A Study of Nesterov's Scheme for Lagrangian Decomposition and MAP Labeling

Bogdan Savchynskyy[1]
bogdan.savchynskyy@iwr.uni-heidelberg.de

Stefan Schmidt[1]
schmidt@math.uni-heidelberg.de

Jörg Kappes[2]
kappes@math.uni-heidelberg.de

Christoph Schnörr[1,2]
schnoerr@math.uni-heidelberg.de

[1]HCI and [2]IPA, Heidelberg University, Germany

Abstract

We study the MAP-labeling problem for graphical models by optimizing a dual problem obtained by Lagrangian decomposition. In this paper, we focus specifically on Nesterov's optimal first-order optimization scheme for non-smooth convex programs, that has been studied for a range of other problems in computer vision and machine learning in recent years. We show that in order to obtain an efficiently convergent iteration, this approach should be augmented with a dynamic estimation of a corresponding Lipschitz constant, leading to a runtime complexity of $O(\frac{1}{\epsilon})$ in terms of the desired precision ϵ. Additionally, we devise a stopping criterion based on a duality gap as a sound basis for competitive comparison and show how to compute it efficiently. We evaluate our results using the publicly available Middlebury database and a set of computer generated graphical models that highlight specific aspects, along with other state-of-the-art methods for MAP-inference.

1. Introduction

Problem We consider the problem of computing the most likely configuration x for a given graphical model, i.e. a distribution $p_{\mathcal{G}}(x;\theta) \propto \exp(-E_{\mathcal{G}}(\theta,x))$. We use the following standard notation [21]:

Let $\mathcal{G} = (\mathcal{V},\mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is a finite set of its nodes and $\mathcal{E} \subset \mathcal{V}\times\mathcal{V}$ is a set of edges. Let also $\mathcal{X}_v,\ v\in\mathcal{V}$ be a finite *set of labels*. The set $\mathcal{X} = \otimes_{v\in\mathcal{V}}\mathcal{X}_v$, where $\otimes$ denotes the Cartesian product, will be called *labeling set* and its elements $x\in\mathcal{X}$ *labelings*. Thus each labeling is a collection $(x_v\colon v\in\mathcal{V})$ of labels. To shorten notation we will use x_{uv} for a pair of labels (x_u,x_v) and $\mathcal{X}_{uv}$ for $\mathcal{X}_u\times\mathcal{X}_v$. Functions of the form $\theta_v:\mathcal{X}_v\to\mathbb{R},\ v\in\mathcal{V}$, and $\theta_{uv}:\mathcal{X}_{uv}\to\mathbb{R},\ uv\in\mathcal{E}$, are called *unary* and *pairwise potentials*, respectively. The collection of all potentials will be denoted by θ.

The problem to compute the most likely labeling x (*MAP labeling problem*) amounts to minimizing the energy function

$$\min_{x\in\mathcal{X}} E_{\mathcal{G}}(\theta,x) = \min_{x\in\mathcal{X}}\left\{\sum_{v\in\mathcal{V}}\theta_v(x_v) + \sum_{uv\in\mathcal{E}}\theta_{uv}(x_{uv})\right\}. \tag{1}$$

Background and Motivation Problem (1) is known to be NP-complete in general for graphs with cycles. We will concentrate mainly on the linear programming (LP) relaxation of the problem originally proposed by Schlesinger [17] – see [22] for a recent review.

Schlesinger [17] analysed also the dual LP as an upper bound of an integer solution and proposed two minimization algorithms: the DAG algorithm and a diffusion algorithm (cf. [22]). These algorithms decrease the value of the dual LP monotonically but do not attain its optima in general, since they can be interpreted as (block-)coordinate descent and thus can get stuck due to the non-smoothness of the dual objective.

Another algorithm, known as TRW-S, was proposed by Kolmogorov [8]. This algorithm computes the same fixed points as the diffusion algorithm and generalizes it by considering arbitrary sub-trees of the initial graph as elementary subproblems in contrast to separate nodes and neighboring edges in the diffusion. An alternative sub-gradient based scheme for dual function minimization was proposed independently by Schlesinger [18] and Komodakis [10]. Such sub-gradient iterations are guaranteed to compute the optimum of the dual function but have two drawbacks: i) no efficient convergence rate is backed by theory – to the best of our knowledge no improvement has been established with respect to the general convergence estimate $O(\frac{1}{\epsilon^2})$ [13] (which specifies up to a constant the maximal number of iterations required to achieve a desired precision ϵ) and ii) absence of a stopping criterion that is sound from the optimization viewpoint.

Disadvantages of both approaches (TRW-S and subgradient iteration) are mainly caused by the non-smoothness of the dual objective. To overcome this problem, smoothing of the objective was proposed in a series of papers [5, 6, 15, 23]. However, questions concerning the worst-case complexity bound and theoretically sound stopping conditions have remained open.

In a most recent work [7], smoothing of the dual objective was addressed with Nesterov's optimal first-order optimization scheme. We will show in this paper, however, that without carefully modifying the generic scheme [13], the resulting complexity bound $O(\sqrt{|\mathcal{V}|}/\epsilon)$ is too loose for almost any real problem instance.

Other competitive optimization schemes have been published recently [2, 4]. Their investigation in the same context is beyond the scope of this paper.

Contribution Our contribution is two-fold:

(i) We propose an algorithm for solving the dual LP problem with a guaranteed complexity of $O(\frac{1}{\epsilon})$ oracle calls (evaluations of the function or its gradient).

(ii) We formulate and analyse a general method for constructing an upper bound for algorithms maximizing the dual LP objective. The method is used in turn to devise a sound stopping criterion based on the duality gap.

Algorithm (i) is based on smoothing the dual objective and applying the optimal first-order optimization scheme by Nesterov [14]. Our approach is similar to the method described in [7] but differs from it in essential technical details:

(a) Instead of using a *fixed* Lipschitz constant for a gradient step of the algorithm, we *adaptively* estimate this constant during the iteration. This leads to a significantly smaller number of outer iterations of the algorithm necessary for convergence, at the cost of a few more oracle calls (no more than 4 on average) in the inner loop of the iteration. Overall, our algorithm is much faster.

(b) Instead of *static* selection of a smoothing value, we select it *dynamically*, which usually gives a significant speed-up.

Method (ii) can be applied to any iterative scheme as soon as an approximate, but not necessarily feasible, primal solution can be computed. Hence, this contribution should be of wider interest. We use this method to define and evaluate a stopping condition for our algorithm. In contrast, no stopping criterion was specified in [7].

For the sake of clarity of our presentation, we consider here the special case of grid-graphs $\mathcal{G}$, mainly because the benchmark [19] conforms to this setting. Although none of our results is restricted to this special case, the quantitative evaluation is, of course. A generalization is mostly straightforward, and we add specific comments where issues might arise. All proofs of theoretical results are available as supplementary material [1], due to the space restriction.

2. Description of the Algorithm

Decomposition and Relaxation Our approach is based on the dual decomposition framework which was proposed for energy minimization by [20] and later on analysed by [10]. Let $\mathcal{G}^i = (\mathcal{V}^i, \mathcal{E}^i),\ i = 1, 2$, be two *acyclic* subgraphs of the *master graph* $\mathcal{G}$. Let $\mathcal{V}^1 = \mathcal{V}^2 = \mathcal{V}$, $\mathcal{E}^1 \bigcup \mathcal{E}^2 = \mathcal{E}$ and $\mathcal{E}^1 \bigcap \mathcal{E}^2 = \emptyset$ (e.g., $\mathcal{E}^1$ may contain all horizontal edges of $\mathcal{G}$ and $\mathcal{E}^2$ – all vertical ones if $\mathcal{G}$ is a grid graph). Then the overall energy becomes the sum of the energies corresponding to these sub-graphs,

$$\begin{aligned} E_{\mathcal{G}}(\theta, x) &= \sum_{i=1}^{2} \sum_{v \in \mathcal{V}^i} \theta_v^i(x_v) + \sum_{uv \in \mathcal{E}^i} \theta_{uv}^i(x_{uv}) \\ &= E_{\mathcal{G}^1}(\theta^1, x) + E_{\mathcal{G}^2}(\theta^2, x), \end{aligned} \qquad (2)$$

provided $\theta_{uv}^i = \theta_{uv},\ uv \in \mathcal{E}^i,\ i = 1, 2$ and $\theta_v^1(x_v) + \theta_v^2(x_v) = \theta_v(x_v),\ \forall v \in \mathcal{V}, x_v \in \mathcal{X}_v$. The latter condition can be represented in a parametric way as $\theta_v^1(x_v) = \frac{\theta_v(x_v)}{2} + \lambda_v(x_v)$ and $\theta_v^2(x_v) = \frac{\theta_v(x_v)}{2} - \lambda_v(x_v), v \in \mathcal{V},\ x_v \in \mathcal{X}_v$, where $\lambda_v(x_v) \in \mathbb{R}$. Thus we consider θ^i as a function of λ and obviously have

$$\min_{x \in \mathcal{X}} E_{\mathcal{G}}(\theta, x) \geq \max_{\lambda} \sum_{i=1}^{2} \min_{x \in \mathcal{X}} E_{\mathcal{G}^i}(\theta^i(\lambda), x). \qquad (3)$$

