Noname manuscript No.
(will be inserted by the editor)

A weighted Mirror Descent algorithm for nonsmooth convex optimization problem

**Duy V.N. Luong · Panos Parpas ·
Daniel Rueckert · Berç Rustem**

Received: date / Accepted: date

Abstract Large scale nonsmooth convex optimization is a common problem for a range of computational areas including machine learning and computer vision. Problems in these areas contain special domain structures and characteristics. Special treatment of such problem domains, exploiting their structures, can significantly improve the computational burden. We present a weighted Mirror Descent method to solve optimization problems over a Cartesian product of convex sets. The algorithm employs a nonlinear weighted distance in the iterative projection scheme. The convergence analysis identifies optimal weighting parameters that, eventually, lead to the optimal weighted step-size strategy for every projection on a corresponding convex set. We demonstrate the efficiency of the algorithm by solving the Markov Random Fields optimization problem. In particular, we use a weighted log-entropy distance and a weighted Euclidean distance. Promising experimental results demonstrate the effectiveness of the proposed method.

Keywords Subgradient Projection · Weighted Distance · Mirror Descent · Markov Random Fields

1 Introduction

It is well known that convex optimization problems can be solved in polynomial time at a low iteration count using interior point methods. However, most of these methods do not scale well with the dimension of an optimization problem. A single iteration cost of an interior point method grows nonlinearly with the problem size. As a result, low iteration count becomes expensive in term of CPU performance. Since what matters most in practice is the overall

Duy V.N. Luong
Department of Computing,
Imperial College London,
E-mail: vu.luong@imperial.ac.uk

computational time to solve the problem, first order methods with computationally low-cost iterations become a viable choice for large scale optimization problems. This paper presents an adaptive first order method to solve a general large scale nonsmooth optimization problem over a Cartesian product of convex sets. Consider the following nonsmooth convex optimization problem:

$$\max_{x \in \mathcal{X}} f(x) \tag{1}$$

where $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times ... \times \mathcal{X}_N$ is the Cartesian product of N closed convex set. In this problem, the decision variable x can be decomposed to N disjoint blocks, where each block $x_i \in \mathcal{X}_i$. In addition, we assume the following for (1):

- The objective function $f : \mathcal{X} \rightarrow \Re$ is a concave Lipschitz continuous function.
- $f^* := f(x^*)$ denotes optimal objective value, where $x^* \in \mathcal{X}$.
- A subgradient $f'(x), \forall x \in \mathcal{X}$ exists.

Our method is a variant of the Mirror Descent (MD) algorithm [1,16,10], an iterative first order approach for nonsmooth optimization problems. The main idea of MD is to adapt a Bregman distance [11] to the feasible domain. In the special case of a single feasible domain $\mathcal{X}$ (i.e. $N = 1$), problem (1) is exactly the problem addressed by the general MD framework. The main difference between the standard MD algorithm and the proposed weighted MD is that we consider the optimal step-size strategy for each projection on the corresponding subset instead of using a common step-size for the projection on the entire domain. In order to achieve this, we employ a weighted distance function for projecting in the corresponding domain. The weigthed distance function exploits the 'disjoint' property of the feasible set $\mathcal{X}$ by considering suitable *weights* α_i for every subset $\mathcal{X}_i$. By assessing the optimality bound for the proposed algorithm, we establish the optimal weighting parameters for each distance function of the corresponding subset. These weighting parameters influence the projection step as scaling factors of the common step-size. Thus, the step-size is scaled appropriately for corresponding subset projection.

As an illustration, we demonstrate the performance of the weighted MD algorithm by solving the Markov Random Fields (MRF) optimization problem [12,18]. This problem often arises from the areas of image analysis and machine learning [15]. We employ the proposed weighted MD algorithm with log-entropy distances and optimal subset-dependent step-sizes to initialize the starting point. Subsequently, we use the weighted MD algorithm with Euclidean distances and incorporate the duality gap in the step-sizes computation. Experimental results demonstrate the superiority of the weighted MD over the basic (unweighted) MD algorithm.

