i) - \max_{i:\, a_{ij} > 0} (a_i^T x^{-j} + b_i) \Big) \tag{5}$$

where $x_k^{-j} = x_k$ for all $k \neq j$ and $x_j^{-j} = 0$.

Example 2.3. Let

$$f(x_1, x_2) = \max\{x_1,\, x_2,\, -x_1 - x_2\}. \tag{6}$$

According to (3), we thus have

$$g_1(x_1, x_2) = \max\{x_1,\, -x_1 - x_2\},$$
$$g_2(x_1, x_2) = \max\{x_2,\, -x_1 - x_2\}.$$

Let us show two iterations of the method, starting from the point $(x_1, x_2) = (1, 1)$.

To update x_1, we minimize the univariate function $x_1 \mapsto g_1(x_1, x_2) = \max\{x_1, -x_1 - 1\}$, which has the unique minimizer $x_1 = -\frac{1}{2}$. In contrast, the function $x_1 \mapsto f(x_1, x_2) = \max\{x_1, 1, -x_1 - 1\}$ attains its minima on the interval $[-2, 1]$, which contains the point $-\frac{1}{2}$ (see Figure 1). Note that this update actually did not decrease the objective (6), because $f(1, 1) = f(-\frac{1}{2}, 1) = 1$.

To update x_2, we minimize the function $x_2 \mapsto g_2(x_1, x_2) = \max\{x_2,\, \frac{1}{2} - x_2\}$, which has the unique minimizer $x_2 = \frac{1}{4}$. In this case, the function $x_2 \mapsto f(x_1, x_2) = \max\{-\frac{1}{2},\, x_2,\, \frac{1}{2} - x_2\}$ has the same unique minimizer. This update decreased the objective: $f(-\frac{1}{2}, 1) = 1 > f(-\frac{1}{2}, \frac{1}{4}) = \frac{1}{4}$.

One might think that the updates that do not decrease the objective can be skipped – but they cannot. Indeed, f cannot be decreased from the point $(x_1, x_2) = (1, 1)$ by changing any variable separately but the method nevertheless converges to the global minimizer $(x_1, x_2) = (0, 0)$ of f.

The above iterative method is summarized in Algorithm 1, including a stopping condition. A *fixed point* of the method is any point that satisfies (4) for all $j \in [n]$.

Algorithm 1 Minimizing pointwise maximum of affine functions by coordinate descent

Input: data A, b, initial point $x \in \mathbb{R}^n$, precision $\varepsilon \geq 0$
$\eta \leftarrow \infty$
while $\eta \geq \varepsilon$ **do**
  $\eta \leftarrow 0$
  for $j \in [n]$ **do**
    $x_j^* \leftarrow \operatorname{argmin}_{x_j \in \mathbb{R}} g_j(x)$
    $\eta \leftarrow \max\{|x_j - x_j^*|, \eta\}$
    $x_j \leftarrow x_j^*$
  end for
end while

Remark 2.4. One can consider an equivalent version of the algorithm, where we do not explicitly keep the variables x in the memory but instead modify the offsets b. Indeed, the vector $Ax + b$ does not change if we set $b \leftarrow Ax + b$ and then reset $x \leftarrow 0$. Detail can be found in (Werner, 2017). In MAP inference literature, this corresponds to 'message-free' versions of message-passing algorithms.

Of course, the method still may have fixed points that are not global minima. For example, the function

$$f(x_1, x_2) = \max\{x_1 - 2x_2,\, x_2 - 2x_1\} \tag{7}$$

is unbounded but any point $x_1 = x_2$ is fixed for the method. This is a well-known drawback of coordinate descent applied to non-smooth convex functions (Warga, 1963).

Next we present an example when some components of the vector $Ax + b$ (i.e., the values of some affine functions) decrease unboundedly during the iterations.

Example 2.5 ((Werner, 2017)). Let

$$f(x_1, x_2, x_3) = \max\{x_1 - x_2 - x_3,\, x_1 + 4,\, x_1 + x_2 + x_3,\, -x_1 + x_2 + 2\}. \tag{8}$$

Starting from $(x_1, x_2, x_3) = (0, 0, 0)$, the updates of

x_1, x_2, x_3 (in this order) read

$$x_1 \leftarrow -1 = \operatorname*{argmin}_{x_1} \max\{x_1,\, x_1+4,\, x_1,\, -x_1+2\},$$
$$x_2 \leftarrow -2 = \operatorname*{argmin}_{x_2} \max\{-x_2-1,\, x_2-1,\, x_2+3\},$$
$$x_3 \leftarrow \;\; 2 = \operatorname*{argmin}_{x_3} \max\{1-x_3,\, -3+x_3\}.$$

By these three updates, each component of the vector $Ax+b$ decreases by 1. This can be seen as decreasing each component of b by 1 and resetting $(x_1, x_2, x_3) \leftarrow (0,0,0)$, as in Remark 2.4. Thus, the vector $Ax+b$ will diverge.

It might seem this behaviour cannot occur if the objective f is bounded below – but it is not so. Consider, for example, the function $f'(x_1,x_2,x_3) = \max\{f(x_1,x_2,x_3), 0\}$ where f is given by (8), which is bounded below by 0. The zero affine function will be ignored in all iterations (not involved in any function g_j), so the values of the remaining four affine functions will be decreasing exactly the same way as in Example 2.5.