It is well-known [9] that all collections (of arbitrary cardinality) of acyclic sub-graphs covering the master graph are equivalent, in the sense that they lead to the same lower bound as the one presented on the right-hand side of equation (3). It is also well-known that this lower bound is equal to the solution of the following linear programming problem:

$$\begin{aligned} \min_{\mu} \sum_{v \in \mathcal{V}} \sum_{x_v \in \mathcal{X}_v} & \theta_v(x_v)\mu_v(x_v) \qquad (4) \\ & + \sum_{uv \in \mathcal{E}} \sum_{x_{uv} \in \mathcal{X}_{uv}} \theta_{uv}(x_{uv})\mu_{uv}(x_{uv}) \end{aligned}$$

$$\text{s.t.} \quad \begin{array}{l} \sum_{x_v \in \mathcal{V}} \mu_v(x_v) = 1,\ v \in \mathcal{V} \\ \sum_{x_v \in \mathcal{V}} \mu_{uv}(x_{uv}) = \mu_u(x_u),\ x_u \in \mathcal{X}_u,\ uv \in \mathcal{E} \\ \sum_{x_u \in \mathcal{V}} \mu_{uv}(x_{uv}) = \mu_v(x_v),\ x_v \in \mathcal{X}_v,\ uv \in \mathcal{E} \\ \mu_{uv}(x_{uv}) \geq 0,\ x_{uv} \in \mathcal{X}_{uv},\ uv \in \mathcal{E}\,. \end{array} \qquad (5)$$

This formulation is based on the overcomplete representation commonly used for discrete graphical models [21], in terms of relaxed indicator vectors μ constrained to the *local polytope* $\mathcal{L}(\mathcal{G})$, that is defined by the constraints of (4).

[1] http://hci.iwr.uni-heidelberg.de/publications/mip/techrep/ /savchynskyy_11_study_supplement.pdf

It is well-known that $\mathcal{L}(\mathcal{G})$ constitutes an outer bound (relaxation) of the convex hull of all indicator vectors of labelings (marginal polytope; cf. [21]). Consequently, (4) simply reads $\min_{\mu \in \mathcal{L}(\mathcal{G})} \langle \theta, \mu \rangle$.

Problem Smoothing Consider a single summand on the right-hand side of (3). It can be expressed as inner product of the local potential vector θ^i and a correspondingly chosen binary indicator vector $\phi(x)$:

$$U^i(\lambda) := \min_{x \in \mathcal{X}} E_{\mathcal{G}^i}(\theta^i(\lambda), x) = \min_{x \in \mathcal{X}} \left\langle \theta^i(\lambda), \phi(x) \right\rangle . \quad (6)$$

Since this is a non-smooth function, the objective – right-hand side of (3) – is also non-smooth. Applying the well-known approximation of the $\min$ (or $-\max$) function by the log-exponential function (cf. [14, 16]) leads to the smooth version

$$\hat{U}^i_\rho(\lambda) = -\rho \log \sum_{x \in \mathcal{X}} \exp \left\langle -\theta^i(\lambda)/\rho, \phi(x) \right\rangle \quad (7)$$

with *smoothing parameter* ρ, that uniformly approximates U^i, that is

$$\hat{U}^i_\rho(\lambda) \le U^i(\lambda) \le \hat{U}^i_\rho(\lambda) + \rho \log |\mathcal{X}|. \quad (8)$$

Thus, for $\hat{U}_\rho = \sum_{i=1}^2 \hat{U}^i_\rho$ and $U = \sum_{i=1}^2 U^i$,

$$\hat{U}_\rho(\lambda) \le U \le \hat{U}_\rho(\lambda) + 2\rho \log |\mathcal{X}|. \quad (9)$$

We will call a gradient ∇f of a differentiable function $f \colon \mathbb{R}^n \to \mathbb{R}$ *Lipschitz continuous* with *Lipschitz constant* L if

$$\|\nabla f(z) - \nabla f(w)\| \le L\|z - w\|, \; \forall z, w \in \mathbb{R}^n , \quad (10)$$

where $\|\cdot\|$ is the ℓ_2-norm in $\mathbb{R}^n$.

Defining vectors $D\hat{U}^i_\rho(\lambda) \in \mathbb{R}^{\sum_{v \in \mathcal{V}} |\mathcal{X}_v|}$ by

$$D\hat{U}^i_\rho(\lambda)_{v,x_v} := \frac{\sum\limits_{x \in \mathcal{X}(v,x_v)} \exp \left\langle -\theta^i(\lambda)/\rho, \phi(x) \right\rangle}{\exp(-\hat{U}^i_\rho(\lambda)/\rho)} , \quad (11)$$

where $\mathcal{X}(v, x_v) = \{x' \in \mathcal{X} \colon x'_v = x_v\}$, we have:

Lemma 1 *(follows from Theorem 1 in [14]) The function $\hat{U}_\rho(\lambda) = \sum_{i=1}^2 \hat{U}^i_\rho(\lambda)$ is well-defined and continuously differentiable at any $\lambda \in \mathbb{R}^{\sum_{v \in \mathcal{V}} |\mathcal{X}_v|}$. Moreover, this function is concave, and its gradient*

$$\nabla \hat{U}_\rho(\lambda) = D\hat{U}^1_\rho(\lambda) - D\hat{U}^2_\rho(\lambda) \quad (12)$$

is Lipschitz-continuous with constant $L_\rho = 2\frac{|\mathcal{V}|}{\rho}$.

This lemma is analogous to the "Computing Lipschitz" lemma in [7] with the significant difference that we consider the ℓ_2-norm instead of the ℓ_1-norm. Jojic at al. [7] inconsistently apply an algorithm based on the ℓ_2-norm, however. Therefore, the role of the ℓ_1–Lipschitz estimate (which reads $L_\rho = \frac{2}{\rho}$, see [7]) for the algorithm design remains unclear in [7].

Optimal First-Order Iterative Optimization It is known [13] that concave continuously differentiable (with Lipschitz constant L) functions can be maximized by iterative first-order optimization methods in $O(\sqrt{\frac{L}{\epsilon}})$ iterations, where ϵ determines the absolute precision of achieved objective value. Thus, by virtue of Lemma 1, the number of iterations can grow as $\sqrt{|\mathcal{V}|}$ with the size of a model, in the worst case.

Next, we present such an algorithm, omitting technical details which can be found in Nesterov's book ([13] p. 76) and in the original paper [12].