The remainder of this paper focuses on analyzing and describing the proposed weighted MD algorithm and its application to the MRF optimization problem. In the next section, we review the basic MD algorithm and its optimality bound for solving the problem (1). Section 3 introduces required definitions for developing the weighted MD algorithm. These include weighted definitions for distance, compatible norm and dual norm, and Lipschitz constant. By using these definitions, we are able to derive the optimality bound for the proposed weighted MD algorithm and show that it is either an improvement to, or in the worst-case as good as, the standard MD algorithm. In section 4, we consider the dual of the MRF optimization problem. The MRF dual has the form of (1), and therefore can be solved by the proposed weighted MD algorithm. We report very promising computational results.

2 Basic Mirror Descent algorithm

The Mirror Descent algorithm[1,2,10] is a generalization of the projected subgradient method. The standard subgradient approach employs the Euclidean distance function with a suitable step-size in the projection step. Mirror Descent extends the standard projected subgradient method by employing a nonlinear distance function with an optimal step-size in the nonlinear projection step. In this section, we review a basic Mirror Descent algorithm for solving problem (1) without considering the domain geometry.
A basic Mirror Descent algorithm employs a sequence of nonlinear projection:

$$x^{k+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \langle f'_{x^k}, x \rangle - \frac{1}{\mu} \mathrm{D}_{\mathcal{X}}(x, x^k) \quad . \tag{2}$$

where f'_{x^k} is a subgradient at the point x^k, μ is the optimal step-size and $\mathrm{D}_{\mathcal{X}}(x, x^k)$ is a nonlinear distance between two points x and x^k. The set up of Mirror Descent [10] requires a distance function $D(.,.)$ *compatible* with the norm:

- $\|.\|_{\mathcal{X}}$ on the space embedding $\mathcal{X}$, and its dual norm
- $\|\xi\|_{\mathcal{X}*} = \max_{x \in \mathcal{X}} \{\langle x, \xi \rangle : \|x\|_{\mathcal{X}} \leq 1\}$.

Let $\Omega_{\mathcal{X}}$ denote the maximum distance between two points in the set $\mathcal{X}$, i.e.

$$\Omega_{\mathcal{X}} = \max_{x,y \in \mathcal{X}} \mathrm{D}_{\mathcal{X}}(x, y) \quad .$$

Suppose $f(x)$ is Lipschitz continuous on $\mathcal{X}$ with the Lipschitz constant:

$$\mathcal{L}_{\mathcal{X}} = \max_{x \in \mathcal{X}} \|f'_x\|_{\mathcal{X}*} < \infty \quad ,$$

Theorem 1 *[1,10] Let f^* denotes the global optimal objective function and $\bar{x} = \operatorname*{argmax}_{x=\{x^1,..,x^K\}} f(x)$. Then, using the optimal step-size:*

$$\mu = \frac{\sqrt{2\,\Omega_{\mathcal{X}}}}{\mathcal{L}_{\mathcal{X}}\sqrt{K}} \quad , \tag{3}$$

we have the following optimality bound after K iterations:

$$f^* - f(\bar{x}) \leq \frac{\mathcal{L}_{\mathcal{X}}\sqrt{2\,\Omega_{\mathcal{X}}}}{\sqrt{K}} \quad . \tag{4}$$

Remark 1. When $\mathcal{X}$ is the Cartesian product of N convex sets $\mathcal{X}_i, i \in \{1, 2, .., N\}$, the distance between two vectors $x, y \in \mathcal{X}$ is the sum of distances between any two blocks $x_i, y_i \in \mathcal{X}_\text{i}$. As a result, the maximum distance $\Omega_\mathcal{X}$ is also the sum of maximum distances on subset $\mathcal{X}_i$. Let $\Omega_{\mathcal{X}_\text{i}}$ denote the maximum distance of a subset $\mathcal{X}_i$, i.e.,

$$\Omega_{\mathcal{X}_\text{i}} = \max_{x_i, y_i \in \mathcal{X}_i} D(x_i, y_i) \quad ,$$

the optimality bound (4) becomes:

$$f^* - f(\bar{x}) \leq \frac{\mathcal{L}_\mathcal{X} \sqrt{2 \sum_{i=1}^{N} \Omega_{\mathcal{X}_\text{i}}}}{\sqrt{K}} \quad . \tag{5}$$

The projection step (2) employs a common step-size μ for the entire domain. While the feasible domain consists a product set of multiple subsets (each subset might have different characteristics or structures), the basic MD algorithm does not consider a suitable step-size for the projection on each subset. In the next section, we address this scenario.