2.1. Convergence Analysis

The key ingredient of our convergence analysis is a novel 'energy function' (9b) that stricly decreases in every non-trivial (i.e., that changes some variable) update. Existence of such a function is not at all obvious – recall, in particular, that the objective $f(x)$ can remain unchanged in some iteration and these iterations cannot be omitted, see Example 2.3.

Definition 2.6. For any $y \in \mathbb{R}^m$ and $k \geq 2$, define

$$E_k(y,\pi) = \sum_{i\in[m]} k^i\, y_{\pi(i)}, \tag{9a}$$
$$E_k(y) = \max_{\pi} E_k(y,\pi) = \sum_{i\in[m]} k^i \operatorname{sort}(y)_i \tag{9b}$$

where k^i denotes the ith power of k, and in (9b) we maximize over all permutations π and the sort is in ascending order.

Note that the function (9b) is convex (although we will not need this property). While we present the function in the full generality, for MAP inference (in §3) it will suffice to consider only the case $k=2$.

Each iteration of the method minimizes, for some j, the univariate function $x_j \mapsto g_j(x)$, which is the pointwise maximum of univariate affine functions. This decreases some of the affine functions and increases some other. We have a bound for this increase because we know both the minimal and maximal slope

$$c = \min_{i,j:\, a_{ij}\neq 0} |a_{ij}|\,, \qquad C = \max_{i,j} |a_{ij}| \tag{10}$$

of the affine functions. This allows us to prove that the energy indeed stricty decreases.

Proposition 2.7. *Let Assumption 2.1 hold. Let c and C be given by (10). In every inner iteration of Algorithm 1, the energy $E_{1+C/c}(Ax+b)$ decreases by at least $c\,|x_j - x_j^*|$.*

Proof. Let x_j^* be the minimizer of the function $x_j \mapsto g_j(x)$. There is an affine function $a_i^T x + b_i$ with a positive slope and another with a negative slope in x_j, which at x_j^* are equal to the minimal value of the function $x_j \mapsto g_j(x)$ as in Equation (4). Thus, there is an affine function for which the minimum of the function $x_j \mapsto g_j(x)$ is attained at x_j^* whose value decreased by at least $c\,|x_j - x_j^*|$ in this iteration. The values of all the other affine functions could not increase by more than $C\,|x_j - x_j^*|$. By Lemma 2.8, the energy $E_{1+C/c}(Ax+b)$ decreased by at least $c\,|x_j - x_j^*|$ in this iteration. □

Lemma 2.8. *Let $c, C > 0$. Let $y, y' \in \mathbb{R}^m$ differ at positions $I \subseteq [m]$ satisfying $y'_i \leq y_i + C$ for all $i \in I$, and at the same time, there is also a position $i^* \in \operatorname{Argmax}_{i\in I} y'_i$ such that $y'_{i^*} \leq y_{i^*} - c$. Then*

$$E_{1+C/c}(y) \geq E_{1+C/c}(y') + c.$$

Proof. Let π be a permutation sorting y' in an increasing order[2] such that $\pi^{-1}(i^*) \geq \pi^{-1}(i)$ for all $i \in I$. Let $j = \pi^{-1}(i^*)$. Thus, in particular, we have $y_{\pi(i)} = y'_{\pi(i)}$ for $i > j$. Now

$$\begin{aligned}
&E_{1+C/c}(y,\pi) - E_{1+C/c}(y',\pi)\\
&= \sum_{i\in[m]} (y-y')_{\pi(i)}\,(1+C/c)^i\\
&\geq c\,(1+C/c)^j - \sum_{i=1}^{j-1} C\,(1+C/c)^i\\
&= c\,(1+C/c)^j - \frac{C\big((1+C/c)^j - 1\big)}{1+C/c-1}\\
&= c
\end{aligned}$$

where the first equality follows from the definition of the energy, using the assumed relations between y_i and y'_i. In particular, we used $y'_i \leq y_i + C$ for the first $j-1$ elements of $(y-y')_\pi$, and $y'_{i^*} \leq y_{i^*} - c$ for the jth one. In the second equality we used the formula for the sum of geometric series. Finally, omitting the subscripts for brevity, we have

$$E(y) \geq E(y,\pi) \geq E(y',\pi) + c = E(y') + c,$$

which finishes the proof. □

Existence of a strictly decreasing quantity shows that the method cannot get into a cycle. This is not yet enough for

[2] That is, $y'_{\pi(1)} \leq y'_{\pi(2)} \leq \cdots \leq y'_{\pi(m)}$. The inverse π^{-1} is then mapping original indices to their order.

convergence because, as we saw in Example 2.5, some components of the vector $Ax + b$ (and hence the energy) can decrease unboundedly. We currently do not know about any concise condition characterizing when this happens. Therefore, in the following theorem, the main result of the paper, we explicitly assume that the iterates are bounded during the algorithm which will be the case for MAP inference in §3.