Algorithm 1 *(Variant of Algorithm 2.2.6 in [13]) In addition to the Lipschitz-constant L_ρ, we introduce variables $\gamma^t, \alpha^t, \omega \in \mathbb{R}$ and vectors $\lambda^t, v^t, y^t \in \mathbb{R}^{\sum_{v \in \mathcal{V}} |\mathcal{X}_v|}$. Superscript t indexes the iteration.*

```
1. Choose $\lambda^0 = v^0 \in \mathbb{R}^{\sum_{v \in \mathcal{V}} |\mathcal{X}_v|}$ and set $\gamma^0 = L_\rho$.
2. t-th iteration ($t \ge 0$):
   (a) Compute $\hat{U}_\rho(\lambda^t)$ and $\nabla \hat{U}_\rho(\lambda^t)$.
   (b) Find $\omega^t \le L_\rho$ as small as possible, such that
       $$\hat{U}_\rho(y^t) \ge \hat{U}_\rho(\lambda^t) + \frac{1}{2\omega^t} \|\nabla \hat{U}_\rho(\lambda^t)\|^2 , \quad (13)$$
       where $y^t = \lambda^t + \frac{1}{\omega^t} \nabla \hat{U}_\rho(\lambda^t)$.
   (c) Compute $\alpha^t \in (0,1)$ from $\omega^t (\alpha^t)^2 = (1 - \alpha^t)\gamma^t$ and set $\gamma^{t+1} = (1 - \alpha^t)\gamma^t$.
   (d) Set $v^{t+1} = \frac{(1-\alpha^t)\gamma^t v^t + \alpha^t \nabla \hat{U}_\rho(\lambda^t)}{\gamma^{t+1}}$.
   (e) Choose $\lambda^{t+1} = \frac{\alpha^t \gamma^t v^{t+1} + \gamma^{t+1} y^t}{\gamma^t}$.
```

Lemma 2 *[13] Condition (13) is fulfilled for any $\omega^t \ge L_\rho$.*

Theorem 1 *(modified Theorem 2.2.2 in [13]) Algorithm 1 has the following bound on worst-case sub-optimality:*

$$\hat{U}^*_\rho - \hat{U}_\rho(\lambda^t) \le \frac{L}{\left(1 + \frac{t}{2}\sqrt{\frac{L}{\omega^{*t}}}\right)^2} \|\lambda^0 - \lambda^*\|^2 , \quad (14)$$

where $\hat{U}^*_\rho$ and λ^* are optimal function and variable values and $\omega^{*t} = \max_{k \le t} \omega^k$.

The estimate (14) shows that ω^t should be as small as possible. One possible way to achieve this is to perform exact linear search in the direction of $\nabla \hat{U}_\rho(\lambda^t)$ at each iteration of Algorithm 1, which is not particular efficient however. A simple alternative is to set $\omega^t = L_\rho$ in view of

Lemma 2, as done in [7]. Our experiments however show that the worst-case estimate of L_ρ according to Lemma 1 is quite loose and leads to a poor convergence rate.

Instead, we applied backtracking linear search ([3] p. 464, [11]), which consistently leads to speed-up factors up to 100 for our datasets. The analysis of backtracking linear search (see [11], eqn. (4.12), for a detailed proof) shows that for t iterations one needs no more than

$$N_k \leq 2\left[1 + \frac{\ln d}{\ln u}\right](t+1) + \frac{1}{\ln u}\ln\frac{2uL}{dL_0} \qquad (15)$$

oracle calls, where $d, u, L_0 \in \mathbb{R}$ are parameters of the search procedure. Using $d = u = L_0 = 2$, for example, each iteration of Algorithm 1 requires about 4 oracle calls on average (empirically, for our datasets, 3 or 4 oracle calls per iteration in most cases).

Selecting the Smoothing Parameter Inequality (9) and Lemma 1 show that selection of the smoothing value ρ is a trade-off between accuracy of the approximation and speed of the algorithm. The following lemma describes how to optimally select ρ for any algorithm $\mathcal{A}$ that satisfies some conditions.

Lemma 3 *Let $\mathcal{A}$ be any algorithm depending on a smoothing parameter $\rho > 0$ with convergence rate $\hat{U}^*_\rho - \hat{U}_\rho(\lambda^t) \leq \frac{1}{\rho\tau(t)}$, where $\tau(t)$ is a monotonously non-decreasing function of the number of iterations t. Suppose that $U(\lambda) - \hat{U}_\rho(\lambda) \leq \rho\Delta$, for some $\Delta > 0$ and $\forall\rho, \lambda$, where $\hat{U}_\rho$ is the smoothed objective function U. Let ϵ be the prescribed precision. Then, selection of the smoothing parameter ρ as*

$$\rho = \frac{\epsilon}{2\Delta} \qquad (16)$$

minimizes the worst-case bound on the number of iterations to achieve precision ϵ.

Our empirical results show $\omega^{*t} \propto \frac{1}{\rho}$ for ω^{*t} defined in Theorem 1. Thus, according to (14) and (9), this lemma can be directly applied to Algorithm 1, as done in [14] and [7]. In these papers, an upper bound $\Delta = 2\log|\mathcal{X}|$ was used, leading to

$$\rho = \frac{\epsilon}{4\log|\mathcal{X}|}\,. \qquad (17)$$

This bound, however, can be rather loose in practice, that slows down convergence.

In contrast to this worst-case approach, we *adapt* ρ so as to allow for stronger smoothing in the initial and intermediate phase of the iteration, while still achieving the precision ϵ at convergence. We select ρ such that $\Delta \approx U(\lambda^0) - \hat{U}_\rho(\lambda^0) \lessapprox \epsilon/2$, increasing ρ by the factor 2 if necessary. Usually, 3 to 6 computations of $\hat{U}_\rho(\lambda^0)$ suffice until $U(\lambda^0) - \hat{U}_\rho(\lambda^0) > \epsilon/2$. Such adaptive estimation of ρ leads to a speed up of the overall algorithm of order $2^2 \ldots 2^5$. We check the inequality $U(\lambda^t) - \hat{U}_\rho(\lambda^t) < \epsilon/2$ during the iteration. If it does not hold (e.g. when convergence slows down close to the optima of $\hat{U}_\rho$), we decrease ρ by the factor 2.

3. Stopping Criterion

The stopping criterion we propose is based on a *duality gap* between the value of the primal LP, given by (4), and its dual $U(\lambda)$, given by right-hand side of (3). Since we optimize the dual problem and thus know its value, we focus in this section on estimating the value of the primal function, whose objective we will denote by P. We further denote by $\mathbb{R}_+(\mathcal{G}) = \mathbb{R}_+^{|\otimes_{v\in\mathcal{V}}\mathcal{X}_v| + |\otimes_{uv\in\mathcal{E}}\mathcal{X}_{uv}|}$ a nonnegative linear half-space containing the local polytope $\mathcal{L}(\mathcal{G})$. Finally, we denote the optimal primal value over the local polytope by $P^* = \min_{\mu\in\mathcal{L}(\mathcal{G})} P(\mu) = \min_{\mu\in\mathcal{L}(\mathcal{G})} \langle\theta, \mu\rangle$.

A typical issue for many algorithms which optimize a dual problem (3) is that, during the iteration, one can only get *infeasible* primal points $\tilde{\mu}$, that is $\tilde{\mu}$ does not satisfy the constraints of (4). In this connection, we propose to construct a mapping $\chi\colon \mathbb{R}_+(\mathcal{G}) \to \mathcal{L}(\mathcal{G})$ yielding primal feasible points, which enjoys the following properties:

Lemma 4 *Let $\tilde{\mu}^t \in \mathbb{R}_+(\mathcal{G})$ be any sequence such that $P(\tilde{\mu}^t) \to P^*$. Let also $\min_{\mu\in\mathcal{L}(G)} \|\tilde{\mu}^t - \mu\| \to 0$. Then $P(\chi(\tilde{\mu}^t)) \to P^*$.*

We define the shorthand $\mu' := \chi(\tilde{\mu})$ and the set $\mathcal{L}(\mathcal{G}, \mu'(\mathcal{V})) = \{\mu \in \mathcal{L}(\mathcal{G})\colon \mu_v = \mu'_v,\ v \in \mathcal{V}\}$. A mapping χ as characterized by Lemma 4 can be constructed in the following two-steps way:

$$\mu''_v = \frac{\tilde{\mu}_v}{\sum_{x_v\in\mathcal{X}_v}\tilde{\mu}_v(x_v)},\ v \in \mathcal{V}, \qquad (18)$$

$$\mu' = \arg\min_{\mu\in\mathcal{L}(\mathcal{G},\mu''(V))} \langle\theta, \mu\rangle\,. \qquad (19)$$

It is easy to see that problem (19) decomposes into $|\mathcal{E}|$ independent optimization problems (for each $uv \in \mathcal{E}$) of the form

$$\begin{aligned}
\mu'_{uv} = \arg\min_{\mu_{uv}} &\sum_{x_{uv}\in\mathcal{X}_{uv}} \theta_{uv}(x_v)\mu_{uv}(x_{uv}),\\
\text{s.t.}\ &\begin{array}{l}\sum_{x_v\in\mathcal{V}}\mu_{uv}(x_{uv}) = \mu''_u(x_u),\ x_u \in \mathcal{X}_u\\ \sum_{x_u\in\mathcal{V}}\mu_{uv}(x_{uv}) = \mu''_v(x_v),\ x_v \in \mathcal{X}_v\\ \mu_{uv}(x_{uv}) \geq 0,\ x_{uv} \in \mathcal{X}_{uv}.\end{array}
\end{aligned} \qquad (20)$$

Such linear programs are well-studied and known as *transportation problems*. Since the size of each individual problem is small, they can be easily solved by any appropriate method of linear programming.