3 Weighted Mirror Descent

We consider a distance measurement on the given domain (the product set of multiple subsets) as a sum of weighted subset-distances. In this setting, each subset is equipped with a specific distance function and a weighting parameter. We subsequently utilize this weighted distance in the projection step to develop a weighted Mirror Descent algorithm.

3.1 Weigthed distance function

The distance function $\mathrm{D}_\mathcal{X}(x, y)$ is defined as the Bregman distance:

$$\mathrm{D}_\mathcal{X}(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y \rangle \quad ,$$

where $\psi(.)$ needs to be a σ-strongly convex function over a *compatible* norm $\|.\|_\mathcal{X}$, i.e.,

$$\langle \nabla\psi(x) - \nabla\psi(y), x - y \rangle \geq \sigma \|x - y\|_\mathcal{X}^2 \quad , \quad \forall x, y \in \mathcal{X} \tag{6}$$

Without loss of generality, we assume[1] $\sigma = 1$ throughout the paper. A compatible norm $\|.\|_\mathcal{X}$ is dependent of the choice of distance function. For example, l_1-norm is chosen for log-entropy distance [1], l_2-norm for Euclidean distance. Now, instead of using one distance function over the entire domain, let us consider the choice of Bregman distance $\mathrm{D}_{\mathcal{X}_\text{i}}$ for subset $\mathcal{X}_\text{i}, i \in \{1, 2, ..., N\}$:

$$\mathrm{D}_{\mathcal{X}_\text{i}} = \psi^i(x_i) - \psi^i(y_i) - \langle \nabla\psi^i(y_i), x_i - y_i \rangle \quad , \quad \forall x_i, y_i \in \mathcal{X}_\text{i} \tag{7}$$

Various choices of distance functions and their compatible norms $\|.\|_{\mathcal{X}_\text{i}}$ are discussed in [7,8,11]. Two basic examples of Bregman distance are:

[1] Note that Theorem 1 assumes $\sigma = 1$.

- Euclidean distance: $\mathrm{D}_{\mathcal{X}_\mathrm{i}}(x_i, y_i) = \frac{1}{2}\|x_i - y_i\|_2^2$. In this case, $\psi^i(x_i) = \frac{1}{2}\|x_i\|_2^2$ and it is straightforward to show $\psi^i(.)$ is 1-strongly convex w.r.t $\|.\|_2$.
- Log-entropy distance: $\mathrm{D}_{\mathcal{X}_\mathrm{i}}(x_i, y_i) = \sum_j x_i^j \log(x_i^j / y_i^j) + y_i^j - x_i^j$. In this case, $\psi^i(x_i) = \sum_j x_i^j \log x_i^j - x_i^j$ is also shown to be 1-strongly convex w.r.t. $\|.\|_1$ [1].

For each subset-distance $\mathrm{D}_{\mathcal{X}_\mathrm{i}}$ let us introduce a weighting parameter $\alpha_\mathrm{i} > 0$. The weighted distance D_w is then defined as a combination of these weighted subset-distances:

$$\mathrm{D}_\mathrm{w}(x, y) = \sum_{i=1}^{N} \alpha_\mathrm{i}\, \mathrm{D}_{\mathcal{X}_\mathrm{i}}(x_i, y_i) = \sum_{i=1}^{N} \alpha_\mathrm{i}\, \psi^i(x_i) - \alpha_\mathrm{i}\, \psi^i(y_i) - \alpha_\mathrm{i}\, \langle\, \nabla\psi^i(y_i), x_i - y_i \,\rangle$$

We then propose to employ the weighted distance D_w in the projection step (2) instead of the distance $\mathrm{D}_\mathcal{X}$ that does not employ weighting. The weighted Mirror Descent algorithm iteratively computes a search point:

$$x^{k+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \langle f'_{x^k}, x \rangle - \frac{1}{\mu}\, \mathrm{D}_\mathrm{w}(x, x^k) \tag{8}$$

$$x^{k+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \langle f'_{x^k}, x \rangle - \frac{1}{\mu} \sum_{i=1}^{N} \alpha_\mathrm{i}\, \mathrm{D}_{\mathcal{X}_\mathrm{i}}(x_i, x_i^k) \quad .$$