Theorem 2.9. *Let Assumption 2.1 hold. Let c and C be given by (10). Let the energy $E_{1+C/c}(Ax + b)$ be bounded during Algorithm 1. For any $\varepsilon > 0$, the algorithm halts in $\mathcal{O}(1/\varepsilon)$ iterations. For $\varepsilon = 0$, we have $\lim_{t\to\infty} \eta^t = 0$ where η^t denotes the value of η after t outer iterations, and both x and $Ax + b$ converge.*

Proof. The energy is decreased in every step, and can decrease only by some constant M in total due to the boundedness. Within every outer iteration, there has to be a decrease in energy by at least $c\varepsilon$ by Proposition 2.7; otherwise the algorithm would terminate. Thus, there can be at most $M/(c\varepsilon) = \mathcal{O}(1/\varepsilon)$ non-terminating outer iterations before the energy is at its minimum.

For the convergence of iterates, we reiterate the previous argument. When x_j changes by ε, then the energy decreases by at least $c\varepsilon$. Thus, boundedness of energy implies the boundedness of $\sum_{t=1}^{\infty} \|x^t - x^{t+1}\|_\infty$ (where x^t denotes x after t inner iterations). So x cannot diverge nor, by Proposition 2.7, can it oscillate between multiple limit points; convergence of x implies convergence of $Ax + b$. □

We remark that the convergence of x or of $Ax + b$ does not follow from the fact that Algorithm 1 halts after at most $\mathcal{O}(1/\varepsilon)$ steps. Clearly, if the sequence (η_t) was, e.g., $(1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots)$, then the algorithm would halt after $1/\varepsilon$ steps but the sequences of x and $Ax + b$ would not necessarily converge.

3. MAP Inference

Now we turn our attention to the combinatorial optimization problem arising in MAP inference. It can be described as follows. We are given a set V of variables, each taking states from a finite label set L, and a set of weight functions, each depending on a (small) subset of the variables. We aim to maximize the sum of the functions over all the variables. For illustrative examples see, e.g., (Wainwright & Jordan, 2008; Savchynskyy, 2019; Kappes et al., 2015). For simplicity of presentation, we consider its pairwise version, when the functions can depend only on individual variables or pairs of variables. This problem reads[3]

$$F(\theta) = \max_{x\in L^V} \Big[\sum_{i\in V} \theta_{i,x_i} + \sum_{\{i,j\}\in E} \theta_{ij,x_i x_j} \Big] \tag{11}$$

[3]In this section, symbol x has a different meaning that in §2.

where (V, E) with $E \subseteq \binom{V}{2}$ is an undirected graph, $\theta_{i,x}$ ($i \in V$, $x \in L$) are unary weights, and $\theta_{ij,xy}$ ($\{i, j\} \in E$, $x, y \in L$) are binary weights (adopting that $\theta_{ij,xy} = \theta_{ji,yx}$). All the weights together form a vector $\theta \in \mathbb{R}^I$ where

$$I = (V \times L) \cup \{\, \{(i, x), (j, y)\} \mid \{i, j\} \in E,\ x, y \in L \,\}. \tag{12}$$

A labeling $x \in L^V$ assigns label $x_i \in L$ to each variable $i \in V$.

Example 3.1. Consider an image segmentation problem where we want to find a dark object on a bright background on a grayscale image. The set of variables V corresponds to the image pixels, and two vertices are connected by an edge $e \in E$ if the corresponding pixels are neighbouring. There are two labels in this task $L = \{\text{object}, \text{background}\}$. The unary weights might encode our prior knowledge that the object is dark, and the background is bright in the following way: $\theta_{i,\text{object}} = 1 - J(i)$ and $\theta_{i,\text{background}} = J(i)$, where $J(i)$ is the image intensity of pixel i. The pairwise weights may encode the fact that both the object and background are continuous, so we should expect neighbouring pixels to share label; then we might want $\theta_{ij,xy} = \mathbf{1}_{x=y}$.

3.1. Max-Sum Diffusion

It is well-known (see, e.g., (Werner, 2007)) that the objective function of (11) can be *reparameterized* by adding constants to some weights and subtracting the same constants from some other weights. The simplest such reparameterization acts on a single triplet $(i, j, x) \in P$, where

$$P = \{\, (i, j, x) \mid i \in V,\ j \in N_i,\ x \in L \,\} \tag{13}$$

and $N_i = \{\, j \in V \mid \{i, j\} \in E \,\}$, as follows: subtract a number ('message') $\delta_{ij,x}$ from $\theta_{i,x}$ and add the same number $\delta_{ij,x}$ to $\theta_{ij,xy}$ for each $y \in L$. Clearly, this preserves the objective of (11) because $\delta_{ij,x}$ cancels out in the sum. Composing these elementary reparameterization for all $(i, j, x) \in P$ changes the initial weight vector $\theta \in \mathbb{R}^I$ to the vector $\theta^\delta \in \mathbb{R}^I$ given by

$$\theta^\delta_{i,x} = \theta_{i,x} - \sum_{j\in N_i} \delta_{ij,x} \tag{14a}$$

$$\theta^\delta_{ij,xy} = \theta_{ij.xy} + \delta_{ij,x} + \delta_{ji,y} \tag{14b}$$

where all the messages $\delta_{ij,x}$ form a vector $\delta \in \mathbb{R}^P$.