We point out that the existence of a sequence $\tilde{\mu}^t$ satisfying the conditions of Lemma 4 is important for the theoretical properties of $\chi(\tilde{\mu}^t)$ to hold. But to compute $\chi(\tilde{\mu}^t)$, one only needs a subset of coordinates of the sequence, namely $\tilde{\mu}_v^t$, $v \in \mathcal{V}$. We show existence of a sequence $\tilde{\mu}_v^t$ by construction.

Theorem 2 *When $\rho \to 0$, $t \to \infty$, for the sequence*

$$\mu_v^{\rho,t} = \frac{D\hat{U}_\rho^1(\lambda^t)_v + D\hat{U}_\rho^2(\lambda^t)_v}{2}, \; v \in \mathcal{V} \qquad (21)$$

a sequence $\tilde{\mu}^{\rho,t} \in \mathbb{R}_+(\mathcal{G})$ exists such that $\tilde{\mu}_v^{\rho,t} = \mu_v^{\rho,t}$, $v \in \mathcal{V}$, and $\tilde{\mu}^{\rho,t}$ satisfies the conditions of Lemma 4, namely $\forall \delta > 0\ \exists \rho > 0$: $\exists t^* : \forall t > t^*\ \|P(\tilde{\mu}^{\rho,t}) - P^*\| < \delta$ and $\min_{\mu \in \mathcal{L}(\mathcal{G})} \|\tilde{\mu}^{\rho,t} - \mu\| < \delta$. Here λ^t is computed by Algorithm 1 for a given ρ, and $D\hat{U}_\rho^i(\lambda)_v$, $i = 1, 2$, are vectors with coordinates $D\hat{U}_\rho^i(\lambda)_{v,x_v}$ given by (11).

This theorem basically says that for ρ *small enough*, values $\mu_v^{\rho,t}$ plugged into formula (18) in place of $\tilde{\mu}_v$ would yield primal objective values which will converge with $t \to \infty$ to a value *close enough* to P^*.

4. Experiments

In our experiments we study different grid structured models with potentials of first and second order. Exemplarily we will discuss two of them. The first one is a synthetic model with 20×20 nodes, five labels and potential functions sampled uniformly from the interval $[0; 0.5]$ (corresponding plot in Figure 4 and top plots in Figures 1-3), the second is the Tsukuba stereo problem from the Middlebury MRF-Benchmark [19] (bottom plots in Figures 1-3).

We compare different variants of the Nesterov's method (NEST) implemented in openGM [1] among each other and with standard methods, namely TRW-S [8], Norm-Product Belief-Propagation (NPBP) [5] and sub-gradient methods [10]. Thanks to the authors we can use their original code for TRW-S and NPBP. Since we compare different implementations of these methods, on the time axis we plot the number of oracle calls (function or gradient evaluations) instead of direct time measurements.

For the lower and upper bounds shown in our plots, we used values of the non-smooth dual objective U (see its definition after eq. (8)) and primal objective P, evaluated by means of (20), respectively.

Lipschitz Constant Estimation First we compare the performance of Nesterov's method for different estimates of the Lipschitz constant. Adaptive selection of the Lipschitz constant leads to a significantly faster convergence than the fixed one. We also applied the calculation of the Lipschitz constant L_ρ as suggested in [7]. The top plot in Figure 1 shows that for the synthetic model the algorithm does not converge to the optimum, as their estimation of the Lipschitz constant does not yield valid bounds, as empirically observed by checking criterion (13). For the Tsukuba model, this effect is not so pronounced, which explains good applied results reported in [7]. As can be seen in Figure 1 (bottom), the remaining gap is not significantly larger in this case, however we observed violations of (13) for Jojic's method here as well. On the Tsukuba model example one can also see, that a gradient step size, inferred from a fixed L_ρ given by Lemma 1, is so small, that there is almost no improvement of the objective function during iteration. Due to the smoothing, a gap between upper and lower bounds remains for any $\rho > 0$ and decreases with the smoothing (see Theorem 2).

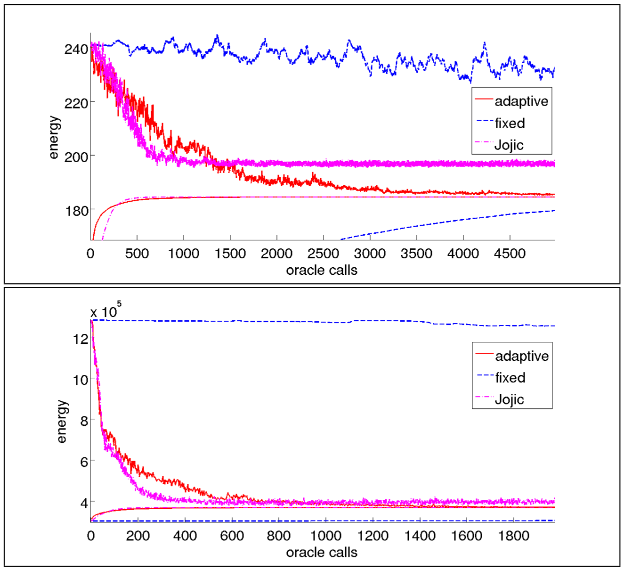

Figure 1. Nesterov's method for the synthetic (top) and Tsukuba (bottom) models with 3 different ways of Lipschitz constant L_ρ selection: (a) fixed, (b) adaptive, (c) L_ρ selected according to [7]. Smoothing value $\rho = 0.01$ is fixed. While adaptive estimation outperforms the fixed setting, the method suggested in [7] produces invalid values of the Lipschitz constant and does not converge to the optimum.

Smoothing selection Next we compare Nesterov's method with fixed smoothing to a method with adaptive smoothing for the same precision. In the first case, precision is selected according to (17). Both methods use adaptive estimation of the Lipschitz constant. Results are shown in Figure 2. Adaptive smoothing often works faster, as can be seen in the Figure 2, since it leads to a smoother function and thus to smaller values of the Lipschitz constant.

Comparison TRW-S and Sub-Gradient Compared to TRW-S and sub-gradient methods, the proposed method gives better lower bound then TRW-S and converges significantly faster than the sub-gradient ascent. As an update rule[2] for the sub-gradient ascent we use $\lambda^{t+1} = \lambda^t + \frac{\partial U(\lambda^t)}{2\sqrt{t+1}}$, where $\partial U(\lambda^t)$ denotes a sub-gradient of the dual func-

[2] Other step-size rules are also conceivable, but they do not change the

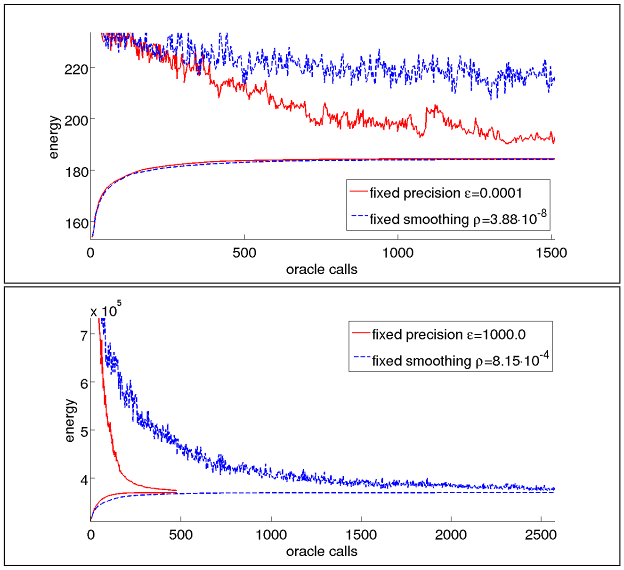

Figure 2. Nesterov's method for the synthetic (top) and Tsukuba (bottom) models with (a) fixed smoothing ρ and (b) adaptive smoothing calculated from a fixed precision ϵ. Parameters ρ and ϵ are connected by (17). Adaptive smoothing usually works faster, since it leads to a smoother function and thus to smaller values of the Lipschitz constant. The bottom plot shows that the algorithm, which uses adaptive smoothing, requires less than 500 iterations to achieve the required precision and to stop.