Essentially, the property of $\mathcal{X}$ triggers an ability to independently compute the projection (8) on each subset $\mathcal{X}_\mathrm{i}$. In other words, if we consider the optimality condition of the optimization problem (8) w.r.t. each block $x_i \in \mathcal{X}_\mathrm{i}$, then (8) is separable and is equivalent to:

$$\forall i \in \{1, .., N\} : \quad x_i^{k+1} = \operatorname*{argmax}_{x_i \in \mathcal{X}_\mathrm{i}} \left\langle f'_{x_i^k}, x_i \right\rangle - \frac{\alpha_\mathrm{i}}{\mu}\, \mathrm{D}_{\mathcal{X}_\mathrm{i}}(x_i, y_i) \quad . \tag{9}$$

As a result, we hope to achieve better performance by using suitable (or optimal) weighting pararmeters α_i for the corresponding subset $\mathcal{X}_\mathrm{i}$.

3.2 Compatible norm, dual norm, weighted Lipschitz constant and maximum weighted distance

In order to analyze the convergence of the sequence generated by (8), we need to establish the Lipschitz constant associated with the weighted distance. This can be computed as the upper bound of the *compatible* dual norm. To this end, we rewrite D_w in the form of the Bregman distance:

$$\mathrm{D}_\mathrm{w}(x, y) = \psi_\mathrm{w}(x) - \psi_\mathrm{w}(y) - \langle\, \nabla\, \psi_\mathrm{w}(y), x - y \,\rangle \quad .$$

This, in turn, yields the definition for $\psi_\mathrm{w}(x)$ as a weighted sum of convex function $\psi^i(x_i)$:

$$\psi_\mathrm{w}(x) = \sum_{i=1}^{N} \alpha_\mathrm{i}\, \psi^i(x_i) \quad ,$$

where $\psi_{\mathrm{w}}(.)$ is a convex function and we need to define a *compatible* norm $\|.\|_{\mathrm{w}}$ such that, $\psi_{\mathrm{w}}(.)$ is 1-strongly convex w.r.t. $\|.\|_{\mathrm{w}}$.

Lemma 1 *For all $i \in \{1,..,N\}$, let $\alpha_{\mathrm{i}} > 0, \psi^i(x_i)$ is 1-strongly convex w.r.t. $\|x_i\|_{\mathcal{X}_{\mathrm{i}}}$, then the weighted function:*

$$\psi_{\mathrm{w}}(x) = \sum_{i=1}^{N} \alpha_{\mathrm{i}}\, \psi^i(x_i) \quad ,$$

is 1-strongly convex w.r.t. the weighted norm:

$$\|x\|_{\mathrm{w}} = \sqrt{\sum_{i=1}^{N} \alpha^i \|x_i\|^2_{\mathcal{X}_{\mathrm{i}}}} \quad . \tag{10}$$

Proof We have, $\forall x, y \in \mathcal{X}$:

$$\begin{aligned}\langle \nabla\, \psi_{\mathrm{w}}(x) - \nabla\, \psi_{\mathrm{w}}(y), x - y\rangle &= \sum_{i=1}^{N} \alpha^i \langle \nabla \psi^i(x_i) - \nabla \psi^i(y_i), x_i - y_i\rangle \\ &\geq \sum_{i=1}^{N} \alpha^i \|x_i - y_i\|^2_{\mathcal{X}_{\mathrm{i}}} \\ &= \|x - y\|^2_{\mathrm{w}} \quad .\end{aligned}$$

□

The dual norm $\|.\|_{\mathrm{w}\,*}$ of the proposed weighted norm (10) can be derived using the definition of dual norm (see section 2 and [4]):

$$\|\xi\|_{\mathrm{w}\,*} = \sqrt{\sum_{i=1}^{N} \frac{\|\xi_i\|^2_{\mathcal{X}_{\mathrm{i}}\,*}}{\alpha_{\mathrm{i}}}} \quad , \tag{11}$$

where $\|.\|_{\mathcal{X}_{\mathrm{i}}\,*}$ is a dual norm of $\|.\|_{\mathcal{X}_{\mathrm{i}}}$ over the subset $\mathcal{X}_{\mathrm{i}}$. Let $\mathcal{L}_{\mathcal{X}_{\mathrm{i}}} = \max_{x_i \in \mathcal{X}_{\mathrm{i}}} \|f'_{x_i}\|_{\mathcal{X}_{\mathrm{i}}\,*}$ denote the local Lipschitz constant w.r.t. to a subset $\mathcal{X}_{\mathrm{i}}$, then the weighed Lipschitz constant is given by:

$$\mathcal{L}_{\mathrm{w}} = \max_{x \in \mathcal{X}} \|f'_x\|_{\mathrm{w}\,*} = \sqrt{\sum_{i=1}^{N} \frac{\mathcal{L}_{\mathcal{X}_{\mathrm{i}}}{}^2}{\alpha_{\mathrm{i}}}} \quad . \tag{12}$$