Many LP-based MAP inference algorithms minimize a convex piecewise-affine upper bound on (11) over reparameterizations. We consider two such quantities

$$U_1(\theta) = \sum_{i\in V} \max_{x\in L} \theta_{i,x} + \sum_{\{i,j\}\in E} \max_{x,y\in L} \theta_{ij,xy},$$

$$U_2(\theta) = \max\Big\{ \max_{i\in V} \max_{x\in L} \theta_{i,x},\ \max_{\{i,j\}\in E} \max_{x,y\in L} \theta_{ij,xy} \Big\},$$

which clearly upper-bound (11) as

$$F(\theta) \leq U_1(\theta) \leq (|V| + |E|)\, U_2(\theta). \tag{15}$$

To obtain the best upper bound, we want to minimize $U_1(\theta^\delta)$ or $U_2(\theta^\delta)$ over δ. This can be formally obtained as a dual LP relaxation of (11). If the graph (V, E) is connected, at optimum we have $U_1(\theta^\delta) = (|V| + |E|)U_2(\theta^\delta)$, so these two relaxations are equivalent (Werner, 2007).

Arguably the simplest convex message passing method to minimize the above upper bound is known as *max-sum diffusion* (Kovalevsky & Koval, 1975; Schlesinger & Antoniuk, 2011; Werner, 2007). It has been formulated in several slightly different versions, we describe here a version that is monotonic in $U_2(\theta^\delta)$. Its update is as follows: pick a triplet $(i, j, x) \in P$ and change the variable $\delta_{ij,x}$ such that the equality

$$\theta^\delta_{i,x} = \max_{y \in L} \theta^\delta_{ij,xy} \tag{16}$$

becomes satisfied. Due to (14), this is done by setting

$$\delta_{ij,x} \leftarrow \delta_{ij,x} + \tfrac{1}{2}\big(\theta^\delta_{i,x} - \max_{y \in L} \theta^\delta_{ij,xy}\big). \tag{17}$$

Any point satisfying (16) for all $(i, j, x) \in P$ is a *fixed point* of max-sum diffusion.

Algorithm 2 Max-sum diffusion

Input: weights θ, initial point δ, precision $\varepsilon \geq 0$
$\eta \leftarrow \infty$
while $\eta \geq \varepsilon$ **do**
  $\eta \leftarrow 0$
  for $(i, j, x) \in P$ **do**
    $d \leftarrow \frac{1}{2}\big(\theta^\delta_{i,x} - \max_{y \in L} \theta^\delta_{ij,xy}\big)$
    $\delta_{ij,x} \leftarrow \delta_{ij,x} + d$
    $\eta \leftarrow \max\{|d|, \eta\}$
  end for
end while

Max-sum diffusion is a particular case of the method from §2. The function $\delta \mapsto U_2(\theta^\delta)$ has the form (1), being the pointwise maximum of affine functions (14) with coefficients in $\{-1, 0, +1\}$ (thus we have $c = C = 1$ in (10)) and offsets θ. The max-sum diffusion update on triplet $(i, j, x) \in P$ seeks to minimize the maximum over variable $\delta_{ij,x}$ of only those affine functions that depend on variable $\delta_{ij,x}$, i.e., the functions $\theta^\delta_{i,x}$ and $\{\theta^\delta_{ij,xy}\}_{y \in L}$, keeping the other variables fixed. For any $d \in \mathbb{R}$, we have

$$\theta^{\delta + d e_{ij,x}}_{i,x} = \theta^\delta_{i,x} \quad - d \tag{18a}$$

$$\theta^{\delta + d e_{ij,x}}_{ij,xy} = \theta^\delta_{ij,xy} + d \qquad \forall y \in L \tag{18b}$$

where $e_{ij,x}$ is the (i, j, x)-th standard basis vector of $\mathbb{R}^P$. Minimizing the maximum of functions (18) over d yields $d = \frac{1}{2}\big(\theta^\delta_{i,x} - \max_{y \in L} \theta^\delta_{ij,xy}\big)$.

To meet the assumptions of Theorem 2.9, we show that the weights remain bounded during max-sum diffusion.

Lemma 3.2. *Every element of the weight vector θ^δ is bounded from below during Algorithm 2.*

Proof. Consider the expression

$$\sum_{(i,j,x) \in P} \Big(|L| \theta^\delta_{i,x} + \sum_{y \in L} \theta^\delta_{ij,xy}\Big).$$

It does not depend on any $\delta_{ij,x}$ as they cancel out. Moreover, it is a non-negative combination of the elements of θ^δ. Since the maximum $U_2(\theta^\delta)$ of these elements is bounded from above, each element of θ^δ has to be bounded also from below. □

Now we are ready to state the convergence result:

Theorem 3.3. *For any $\varepsilon > 0$, Algorithm 2 terminates within $\mathcal{O}(1/\varepsilon)$ steps. For $\varepsilon = 0$, vectors δ converge to a max-sum diffusion fixed point, given by (16).*

Proof. Consequence of Lemma 3.2 and Theorem 2.9. □

3.2. Max-marginal Averaging

A powerful approach to construct convex bounds on combinatorial optimization problems is Lagrangian decomposition (Guignard & Kim, 1987). For MAP inference, it was applied by (Johnson et al., 2007) and popularized by (Komodakis et al., 2011). It also underlies the tree-decomposition methods such as TRW-S (Kolmogorov, 2006).