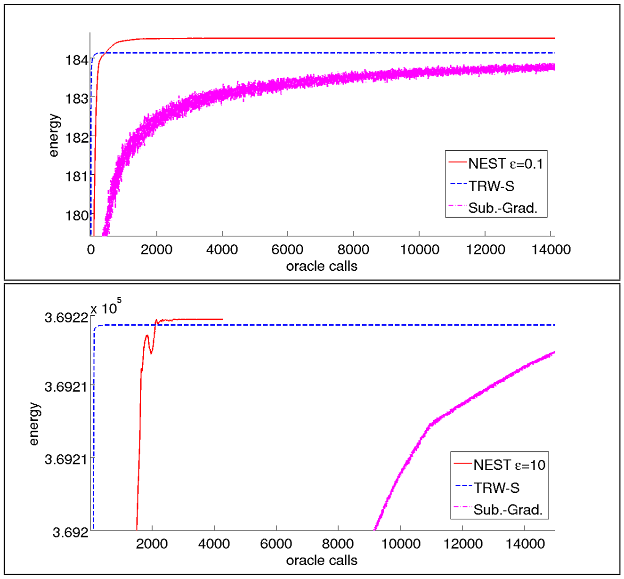

Figure 3. Comparison of (a) Nesterov's, (b) TRW-S and (c) sub-gradient methods for the synthetic (top) and Tsukuba model (bottom). The plot shows LP lower bounds. TRW-S is the fastest one, but it gets stuck in a fixed point in the top plot, whereas Nesterov's method calculates a tighter lower bound on the objective. The sub-gradient method is the slowest one.

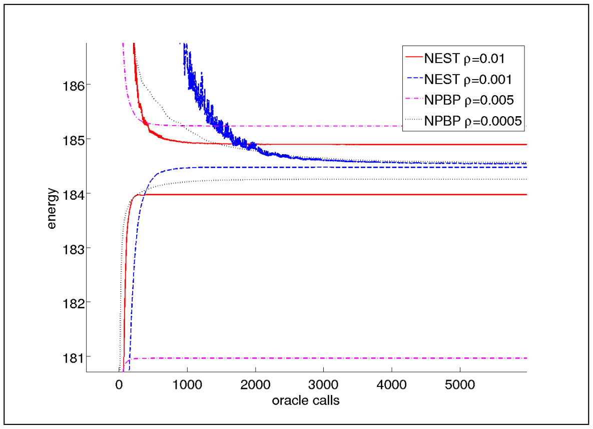

Figure 4. Comparison of (a) Nesterov's method and (b) NPBP for the synthetic model for two different smoothing values. Corresponding values of ρ for Nesterov's method and NPBP differ by a factor of two due to different entropy approximations used in these methods. For smaller ρ both methods produce tighter bounds, but show slower convergence.

tion U. TRW-S is enormously fast, but can get stuck in local fixed points, as shown in the top plot of Figure 3. Unlike TRW-S, the sub-gradient method is guaranteed to converge to the optimum, but its convergence is extremely slow.

Comparison to Smoothed NPBP Finally, we compare our method of solving a smoothed objective to NPBP, for which we use the entropy approximation as suggested in [5] and set $c_{ab} = 1$, $c_a = 0$ and $c_{ab,a} = 0$. We have selected different values of smoothing parameters ρ for these methods to guarantee, that upper bounds to a difference between smoothed and non-smoothed objectives coincide. For NPBP we apply additionally our method to construct a primal bound. This ends up in a mathematically sound stopping criterion for NPBP, which is lacking in [5]. However, since we optimize different smoothed functions, their optimal values differ and a fair comparison is not obvious. With less smoothing we obtain tighter bounds for both methods as shown in Figure 4, while the speed of convergence decreases when the smoothing decreases.

5. Conclusion

We presented an in-depth study of Nesterov's optimal first-order optimization scheme applied to the MAP labeling problem based on Lagrangian decomposition. Our study shows that a direct application of the scheme leads to poor convergence rates based on parameter settings governed by the worst-case optimality bounds. As a remedy, we proposed to modify the approach by i) adaptively estimating and selecting *both* the Lipschitz constant and the smoothing parameter, respectively, and ii) a sound termination condition based on the primal-dual gap.

plots qualitatively.

Modification i) still enables to theoretically infer favorable complexity bounds and runtime guarantees. Contribution ii) removes ad-hoc thresholds for stopping the iteration and thus ensures comparability and reproducibility of results. It entails a method for constructing a primal feasible solution that should also be applicable to alternative approaches focusing on dual objective optimization. In our experiments we applied it to generate a primal solution for a Norm-Product Belief Propagation [5]. Our experiments also show that our method i) converges significantly faster than the sub-gradient ascent and ii) has a comparable convergence to the state-of-the-art smoothed Norm-Product Belief Propagation.

Our further work will focus on graphical models that are more general than the grid graphs considered in this paper. While such grid graphs naturally appear in standard low-level vision problems as current benchmarks show, less structured graphs are also of vital interest for various applications. Early experiments indicate that the relative performance of our method increase considerably in these cases, and that our contribution provides a solid basis for tackling such problems.

Acknowledgements. Authors gratefully acknowledge support by the German Science Foundation (DFG) within the Excellence Initiative (HCI) and by the joint research project "Spatio/Temporal Graphical Models and Applications in Image Analysis", grant GRK 1653.

References

[1] B. Andres, J. Kappes, U. Köthe, C. Schnörr, and F. Hamprecht. An empirical comparison of inference algorithms for graphical models with higher order factors using opengm. In *Pattern Recognition, Proc. 32th DAGM Symposium*, 2010.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, USA, 2004.

[4] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Math. Imaging and Vision, to appear*, 2011.

[5] T. Hazan and A. Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Trans. on Inf. Theory,*, 56(12):6294 –6316, 2010.

[6] J. K. Johnson, D. Malioutov, and A. S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *45th Ann. Allerton Conf. on Comm., Control and Comp.*, 2007.

[7] V. Jojic, S. Gould, and D. Koller. Accelerated dual decomposition for MAP inference. In *ICML*, pages 503–510, 2010.

[8] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. on PAMI*, 28(10):1568–1583, 2006.

[9] N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *IEEE Trans. PAMI (in press)*.

[10] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[11] Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE Discussion Paper 2007/76*.

[12] Y. Nesterov. A method for solving a convex programming problem with convergence rate $1/k^2$. *Soviet Math. Dokl.*, 27(2):372–376, 1983.

[13] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[14] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, Ser. A(103):127–152, 2004.

[15] P. Ravikumar, A. Agarwal, and M. Wainwright. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *JMLR*, 11:1043–1080, 2010.

[16] R. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 2nd edition, 2004.

[17] M. Schlesinger. Syntactic analysis of two-dimensional visual signals in the presence of noise. *Kibernetika*, (4):113–130, 1976.

[18] M. Schlesinger and V. Giginyak. Solution to structural recognition (max,+)-problems by their equivalent transformations. in 2 Parts. *Control Systems and Computers*, (1-2), 2007.

[19] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:1068–1080, June 2008.

[20] M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on (hyper)trees: message passing and linear programming approaches. In *Allerton Conf. on Communication, Control and Computing*, 2002.

[21] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, 2008.

[22] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Trans. on PAMI*, 29(7), July 2007.

[23] T. Werner. Revisiting the decomposition approach to inference in exponential families and graphical models. Technical report, CMP, Czech TU, 2009.

6. Proofs

6.1. Proof of Lemma 1

Our proof is based on Theorem 1 p.131 from [14]. We repeat its formulation here in a simplified form to make our proof self-contained.