In addition, the maximum weighted distance Ω_{w} becomes:

$$\Omega_{\mathrm{w}} = \max_{x,y \in \mathcal{X}} \mathrm{D}_{\mathrm{w}}(x, y) = \sum_{i=1}^{N} \alpha_{\mathrm{i}}\, \Omega_{\mathcal{X}_{\mathrm{i}}} \quad ,$$

where $\Omega_{\mathcal{X}_{\mathrm{i}}} = \max_{x_i, y_i \in \mathcal{X}_i} \mathrm{D}_{\mathcal{X}_{\mathrm{i}}}(x_i, y_i)$.

Remark 2. The basic MD algorithm in section 2 uses the same distance function definition for all subset $\mathcal{X}_{\mathrm{i}}$ and assigns $\alpha_{\mathrm{i}} = 1$, $\forall i = 1, 2, .., N$. This choice for α_{i} yields the corresponding unweighted Lipschitz constant and maximum distance:

$$\mathcal{L}_{\mathcal{X}} = \sqrt{\sum_{i=1}^{N} {\mathcal{L}_{\mathcal{X}_{\mathrm{i}}}}^2} \quad \text{and} \quad \Omega_{\mathcal{X}} = \sum_{i=1}^{N} \Omega_{\mathcal{X}_{\mathrm{i}}} \tag{13}$$

3.3 Optimality bound of the weighted MD algorithm

We show the first result for optimality bound of the weighted MD algorithm.

Lemma 2 *Let f^* denote the global optimal objective function and $\bar{x} = \underset{x=\{x^1,..,x^K\}}{\operatorname{argmax}} f(x)$ and μ is the step-size. We have the following optimality bound after K iterations:*

$$f^* - f(\bar{x}) \leq \frac{\Omega_{\mathrm{w}}}{K\mu} + \frac{\mu\, {\mathcal{L}_{\mathrm{w}}}^2}{2} \quad . \tag{14}$$

Similar results can be found in [1,10,16]. The initial bound (14) depends on three terms μ, $\mathcal{L}_{\mathrm{w}}$ and Ω_{w}, where the last two terms are themselves functions of the weighting parameters α_{i}. Therefore, we can tighten the bound (14) by considering its minimization w.r.t. μ and α_{i}.

Theorem 2 *For each subset $\mathcal{X}_{\mathrm{i}}$, let $\mathcal{L}_{\mathcal{X}_{\mathrm{i}}} = \max_{x_i \in \mathcal{X}_{\mathrm{i}}} \|f'_{x_i}\|_{\mathcal{X}_{\mathrm{i}}*}$ be the local Lipschitz constant and $\Omega_{\mathcal{X}_{\mathrm{i}}} = \max_{x_i, y_i \in \mathcal{X}_i} \mathrm{D}_{\mathcal{X}_{\mathrm{i}}}(x_i, y_i)$ be the maximum subset distance. Then, the optimal weighting parameters are given by:*

$$\alpha_{\mathrm{i}} = \frac{\mathcal{L}_{\mathcal{X}_{\mathrm{i}}}}{\sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}} \left(\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\mathrm{i}}} \sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}}\right)} \quad , \forall i = 1, 2, ..., N \ . \tag{15}$$

In addition, these parameters yield the optimal step-size:

$$\mu = \frac{\sqrt{2}}{\sqrt{K} \left(\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\mathrm{i}}} \sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}}\right)} \quad . \tag{16}$$