Many combinatorial optimization problems can be written in the general form

$$F(\theta) = \max_{\mu \in M} \langle \theta, \mu \rangle \tag{19}$$

where $\theta \in \mathbb{R}^I$ are weights, $M \subseteq \{0, 1\}^I$ is the feasible set (combinatorially large), I is a finite set of 'features', and $\langle \cdot, \cdot \rangle$ denotes the dot product. The MAP inference problem (11) is obtained as a special case for I given by (12), $M = \phi(L^V) = \{\, \phi(x) \mid x \in L^V \,\}$, and the 'feature map' $\phi\colon L^V \to \{0, 1\}^I$ being such that the function $\langle \theta, \phi(x) \rangle$ coincides with the objective function of (11). The convex hull of M is known as the *marginal polytope* (Wainwright & Jordan, 2008).

An upper bound on (19) is constructed by decomposition to subproblems[4]. Let S denote the set of subproblems and $\theta_s \in \mathbb{R}^I$ the weight vector of subproblem $s \in S$. These

[4] To see how the bound (21) arises from Lagrangian decomposition, write problem (19) as minimization of $\sum_{s \in S} \langle \theta_s, \mu_s \rangle$ subject to $\mu_s = \mu$ and $\mu, \mu_s \in M$, and then dualize the coupling constraints $\mu_s = \mu$.

weights satisfy

$$\sum_{s\in S}\theta_s = \theta, \tag{20a}$$

$$\theta_{s,i} = 0 \qquad \forall s \in S,\ i \in I \setminus I_s, \tag{20b}$$

where $\theta_{s,i}$ denotes the ith component of vector θ_s and each set $I_s \subseteq I$ is such that the function $F(\theta_s)$ is tractable to compute (e.g., each I_s defines a acyclic subproblem). By swapping max and sum in (19), we obtain two upper bounds (analogically to (15))

$$F(\theta) = F\Big(\sum_{s\in S}\theta_s\Big) \le \sum_{s\in S} F(\theta_s) \le |S| \max_{s\in S} F(\theta_s). \tag{21}$$

We want to minimize one of the upper bounds in (21) over the variables $(\theta_s)_{s\in S}$ subject to (20).

For I and ϕ defined by (11) and natural choices of sets I_s (e.g., the rows and columns of an image), the numbers $F(\theta_s)$ can always be made equal for all $s \in S$ while keeping (20). Hence the two upper bounds in (21) coincide at optimum. Thus, we can focus only on the second bound.

Minimization of the upper bound in Lagrangian decomposition is traditionally done by subgradient methods, which for the MAP inference problem was applied by (Komodakis et al., 2011). An alternative but less common approach is so-called *max-marginal averaging*, for MAP inference first proposed by (Kolmogorov, 2006; Johnson et al., 2007). It can be seen as a block-coordinate descent, where in each iteration we pick some $i \in I$ and minimize the upper bound over the variables $(\theta_{s,i})_{s\in S}$ subject to (20). We shall describe here its slightly different version (with similar convergent properties), in which only a *pair* of max-marginals is averaged in each update.

The *max-marginal* of the function $\langle\theta, \mu\rangle$ associated with a feature $i \in I$ is the number

$$F_i(\theta) = \max_{\mu\in M:\, \mu_i=1} \langle\theta, \mu\rangle. \tag{22}$$

Note that F_i depends on θ_i linearly: for any $d \in \mathbb{R}$ we have

$$F_i(\theta + de_i) = F_i(\theta) + d \tag{23}$$

where $e_i \in \mathbb{R}^I$ denotes the ith standard basis vector of $\mathbb{R}^I$.

Suppose that for each feature $i \in I$ we have chosen a set $E_i \subseteq S_i \times S_i$ where $S_i = \{\, s \in S \mid i \in I_s \,\}$. One update of pairwise max-marginal averaging proceeds as follows: pick some $i \in I$ and $(s,t) \in E_i$, and change the variables $\theta_{s,i}$ and $\theta_{t,i}$ to enforce $F_i(\theta_s) = F_i(\theta_t)$. To maintain (20a), due to (23) this can be done by adding a number d to $\theta_{s,i}$ and subtracting the same number from $\theta_{t,i}$. Clearly, such number is uniquely given by[5]

$$d = \tfrac{1}{2}\big(F_i(\theta_t) - F_i(\theta_s)\big). \tag{24}$$

[5]Computing the involved max-marginals from scratch before every update would be costly in practice. However, it is often possible to reuse partial results from computing earlier max-marginals, which can increase efficiency significantly. Compare, e.g., the algorithms in Figures 1 and 2 in (Kolmogorov, 2006).

This update in fact minimizes $\max\{F_i(\theta_s), F_i(\theta_t)\}$ over $\theta_{s,i}$ and $\theta_{t,i}$ subject to (20).