Nesterov's smoothing Theorem Let Q be a closed bounded set in an Euclidean space Π equipped with the norm $\|\cdot\|_\Pi$ and a scalar product $\langle\cdot,\cdot\rangle_\Pi$. Let Λ be another Euclidean space equipped with the norm $\|\cdot\|_\Lambda$ and scalar product $\langle\cdot,\cdot\rangle_\Lambda$. Let the function $\psi\colon Q \to \mathbb{R}$ be continuous and convex and $A\colon \Lambda \to \Pi$ be a linear operator. Its norm is defined as follows:

$$\|A\|_{\Lambda,\Pi} = \max_{\lambda,p}\{\langle A\lambda, p\rangle_\Pi : \|\lambda\|_\Lambda = 1,\ \|p\|_\Pi = 1\}\,. \quad (22)$$

Let $d\colon Q \to \mathbb{R}$ be a *prox-function* of the set Q, assuming that $d(p)$ is continuous and strongly convex on Q. Let $p_0 := \arg\min_{p\in Q} d(p)$ and for any $p \in Q$ and some $\sigma \geq 0$

$$d(p) \geq \frac{1}{2}\sigma\|p - p_0\|^2_\Pi + d(p_0)\,. \quad (23)$$

Theorem 1 p. 131 [14] *The function*

$$F_\rho(\lambda) = \max_{p\in Q}\langle A\lambda, p\rangle_\Pi - \psi(p) - \rho d(p) \quad (24)$$

is well-defined and continuously differentiable at any $\lambda \in \Lambda$. Moreover, this function is convex and its gradient is Lipschitz continuous with constant

$$L_\rho = \frac{\|A\|^2_{\Lambda,\Pi}}{\rho\sigma}\,. \quad (25)$$

Analysis of the smooth approximation (7) Our goal now is to represent (7) in the form (24) to be able to apply Nesterov's smoothing Theorem.

Let us define space Λ as $\mathbb{R}^{\sum_{v\in\mathcal{V}}|\mathcal{X}_v|}$ and let $\|\cdot\|_\Lambda$ be the ℓ_2 norm. Let also $\Pi = \mathbb{R}^{|\mathcal{X}|}$ and corresponding norm $\|\cdot\|_\Pi$ be the ℓ_1-norm. Scalar products $\langle\cdot,\cdot\rangle_\Lambda$ and $\langle\cdot,\cdot\rangle_\Pi$ are coordinate-wise dot products in spaces of corresponding dimensionality. For the clarity of notation we will use also notation $\langle\cdot,\cdot\rangle_\Theta$ for a coordinate-wise dot product in the space $\mathbb{R}^{|\otimes_{v\in\mathcal{V}}\mathcal{X}_v| + |\otimes_{uv\in\mathcal{E}}\mathcal{X}_{uv}|} \ni \theta$. As it is shown in e.g. [14], p. 139, $\hat{U}^2_\rho(\lambda)$ defined by (7) can be represented as follows

$$\hat{U}^2_\rho(\lambda) = \min_{p\in Q}\left\langle \theta^i(\lambda), \sum_{x\in\mathcal{X}} p(x)\phi(x)\right\rangle_\Theta + \rho\sum_{x\in\mathcal{X}} p(x)\log p(x)\,. \quad (26)$$

Here the set Q is a $|\mathcal{X}|$-dimensional simplex in Π. It turns out that (26) has the following form:

$$\hat{U}^2_\rho(\lambda) = -\max_{p\in Q}\langle A\lambda, p\rangle_\Pi - \psi(p) - \rho d(p)\,. \quad (27)$$

Here

$$\psi(p) = \left\langle \theta', \sum_{x\in\mathcal{X}} p(x)\phi(x)\right\rangle_\Theta \quad \text{for} \quad (28)$$

$$\theta'_v(x_v) = \frac{\theta_v(x_v)}{2},\ x_v \in \mathcal{X}_v,\ v \in \mathcal{V}\,,$$
$$\theta'_{uv}(x_{uv}) = \theta_{uv}(x_{uv}),\ x_{uv} \in \mathcal{X}_{uv},\ uv \in \mathcal{E}\,, \quad (29)$$

and

$$d(p) = \sum_{x\in\mathcal{X}} p(x)\log p(x)\,. \quad (30)$$

The rectangular matrix A has $|\mathcal{X}|$ rows and $\sum_{v\in\mathcal{V}}|\mathcal{X}_v|$ columns. Each row corresponds to a single labeling $x \in \mathcal{X}$. Thus let us index rows with $x \in \mathcal{X}$ and columns with $(v, x_v),\ x_v \in \mathcal{X}_v,\ v \in \mathcal{V}$. Then

$$(A)_{x',(v,x_v)} = \begin{cases} 1, & x'_v = x_v \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

We will denote by $(A)_x$ a x-row of matrix A.

As proved in [14] p.139, function $d(p)$ is continuous and strictly convex and

$$d(p) \geq \frac{1}{2}\sigma||p - p_0||^2_\Pi + d(p_0)\,, \quad (32)$$

where

$$p_0 = \arg\min_{p\in Q} d(p) = \frac{1}{|\mathcal{X}|}, \quad (33)$$

$d(p_0) = -\log|\mathcal{X}|$ and $\sigma = 1$.

The function $\psi(p)$ is continuous and convex, thus we can apply Nesterov's smoothing Theorem, according to which function $-\hat{U}^2_\rho(\lambda)$ in (27) is well-defined and continuously differentiable at any $\lambda \in \Lambda$. Moreover, it is convex and its gradient is Lipschitz continuous with constant

$$L_\rho = \frac{1}{\rho}\|A\|^2_{\Lambda,\Pi}\,. \quad (34)$$

Computing $||A||_{\Lambda,\Pi}$ Let us consider eq. (22). It can be rewritten as

$$\|A\|_{\Lambda,\Pi} = \max_{\lambda\in\Lambda}\max_{x\in\mathcal{X}}\{\langle\lambda, (A)_x\rangle_\Lambda : \|\lambda\|_\Lambda = 1\}\,. \quad (35)$$

Since $\|\cdot\|_\Lambda$ is the ℓ_2-norm and all vectors $(A)_x$ have exactly $|\mathcal{V}|$ non-zero entries, each equal to 1, then for any

$x \in \mathcal{X}$ the maximum in (35) is attained at $\lambda = \frac{(A)_x}{\|(A)_x\|_\Lambda} = \frac{(A)_x}{\sqrt{\mathcal{V}}}$. Thus

$$\|A\|_{\Lambda,\Pi} = \frac{\langle (A)_x, (A)_x \rangle}{\sqrt{\mathcal{V}}} = \sqrt{\mathcal{V}}. \tag{36}$$

To finalize our proof, we remark, that analogous considerations can be applied to $\hat{U}^1_\rho$ and from $\hat{U}_\rho = \hat{U}^1_\rho + \hat{U}^2_\rho$ and (34) follows statement of the lemma.

Remark Please note, that considering the ℓ_1-norm for $\|\cdot\|_\Lambda$ would lead to $\|A\|_{\Lambda,\Pi} = 1$, which is exactly the result obtained in [7]. But this estimation of Lipschitz constant can be used only with ℓ_1-norm based Nesterov's algorithm, that is not the case in [7].

6.2. Proof of Lemma 3

Total gap between current value of the smooth function $\hat{U}_\rho(\lambda^t)$ and an optimum of the non-smooth one by definition is equal by:

$$\epsilon = U^* - \hat{U}_\rho(\lambda^t) = \hat{U}^*_\rho - \hat{U}_\rho(\lambda^t) + U^* - \hat{U}^*_\rho\,. \tag{37}$$

Plugging assumptions of the lemma to (37) we get

$$\epsilon \le \frac{1}{\rho\tau(t)} + \rho\Delta\,. \tag{38}$$

Thus

$$\frac{1}{\tau(t)} \ge \rho\epsilon - \rho^2\Delta\,. \tag{39}$$

Maximizing the right-hand-side w.r.t. ρ yields $\rho = \frac{\epsilon}{2\Delta}$, which proves this lemma, since the maximum of $\frac{1}{\tau(t)}$ corresponds to a minimal number of steps t.