Proof Let us consider the optimality bound (14):

$$\frac{\Omega_{\mathrm{w}}}{K\mu} + \frac{\mu\, {\mathcal{L}_{\mathrm{w}}}^2}{2} ,$$

where Ω_{w} and $\mathcal{L}_{\mathrm{w}}$ are functions of the weighting parameters α_{i}. For any $\alpha_{\mathrm{i}} > 0, \forall i = 1, 2, ..., N$, we can compute the corresponding values Ω_{w} and $\mathcal{L}_{\mathrm{w}}$. For given values of Ω_{w} and $\mathcal{L}_{\mathrm{w}}$, consider the minimization of $\frac{\Omega_{\mathrm{w}}}{K\mu} + \frac{\mu\, {\mathcal{L}_{\mathrm{w}}}^2}{2}$ w.r.t. μ. This yields the optimal step-size dependent on α_{i}:

$$\mu = \frac{\sqrt{2\, \Omega_{\mathrm{w}}}}{\mathcal{L}_{\mathrm{w}} \sqrt{K}} \quad . \tag{17}$$

This has the same form as the optimal step-size (3) for the basic MD algorithm (where $\alpha_{\mathrm{i}} = 1, \forall i = 1, 2, ..., N$). The optimality bound (14) corresponding to (17) is thus given by:

$$\frac{\mathcal{L}_{\mathrm{w}} \sqrt{2\,\Omega_{\mathrm{w}}}}{\sqrt{K}} \quad .$$

For $\alpha = [\alpha^1, \alpha^2, ..., \alpha^N]^\top$, the above optimality bound is a function of α. The best optimality bound can be achieved by considering a minimization of the following function of α:

$$\phi(\alpha) = \mathcal{L}_{\mathrm{w}}{}^2(\alpha)\, \Omega_{\mathrm{w}}(\alpha) = \sum_{i=1}^{N} \frac{\mathcal{L}_{\mathcal{X}_{\mathrm{i}}}{}^2}{\alpha_{\mathrm{i}}} \sum_{i=1}^{N} \alpha_{\mathrm{i}}\, \Omega_{\mathcal{X}_{\mathrm{i}}} \quad .$$

The optimizer of $\phi(\alpha)$ needs to satisfy the following optimality condition:

$$\frac{\alpha_{\mathrm{i}}{}^2\, \Omega_{\mathcal{X}_{\mathrm{i}}}}{\mathcal{L}_{\mathcal{X}_{\mathrm{i}}}{}^2} \sum_{j=1, j\neq i}^{N} \frac{\mathcal{L}_{\mathcal{X}_j}{}^2}{\alpha_j} = \sum_{j=1, j\neq i}^{N} \alpha_j \Omega_{\mathcal{X}_j} \ , \ \forall i = 1, 2, ..., N. \tag{18}$$

Now, let us rewrite the optimality bound $\frac{\Omega_{\mathrm{w}}}{K\mu} + \frac{\mu\, \mathcal{L}_{\mathrm{w}}{}^2}{2}$ in (14) as:

$$\frac{\Omega_{\mathrm{w}}}{K\mu} + \frac{\mu\, \mathcal{L}_{\mathrm{w}}{}^2}{2} = \frac{\sum_{i=1}^{N} \alpha_{\mathrm{i}}\, \Omega_{\mathcal{X}_{\mathrm{i}}}}{K\mu} + \frac{\mu}{2} \sum_{i=1}^{N} \frac{\mathcal{L}_{\mathcal{X}_{\mathrm{i}}}{}^2}{\alpha_{\mathrm{i}}} \quad .$$

Minimizing the RHS of the above equality w.r.t. α_{i} and substituting the optimal step-size $\mu = \frac{\sqrt{2\,\Omega_{\mathrm{w}}}}{\mathcal{L}_{\mathrm{w}} \sqrt{K}}$ in the minimizer gives:

$$\alpha_{\mathrm{i}} = \frac{\mathcal{L}_{\mathcal{X}_{\mathrm{i}}} \sqrt{\Omega_{\mathrm{w}}}}{\mathcal{L}_{\mathrm{w}} \sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}}} \ , \ \forall i = 1, 2, ..., N.$$

Substituting these weighting parameters into the definition of maximum distance $\Omega_{\mathrm{w}} = \sum_{i=1}^{N} \alpha_{\mathrm{i}}\, \Omega_{\mathcal{X}_{\mathrm{i}}}$ yields:

$$\sqrt{\Omega_{\mathrm{w}}} = \frac{\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\mathrm{i}}} \sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}}}{\mathcal{L}_{\mathrm{w}}} \quad .$$