To apply our results from §2, we need to reformulate the upper bound minimization as an *unconstrained* minimization of the maximum of affine functions. For that, it suffices to parameterize the affine subspace defined by (20). If a vector $(\theta_s)_{s\in S}$ satisfies (20), then the vector $(\theta^\delta_s)_{s\in S}$ given by

$$\theta^\delta_{s,i} = \theta_{s,i} + \sum_{t|(s,t)\in S_i} \delta_{st,i} - \sum_{t|(t,s)\in S_i} \delta_{ts,i} \tag{25}$$

also satisfies (20) for any 'messages' $\delta_{st,i} \in \mathbb{R}$. For instance, if $S_i = \{1,2,3,5\}$ and $E_i = \{(1,2),(2,3),(3,5)\}$, then (25) reads

$$\begin{aligned}
\theta^\delta_{1,i} &= \theta_{1,i} + \delta_{12,i} \\
\theta^\delta_{2,i} &= \theta_{2,i} - \delta_{12,i} + \delta_{23,i} \\
\theta^\delta_{3,i} &= \theta_{3,i} - \delta_{23,i} + \delta_{35,i} \\
\theta^\delta_{5,i} &= \theta_{5,i} - \delta_{35,i}
\end{aligned}$$

Clearly $\sum_{s\in S_i}\theta^\delta_{s,i} = \sum_{s\in S_i}\theta_{s,i}$ because $\delta_{st,i}$ cancel out. Note, (25) is a counterpart of (14). If the digraph (S_i, E_i) is connected and its edges E_i cover S_i, then *any* vector $(\theta^\delta_s)_{s\in S}$ satisfying (20) can be parameterized as (25).

Algorithm 3 Max-marginal averaging

Input: weights $(\theta_s)_{s\in S}$, initial point δ, precision $\varepsilon \ge 0$
$\eta \leftarrow \infty$
while $\eta \ge \varepsilon$ **do**
  $\eta \leftarrow 0$
  for $i \in I$ **do**
    for $(s,t) \in E_i$ **do**
      $d \leftarrow \frac{1}{2}\big(F_i(\theta^\delta_t) - F_i(\theta^\delta_s)\big)$
      $\delta_{st,i} \leftarrow \delta_{st,i} + d$
      $\eta \leftarrow \max\{|d|, \eta\}$
    end for
  end for
end while

Now, the upper bound minimization reads as unconstrained minimization of the function $\max_{s\in S} F(\theta^\delta_s)$ over δ and pairwise max-marginal averaging becomes Algorithm 3. The fixed point of the method is any point satisfying $F_i(\theta^\delta_s) = F_i(\theta^\delta_t)$ for all $i \in I$ and $(s,t) \in S_i$.

This is a particular case of the method from §2: the objective function has the form (1), where the affine functions $x \mapsto a_i^T x + b_i$ correspond to functions $\delta \mapsto \langle\theta^\delta_s, \phi(x)\rangle$. One inner iteration minimizes the maximum of only those affine

functions that depend on variable $\delta_{st,i}$. Since μ has non-negative components, it follows from (25) that whenever a message $\delta_{st,i}$ changes, some of these functions increase and some decrease, which verifies condition (2).

Lemma 3.4. *The numbers $\langle \theta^s, \mu \rangle$ for any $\mu \in M$ and $s \in S$ are bounded from below during Algorithm 3.*

Proof. As the objective $\max_{s \in S} F(\theta^s)$ never increases during the algorithm, the numbers $\langle \theta^s, \mu \rangle$ are bounded above. Hence, due to the identity $\langle \theta, \mu \rangle = \sum_{s \in S} \langle \theta^s, \mu \rangle$, they are also bounded below because the LHS is constant. □

Theorem 3.5. *For any $\varepsilon > 0$, Algorithm 3 terminates in $\mathcal{O}(1/\varepsilon)$ steps. For $\varepsilon = 0$, the vectors θ^δ_s converge to a fixed point of max-marginal averaging.*

Proof. Consequence of Lemma 3.4 and Theorem 2.9 □

4. Mid-point Rule in Coordinate Descent

When applying BCD to any (possibly non-smooth and/or constrained) convex problem, (Werner et al., 2020) proposed that if block-minimizers are non-unique we should always choose a minimizer from the relative interior of the block-minimizer set. They proved that this *relative interior rule* is not worse, in a precise sense, than any other rule to choose block-minimizers.

The rule we proposed in §2 for unconstrained minimization of function (1), namely to ignore the affine functions that do not depend on the current variable, is a special case of the relative interior rule. Indeed, the chosen minimizer of the univariate function $x_j \mapsto f(x)$ is always in the interior of the set of minimizers (which is an interval or a singleton), see Example 2.3. However, we could choose any other point inside the interval. A natural way is to choose the middle point of the interval. We call this the *mid-point rule*. For unconstrained minimization of a function (1) with coefficients $A \in \{-1, 0, +1\}^{m \times n}$ these two rules clearly coincide (as in Example 2.3), but for more general coefficients they do not.[6] Moreover, unlike the rule from §2, the mid-point rule is applicable (assuming that the coordinate minimizer sets are closed intervals or singletons) even for constrained convex problems, such as linear programs.

Further in this section, we show that coordinate descent with the mid-point rule can cycle when applied to a problem other than unconstrained minimization of function (1) with coefficients in $\{-1, 0, +1\}$.

[6] Interestingly, the 'middle-point algorithm' proposed by (Wedelin, 1995) for dual LP relaxation of the set cover problem is essentially the same as our algorithm from §2 for this particular problem. He further argued (Wedelin, 2013) that when applied to MAP inference, this algorithm is equivalent to max-sum diffusion. In both cases, the coefficeent are indeed in $\{-1, 0, +1\}$.