6.3. Proof of Lemma 4

Our proof consists of two parts:

i) we will prove that from the lemma conditions follows, that sequence $\tilde{\mu}^t$ converges to a set of optimal solutions of the primal problem. This implies, that corresponding values μ''_v, defining constraints of (19), converge to values μ^*_v of an optimal solution;

ii) since solving (19) with $\mu''_v = \mu^*_v$ leads to the optimal value $\langle\theta, \mu'\rangle = P(\mu') = P^*$ of the primal objective, substitution of μ^*_v by a close value $\tilde{\mu}_v$ leads to a value $P(\mu')$, which is close to P^*.

Convergence of $\tilde{\mu}^t$ to the set of optimal primal solutions Let us denote by M^* a set of all optimal solutions of the primal problem (4). Let us also denote by

$$\|\mu - M^*\| = \min_{\mu^* \in M^*} \|\mu - \mu^*\| \tag{40}$$

distance of a point $\mu \in \mathbb{R}_+(\mathcal{G})$ to the set M^*. Let also $\hat{\mu}^t = \arg\min_{\mu \in \mathcal{L}(\mathcal{G})} \|\tilde{\mu}^t - \mu\|$ be an Euclidean projection of $\tilde{\mu}^t$ to the local polytope $\mathcal{L}(\mathcal{G})$. Then

$$\|\tilde{\mu}^t - \hat{\mu}^t\| + \|\hat{\mu}^t - M^*\| \ge \|\tilde{\mu}^t - M^*\|\,. \tag{41}$$

Since (4) is a linear program, then from $\|\tilde{\mu}^t - \hat{\mu}^t\| \to 0$ follows $P(\tilde{\mu}^t) - P(\hat{\mu}^t) \to 0$ and, since according to lemma conditions $P(\tilde{\mu}^t) - P^* \to 0$, then $P(\hat{\mu}^t) - P^* \to 0$. Thus, due to convexity of (4), $\|\hat{\mu}^t - M^*\| \to 0$. Comparing this to (41) and taking into account that $\|\tilde{\mu}^t - \hat{\mu}^t\| \to 0$ according to lemma conditions, we get

$$\|\tilde{\mu}^t - M^*\| \to 0\,. \tag{42}$$

This means, that for any $\delta \ge 0$ $\exists t^* > 0$ and some $\mu^* \in M^*$, that $\forall t > t^*$

$$\|\tilde{\mu}^t - \mu^*\| < \delta\,. \tag{43}$$

Optimal objective value of (19) changes continously with μ''_v Let us consider problem (19) with plugged in values $\mu'' = \mu^*$. In this case the corresponding objective value $P(\mu')$ is equal to the optimal one P^*.

In what follows we will prove that an optimal value of the problem (20) continuously depends on the values μ'', which define its constraints, *i.e.* small changes of μ'' imply small changes of the objective. From this trivially follows, that the same property holds for (19) as well.

Problem (20) obviously satisfies Slater's condition [3] due to affinity of its constraints and it always has at least one feasible point when

$$\sum_{x_v \in \mathcal{X}_v} \mu''_v(x_v) = \sum_{x_u \in \mathcal{X}_u} \mu''_u(x_u),\ v, u \in \mathcal{V} \tag{44}$$

This condition holds for μ^* due to the first constraint in (4) and for $\tilde{\mu}^t$ due to (18). Since (see (4)) $1 \ge \mu_{uv} \ge 0$ its optimal value is always finite. Thus its Lagrange dual has the same finite optimal value.

The Lagrange dual for (20) reads

$$\begin{aligned}&\max_{\alpha,\beta} \sum_{x_v \in \mathcal{X}_v} \alpha_v(x_v)\mu''_v(x_v) + \sum_{x_u \in \mathcal{X}_u} \beta_u(x_u)\mu''_u(x_u)\\ &\text{s.t. } \theta_{uv}(x_u v) - \alpha_u(x_u) - \beta_v(x_v) \ge 0,\ \forall x_{uv} \in \mathcal{X}_{uv}\end{aligned} \tag{45}$$

It depends on μ'' only throw its objective, which continuously depends on μ''. Since optimal value of (45) is finite, it is attained in one of vertices of its constraint set, which implies that it changes continuously with μ''.

6.4. Proof of Theorem 2

Our proof consists of three steps:

i) we construct a sequence $\tilde{\mu}^{\rho,t}, t \to \infty$ of (possibly infeasible) points in $\mathbb{R}_+(\mathcal{G})$;

ii) we will show that for any $\rho > 0$ this sequence converges to the feasible set $\mathcal{L}(\mathcal{G})$, i.e.

$$\|\tilde{\mu}^{\rho,t} - \mathcal{L}(\mathcal{G})\| \xrightarrow[t\to\infty]{} 0\,; \tag{46}$$

iii) we will show how the smoothing ρ should be selected to guarantee that for any $\delta > 0$ sequence $P(\tilde{\mu}^{\rho,t})$ converges to a δ-neighborhood of P^*, *i.e.* that $\forall \delta > 0\; \exists \rho > 0$ and $t^* > 0$ such that $\forall t \geq t^*\; |P(\tilde{\mu}^{\rho,t}) - P^*| < \delta$.

Constructing sequence $\tilde{\mu}^{\rho,t}$ Let us consider vectors $m^1_\rho(\lambda^t)$ and $m^2_\rho(\lambda^t)$ defined as

$$m^i_\rho(\lambda^t) := \frac{\sum_{x\in\mathcal{X}} \phi(x)\cdot \exp\left\langle -\theta^i(\lambda^t)/\rho, \phi(x)\right\rangle}{\sum_{x\in\mathcal{X}} \exp\left\langle -\theta^i(\lambda^t)/\rho, \phi(x)\right\rangle}\,, i = 1,2 \tag{47}$$

for a sequence λ^t generated by Algorithm 1. Since the $\mathcal{G}^i$ are trees, coordinates of $m^i_\rho(\lambda^t)$ are easily computable by dynamic programming. These coordinates read for $x_v \in \mathcal{X}_v,\ v \in \mathcal{V}$ and $x_{uv} \in \mathcal{X}_{uv},\ uv \in \mathcal{E}^i$ as

$$m^i_\rho(\lambda^t)_{v,x_v} = \frac{\sum\limits_{x\in\mathcal{X}(v,x_v)} \exp\left\langle -\theta^i(\lambda^t)/\rho, \phi(x)\right\rangle}{\sum_{x\in\mathcal{X}} \exp\left\langle -\theta^i(\lambda^t)/\rho, \phi(x)\right\rangle}\,, \tag{48}$$

$$m^i_\rho(\lambda^t)_{uv,x_{uv}} = \frac{\sum\limits_{x\in\mathcal{X}(uv,x_{uv})} \exp\left\langle -\theta^i(\lambda^t)/\rho, \phi(x)\right\rangle}{\sum_{x\in\mathcal{X}} \exp\left\langle -\theta^i(\lambda^t)/\rho, \phi(x)\right\rangle}\,, \tag{49}$$

where $\mathcal{X}(uv, x_{uv}) = \{x' \in \mathcal{X} \colon x'_{uv} = x_{uv}\}$.

Thus coordinates of $m^i_\rho(\lambda)$ are vertex and edge marginals computed for the subgraph $\mathcal{G}^i$. Moreover,

$$m^i_\rho(\lambda^t)_{v,x_v} = D\hat{U}^i_\rho(\lambda^t)_{v,x_v}\ x_v \in \mathcal{X}_v,\ v \in \mathcal{V}\,. \tag{50}$$

We will construct the sequence $\tilde{\mu}^{\rho,t}$ from $m^i_\rho(\lambda^t)$ in the following way:

$$\tilde{\mu}^{\rho,t}_v(x_v) = \frac{m^1_\rho(\lambda^t)_{v,x_v} + m^2_\rho(\lambda^t)_{v,x_v}}{2},\ x_v \in \mathcal{X}_v,\ v \in \mathcal{V} \tag{51}$$

and

$$\tilde{\mu}^{\rho,t}_{uv}(x_{uv}) = \begin{cases} m^1_\rho(\lambda^t)_{uv,x_{uv}}, uv \in \mathcal{E}^1 \\ m^2_\rho(\lambda^t)_{uv,x_{uv}}, uv \in \mathcal{E}^2 \end{cases}\,. \tag{52}$$