Suppose the weighted distance is normalized by the weighting parameters, i.e. $\Omega_{\mathrm{w}} = 1$, then the weighted Lipschitz constant is given by:

$$\mathcal{L}_{\mathrm{w}} = \sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\mathrm{i}}} \sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}} \tag{19}$$

Using the above weighted Lipschitz constant and the normalized maximum distance, $\Omega_{\mathrm{w}} = 1$, yields the optimal weighting parameters:

$$\alpha_{\mathrm{i}} = \frac{\mathcal{L}_{\mathcal{X}_{\mathrm{i}}}}{\sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}} \left(\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\mathrm{i}}} \sqrt{\Omega_{\mathcal{X}_{\mathrm{i}}}} \right)} \ , \ \forall i = 1, 2, ..., N.$$

We can easily verify that the above choice of $\alpha_{\rm i}$ (also in (15)) normalizes the maximum distance, i.e. $\Omega_{\rm w} = 1$, generates the weighted Lipschitz constant (19) using the definition (12) and satisfies the optimality condition (18) of the optimality bound function $\phi(\alpha)$. □

Theorem 3 *Let f^* denotes the global optimal objective function and $\bar{x} = \underset{x=\{x^1,..,x^K\}}{\text{argmax}} \; f(x)$. The weighted MD algorithm with the optimal step-size* (16) *and the optimal weighting parameters* (15) *has the following optimality bound after K iterations:*

$$f^* - f(\bar{x}) \leq \frac{\sqrt{2}\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\rm i}} \sqrt{\Omega_{\mathcal{X}_{\rm i}}}}{\sqrt{K}} \quad . \tag{20}$$

Proof Substituting the optimal step-size (16) and the optimal weighting parameters (15) into (14) directly yields the result. □

The following result establishes the relative performance of the proposed weighted MD algorithm compared to standard MD. The proposed algorithm is an improvement with a worst-case convergence that is the same as standard MD. The numerical experiments discussed in the next section underline this promising performance.

Corollary 1 *The optimality bound* (20) *of the weighed Mirror Descent algorithm satisfies the inequality:*

$$\frac{\sqrt{2}\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\rm i}} \sqrt{\Omega_{\mathcal{X}_{\rm i}}}}{\sqrt{K}} \leq \frac{\mathcal{L}_{\mathcal{X}} \sqrt{2\,\Omega_{\mathcal{X}}}}{\sqrt{K}} \tag{21}$$

Proof From *Remarks 2*, equations (13), we have:

$$\mathcal{L}_{\mathcal{X}}{}^2\, \Omega_{\mathcal{X}} = \left(\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\rm i}}{}^2\right) \left(\sum_{i=1}^{N} \Omega_{\mathcal{X}_{\rm i}}\right)$$

By the Cauchy-Schwarz inequality, we have:

$$\left(\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\rm i}} \sqrt{\Omega_{\mathcal{X}_{\rm i}}}\right)^2 \leq \left(\sum_{i=1}^{N} \mathcal{L}_{\mathcal{X}_{\rm i}}{}^2\right) \left(\sum_{i=1}^{N} \Omega_{\mathcal{X}_{\rm i}}\right)$$

The above inequality directly yields the result (21), which is also the optimality bound (5) of the standard MD algorithm. □

4 Weighted Mirror Descent algorithm for MRF optimization

Markov Random Fields [15] are an important class of graph-structured models in image processing and machine learning. In general, MRF model aims to reveal hidden quantities ξ based on some observations of available input data. Various discussion about MRF modelling and MRF optimization methods in image analysis and machine learning can be found in [15,12,13,20]. In this paper, we only give a high level description of the MRF model for image analysis. This is illustrated by an undirected graph-structured model. We derive a large scale Linear Programming (LP) formulation for the MRF problem and we focus on solving the dual of this LP using the proposed weighted MD algorithm.