Proposition 4.1. *Coordinate descent with the mid-point rule can cycle.*

Proof. We start with a constrained problem. Let $M \subseteq \mathbb{R}^3$ denote the set formed by the 12 points in the first two columns of the following table:

$$\begin{array}{cc|c} (-2,0,0) & (2,0,0) & (\underline{0},0,0) \\ (0,-1,0) & (0,3,0) & (0,\underline{1},0) \\ (0,1,-1) & (0,1,3) & (0,1,\underline{1}) \\ (-1,1,1) & (3,1,1) & (\underline{1},1,1) \\ (1,-2,1) & (1,2,1) & (1,\underline{0},1) \\ (1,0,-2) & (1,0,2) & (1,0,\underline{0}) \end{array} \tag{26}$$

We will later show that each point from M is an extremal point of the convex hull of M, denoted by $\operatorname{conv} M$. Consider the problem of minimizing the constant (e.g., zero) objective on the polytope $\operatorname{conv} M$, and apply to it coordinate descent with the mid-point rule. Thus, whenever we want to find the mid-point w.r.t. a coordinate and the current iterate equals to two points $x, y \in M$ in the other two coordinates, then the next iterate will be their mid-point, $(x + y)/2$. Now we can see that starting from the point $(0, 0, 0)$ and updating the coordinates in cyclic order, we obtain the iterates in the third column of the table, where the current coordinate is always underlined. In particular, when we take an iterate from row i, at two positions (over which we are not updating) it is equal to the points from the first two columns in row $i + 1$, and the third column of row $i + 1$ is the average of the previous two points, indices are $\bmod 6$. After six updates, we return the initial point $(0, 0, 0)$, i.e., the method enters a loop.

Let us show that all points in M are indeed extreme points of $\operatorname{conv} M$. We do this by presenting a supporting hyperplane of $\operatorname{conv} M$ for each point $x \in M$ that passes only through x; thus, x is not a convex combination of any two points from M. The hyperplanes (actually halfspaces) are as follows:

$$\begin{array}{rr} -x_1 \le 2 & 8x_1 - 5x_2 - 3.5x_3 \le 16 \\ -3.5x_1 - 8x_2 - 5x_3 \le 8 & x_2 \le 3 \\ -5x_1 + 3.5x_2 - 8x_3 \le 11.5 & x_3 \le 3 \\ -8x_1 + 5x_2 + 3.5x_3 \le 16.5 & x_1 \le 3 \\ -x_2 \le 2 & 3.5x_1 + 8x_2 + 5x_3 \le 24.5 \\ -x_3 \le 2 & 5x_1 - 3.5x_2 + 8x_3 \le 21 \end{array} \tag{27}$$

Having shown cycling for a constrained problem, we proceed to demonstrate it for an unonstrained minimization of a function in the form (1). Consider the function

$$f(x) = \max\{0, \max_i (Ax - b)_i\} \tag{28}$$

where $Ax \le b$ denotes the system on linear inequalities (27). It is not hard to see that coordinate descent with the mid-point rule applied to unconstrained minimization of function (28) will produce the same iterates as in table (26). □

5. Conclusion

In this paper, we first studied coordinate descent method applied to unconstrained minimization of the pointwise maximum of affine functions with sparse coefficients. This function is non-smooth and can have non-unique coordinate-minimizers, hence it is hard for coordinate descent. We considered the version of coordinate descent that resolves this ambiguity by ignoring the affine functions not depending on the current variable, which (under the technical assumption (2)) ensures uniqueness of coordinate-minimizers.

As our central result, we proved convergence of this method to its fixed point and, more precisely, showed that the method achieves precision $\varepsilon > 0$ in $\mathcal{O}(1/\varepsilon)$ iterations. We did this by designing a novel energy function that strictly decreases with every iteration.

Let us remark that the asymptotic upper bound $\mathcal{O}(1/\varepsilon)$ involves a big constant, which depends exponentially on the problem size. Currently we are not sure if this is only an artifact of our analysis or it is inherent to the method. In experiment, we have never observed an exponential dependence of convergence rate on instance size.

Then we showed that two popular representants of dual coordinate descent algorithms (max-sum diffusion and max-marginal averaging) to minimize LP/Lagrangian upper bound on the MAP inference problem are special cases of the above simple method. This proves the long-standing conjecture that these algorithms converge to a fixed point.

Although we presented the convergence results only for max-sum diffusion and marginal-averaging, we believe the proof straightforwardly extends to a number of other dual (block-)coordinate descent methods to minimize an LP/Lagrangian upper bound on combinatorial optimization problems. An exception is, e.g., SRMP (Kolmogorov, 2015), which by design does not converge to a fixed point but only to a locally consistent set.

Finally, we showed that a slightly modified version of coordinate descent, in which the coordinate-wise minimizers are chosen as the midpoints of the intervals, can cycle. This provides a further insight to the behaviour of (block-)coordinate descent on non-smooth and/or constrained problems, which is still relatively poorly understood.

Acknowledgements

VV was supported by the DFG Cluster of Excellence "Machine Learning – New Perspectives for Science", EXC 2064/1, project number 390727645 and is thankful for the support of Open Philanthropy. TW thanks the CTU institutional support (future fund).

Impact Statement

We are not aware of any.

References

Abbas, A. and Swoboda, P. Fastdog: Fast discrete optimization on gpu. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 439–449, 2022.

Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

Chen, L.-C., Schwing, A., Yuille, A., and Urtasun, R. Learning deep structured models. In *Intl. Conf. on Machine Learning (ICML)*, pp. 1785–1794, 2015.

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

Globerson, A. and Jaakkola, T. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Neural Information Processing Systems*, pp. 553–560, 2008.

Guignard, M. and Kim, S. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming*, 39:215–228, 1987.

Johnson, J. K., Malioutov, D. M., and Willsky, A. S. Lagrangian relaxation for MAP estimation in graphical models. In *45th Allerton Conference on Communication, Control and Computing*, 2007.

Kappes, J. H., Andres, B., Hamprecht, F. A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B. X., Kröger, T., Lellmann, J., Komodakis, N., Savchynskyy, B., and Rother, C. A comparative study of modern inference techniques for structured discrete energy minimization problems. *Intl. J. of Computer Vision*, 115(2):155–184, 2015.

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

Kolmogorov, V. A new look at reweighted message passing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37(5), May 2015.

Komodakis, N., Paragios, N., and Tziritas, G. MRF energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):531–552, 2011.

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

Lange, J. and Swoboda, P. Efficient message passing for 0-1 ILPs with binary decision diagrams. In *Intl. Conf. on Machine Learning (ICML)*, volume 139, pp. 6000–6010. PMLR, 2021.

Martins, A. L., Figueiredo, M. A. T., Aguiar, P. M. Q., Smith, N. A., and Xing, E. P. An augmented Lagrangian approach to constrained MAP inference. In *Intl. Conf. on Machine Learning*, pp. 169–176, 2011.

Meltzer, T., Globerson, A., and Weiss, Y. Convergent message passing algorithms: a unifying view. In *Conf. on Uncertainty in Artificial Intelligence*, pp. 393–401, 2009.

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

Munda, G., Shekhovtsov, A., Knöbelreiter, P., and Pock, T. Scalable full flow with learned binary descriptors. In *German Conf. on Pattern Recognition (GCPR)*, 2017.

Ruozzi, N. and Tatikonda, S. Message-passing algorithms: Reparameterizations and splittings. *IEEE Transactions on Information Theory*, 59(9):5860–5881, 2013.

Savchynskyy, B. A bundle approach to efficient MAP-inference by Lagrangian relaxation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1688–1695, 2012.

Savchynskyy, B. Discrete graphical models – an optimization perspective. *Foundations and Trends in Computer Graphics and Vision*, 11(3-4):160–429, 2019. ISSN 1572-2740.

Savchynskyy, B., Schmidt, S., Kappes, J. H., and Schn orr, C. Efficient MRF energy minimization via adaptive diminishing smoothing. In *Conf. on Uncertainty in Artificial Intelligence*, pp. 746–755, 2012.

Schlesinger, M. I. and Antoniuk, K. Diffusion algorithms and structural recognition optimization problems. *Cybernetics and Systems Analysis*, 47:175–192, 2011. ISSN 1060-0396.

Sontag, D., Globerson, A., and Jaakkola, T. Introduction to dual decomposition for inference. In Sra, S., Nowozin, S., and Wright, S. J. (eds.), *Optimization for Machine Learning*. MIT Press, 2012.

Swoboda, P. and Andres, B. A message passing algorithm for the minimum cost multicut problem. In *Conf. on Computer Vision and Pattern Recognition*, 2017.

Swoboda, P., Kuske, J., and Savchynskyy, B. A dual ascent framework for Lagrangean decomposition of combinatorial problems. In *Conf. on Computer Vision and Pattern Recognition*, 2017a.

Swoboda, P., Rother, C., Abu Alhaija, H., Kainmuller, D., and Savchynskyy, B. A study of Lagrangean decompositions and dual ascent solvers for graph matching. In *Conf. on Computer Vision and Pattern Recognition*, 2017b.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwal, A., Tappen, M., and Rother, C. A comparative study of energy minimization methods for Markov random fields. In *Eur. Conf. on Computer Vision*, pp. II: 16–29, 2006.

Tourani, S., Shekhovtsov, A., Rother, C., and Savchynskyy, B. Mplp++: Fast, parallel dual block-coordinate ascent for dense graphical models. In *The European Conference on Computer Vision (ECCV)*, 2018.

Tourani, S., Shekhovtsov, A., Rother, C., and Savchynskyy, B. Taxonomy of dual block-coordinate ascent methods for discrete energy minimization. In *Intl. Conf. on Artificial Intelligence and Statistics*, pp. 2775–2785, 2020.

Tseng, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, June 2001.

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

Warga, J. Minimizing certain convex functions. *Journal of the Society for Industrial and Applied Mathematics*, 11 (3):588–593, 1963.

Wedelin, D. An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Ann. Oper. Res.*, 57(1):283–301, 1995.

Wedelin, D. Revisiting the in-the-middle algorithm and heuristic for integer programming and the max-sum problem. Chalmers University preprint, https://research.chalmers.se/en/publication/190782, 2013.

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

Werner, T. Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(8):1474–1488, August 2010.

Werner, T. On coordinate minimization of piecewise-affine functions. Technical Report CTU-CMP-2017-05, Dept. of Cybernetics, Fac. of Electrical Eng., Czech Technical Univ. in Prague, September 2017.

Werner, T., Průša, D., and Dlask, T. Relative interior rule in block-coordinate descent. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020.