Proving convergence to $\mathcal{L}(\mathcal{G})$ Note that if

$$m^1_\rho(\lambda^t)_{v,x_v} = m^2_\rho(\lambda^t)_{v,x_v}\ \forall x_v \in \mathcal{X}_v,\ v \in \mathcal{V} \tag{53}$$

then $\tilde{\mu}^{\rho,t} \in \mathcal{L}(\mathcal{G})$ by construction. However, equality (53) usually does not hold exactly, but due to (50), Lemma 1 and the fact, that the gradient of a differentiable function vanishes near its optimum,

$$\|m^1_\rho(\lambda^t)_v - m^2_\rho(\lambda^t)_v\| \xrightarrow[t\to\infty]{} 0,\ v \in \mathcal{V}\,. \tag{54}$$

To prove that

$$\min_{\mu\in\mathcal{L}(\mathcal{G})} \|\tilde{\mu}^{\rho,t} - \mu\| \xrightarrow[t\to\infty]{} 0 \tag{55}$$

it suffices to construct such $\hat{\mu}^{\rho,t} \in \mathcal{L}(\mathcal{G})$ that

$$\|\tilde{\mu}^{\rho,t} - \hat{\mu}^{\rho,t}\| \xrightarrow[t\to\infty]{} 0\,. \tag{56}$$

We construct $\hat{\mu}^{\rho,t}$ as follows:

$$\hat{\mu}^{\rho,t} = \arg\min_{\mu\in\mathcal{L}(\mathcal{G})} \sum_{uv\in\mathcal{E}^2} \|\mu_{uv} - m^2_\rho(\lambda^t)_{uv}\|^2 \tag{57}$$

$$\text{s.t.}\ \begin{array}{l} \mu_v = m^1_\rho(\lambda^t)_v,\ v \in \mathcal{V} \\ \mu_{uv} = m^1_\rho(\lambda^t)_{uv},\ uv \in \mathcal{E}^1 \end{array}\,. \tag{58}$$

Problem (57) can be decomposed into $\mathcal{E}^2$ independent subproblems, one for each $uv \in \mathcal{E}^2$:

$$\hat{\mu}^{\rho,t}_{uv} = \arg\min_{\mu_{uv}} \|\mu_{uv} - m^2_\rho(\lambda^t)_{uv}\|^2 \tag{59}$$

$$\text{s.t.}\ \begin{array}{l} \sum_{x_u\in\mathcal{X}_u} \mu_{uv}(x_{uv}) = m^1_\rho(\lambda^t)_{v,x_v},\ x_v \in \mathcal{X}_v \\ \sum_{x_v\in\mathcal{X}_v} \mu_{uv}(x_{uv}) = m^1_\rho(\lambda^t)_{u,x_u},\ x_u \in \mathcal{X}_u \\ \mu_{uv}(x_{uv}) \geq 0,\ x_{uv} \in \mathcal{X}_{uv} \end{array}\,. \tag{60}$$

Since for $m^2_\rho(\lambda^t)_{uv},\ uv \in \mathcal{E}^2$

$$\begin{array}{l} \sum_{x_u\in\mathcal{X}_u} m^2_\rho(\lambda^t)_{uv,x_{uv}} = m^2_\rho(\lambda^t)_{v,x_v},\ x_v \in \mathcal{X}_v \\ \sum_{x_v\in\mathcal{X}_v} m^2_\rho(\lambda^t)_{uv,x_{uv}} = m^2_\rho(\lambda^t)_{u,x_u},\ x_u \in \mathcal{X}_u \\ \mu_{uv}(x_{uv}) \geq 0,\ x_{uv} \in \mathcal{X}_{uv} \end{array} \tag{61}$$

holds, comparing (61) to (60) and taking into account (54) we conclude that

$$\|\hat{\mu}^{\rho,t}_{uv} - m^2_\rho(\lambda^t)_{uv}\| \xrightarrow[t\to\infty]{} 0,\ uv \in \mathcal{E}^2\,. \tag{62}$$

To complete the proof that

$$\|\tilde{\mu}^{\rho,t} - \hat{\mu}^{\rho,t}\| \xrightarrow[t\to\infty]{} 0\,, \tag{63}$$

it remains to remark only that

$$\|\tilde{\mu}^{\rho,t}_v - m^1_\rho(\lambda^t)_v\| \xrightarrow[t\to\infty]{} 0,\ \forall v \in \mathcal{V} \tag{64}$$

due to (51) and (54), and $\tilde{\mu}^{\rho,t}_{uv} = m^1_\rho(\lambda^t)_{uv},\ uv \in \mathcal{E}^1$ due to (58).

Proving convergence to P^* This part of our proof we will build on the results and notation by Werner in [23]. Let us represent function $\hat{U}^i_\rho$, given by (7), via its conjugate [3] (up to a factor ρ) function $\mathcal{H}^i \colon \mathcal{L}(\mathcal{G}^i) \to \mathbb{R}$ as

$$\hat{U}^i_\rho(\lambda^t) = \min_{\mu \in \mathcal{L}(\mathcal{G}^i)} \left\langle \theta^i(\lambda^t), \mu \right\rangle - \rho \mathcal{H}^i(\mu)\,. \tag{65}$$

This representation is well-known [21, 23] and the function $\mathcal{H}^i$ is usually called *entropy*. It is also known [23] that the minimum of (65) is attained at the point $m^i_\rho(\lambda^t)$ defined by (48).

Let us construct a function $\mathcal{W}^i_\rho \colon \mathbb{R}^{\sum_{v \in \mathcal{V}} |\mathcal{X}_v|} \to \mathbb{R}$ such that

$$\mathcal{W}^i_\rho(\lambda) := \left\langle \theta^i(\lambda), m^i_\rho(\lambda) \right\rangle \overset{\text{see (65)}}{=} \hat{U}^i_\rho(\lambda) + \rho \mathcal{H}^i(m^i_\rho(\lambda))\,. \tag{66}$$

Since for a tree-structured graph $\mathcal{G}^i$

$$\min_{x \in \mathcal{X}} \left\langle \theta^i(\lambda), \phi(x) \right\rangle = \min_{\mu \in \mathcal{L}(\mathcal{G}^i)} \left\langle \theta^i(\lambda), \mu \right\rangle \tag{67}$$

holds (see e.g. [21] for a proof) and from (8) and (65) follows that $\mathcal{H}^i(\mu) \le \log|\mathcal{X}|$. Since $\hat{U}^i_\rho(\lambda) \le U^i(\lambda)$,

$$|\mathcal{W}^i_\rho(\lambda) - U^i(\lambda)| \le |\mathcal{W}^i_\rho(\lambda) - \hat{U}^i_\rho(\lambda)| \le \rho \log|\mathcal{X}| \tag{68}$$

and trivially

$$|\mathcal{W}_\rho(\lambda) - U(\lambda)| \le 2\rho \log|\mathcal{X}|\,, \tag{69}$$

where $\mathcal{W}_\rho(\lambda) = \sum_{i=1}^2 \mathcal{W}^i_\rho(\lambda)$.

Moreover,

$$\mathcal{W}_\rho(\lambda) = \sum_{i=1}^2 \left\langle \theta^i(\lambda), m^i_\rho(\lambda) \right\rangle$$
$$= \langle \theta, \tilde{\mu} \rangle + \sum_{v \in \mathcal{V}} \sum_{x_v \in \mathcal{X}_v} \lambda_v(x_v) \left(m^1_\rho(\lambda)_{v,x_v} - m^2_\rho(\lambda)_{v,x_v} \right)\,, \tag{70}$$

which can be checked by direct comparison of terms of both sides of the equality. From (69) and $P^* = U^*$ follows that

$$|\mathcal{W}_\rho(\lambda^*) - P^*| < 2\rho \log|\mathcal{X}|\,, \tag{71}$$

where λ^* is the point where the optimum of $\hat{U}_\rho(\lambda)$ is attained.

Let us now select $\delta \ge 0$, set up $\rho < \frac{\delta}{2\log|\mathcal{X}|}$ and consider the sequence λ^t generated by Algorithm 1. Let us consider $\tilde{\mu}^{\rho,t}$ given by (51) and (52). From (54), (70) and (71) follows that exist such t^* that starting from it, *i.e.* $\forall t \ge t^*$

$$|\left\langle \theta, \tilde{\mu}^{\rho,t} \right\rangle - P^*| = |P(\tilde{\mu}^{\rho,t}) - P^*| < \delta\,. \tag{72}$$

This completes the proof of the theorem.