4.1 MRF optimization as Linear Programming

MRF can be described by an undirected graph $G = (V, E)$, where V, E denote a set of nodes and a set of edges respectively. The set V contains unobservable features (e.g. pixel or object) of a given image that needs to be estimated. An unobservable, or hidden, quantity $\xi_{a,l}$, for all $a \in V$, can be assigned a label l from the set of discrete labels L. Each label represents a feasible value (to be estimated) of the corresponding unobservable/hidden feature. Each label assignment is subject to an *input cost* of labelling $\theta_{a,l}$, which encodes how much the assignment of a label $l \in L$ to node $a \in V$ disagrees with the observed image data at the node a. Furthermore, the labelling at a node a also influences its neighbouring nodes. The *neighbouring* nodes are represented by the set of edges E. The neighbouring influences are often known a priori and encoded into the pairwise cost $\theta_{ab,lk}$, where $ab \in E$ is an edge connecting neighbouring nodes a and b, while labels $l, k \in L$ are candidates of the assignment for nodes a, b respectively. The optimal labelling for G can be approximately realised by minimizing the cost of label assignments over all possible combinations of unobserved/hidden quantities and observed image data:

$$\min_{\xi \in \Xi^G} \sum_{a \in V} \sum_{l \in L} \theta_{a,l} . \xi_{a,l} + \sum_{ab \in E} \sum_{l \in L} \sum_{k \in L} \theta_{ab,lk} . \xi_{ab,lk}$$

where Ξ^G is given by:

$$\Xi^G \overset{\text{def}}{=} \left\{ \xi \,\middle|\, \begin{array}{cc} \sum_{l \in L} \xi_{a,l} = 1, & \forall a \in V \\ \sum_{k \in L} \xi_{ab,lk} = \xi_{a,l}, & \forall ab \in E, \forall l \in L \\ \xi_{a,l} \in [0,1], & \forall a \in V, \forall l \in L \\ \xi_{ab,lk} \in [0,1], & \forall ab \in E, \forall l, k \in L \end{array} \right\}. \tag{22}$$

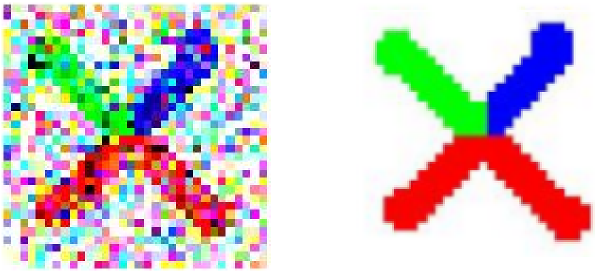
The above LP problem can be written in the following compact form, where θ is a vector of input data and ξ is a vector of decision variables:

$$\min_{\xi \in \Xi^G} \langle \theta, \xi \rangle \tag{23}$$

The above LP relaxation for MRF optimization is shown to be the tightest relaxation amongst other relaxation approaches [13]. The set Ξ^G approximately implies label consistency, such that:

$$\begin{aligned} \xi_{a,l} = 1 &\iff \text{label } l \text{ assigned to node } a \\ \xi_{ab,lk} = 1 &\iff \text{labels } l,k \text{ assigned to the neighbouring } a,b. \end{aligned}$$

A simple application is illustrated using the image segmentation problem, see figure 1. In this example, each image pixel corresponds to a node $a \in V$, whilst a pair of neighbouring pixels forms an edge $ab \in E$. Each node associates with 4 unobserved/hidden quantities, $\xi_{a,l}$, where the label set $L = \{white, red, green, blue\}$. The input data consists of unary costs $\theta_{a,l}$ for each label and pairwise costs $\theta_{ab,lk}$ for each pair of neighbouring nodes. The data cost is specified such that a more likely label assignment is less expensive. The multilabelling problem aims to obtain a label assignment for all nodes such that the overall cost is minimized. This application can be solved approximately by the given LP problem (23).



(a) Corrupted image

(b) Segmented image

Fig. 1 Multilabelling for an image segmentation problem

4.2 Dual decomposition

The MRF optimization problem in image processing and machine learning normally represents every pixel as a node of the graph G. As a result, the LP problem (23) is excessively large, with millions of variables and constraints. Using a standard LP solver such as an interior point method becomes impractical because a single iteration may take too long to compute. It is well-known that certain problems with specific graph-structures can be solved *exactly* by dynamic programming. For example, tree-structured (acyclic) graphs can be solved by max-product belief propagation [21], submodular graph-structured and binary labelling problems can be solved by graph-cut [6,5]. Therefore, one approach to solve the LP problem (23) (i.e. an approximate solution to the MRF problem) is to decompose the graph G associated with the LP into sub-graphs with favourable structures and then combine the exact solutions of these sub-graphs to obtain the solution of the orginal graph. This approach employs the dual decomposition technique in optimization.

Dual decomposition decomposes the original graph-based problem into easier and smaller sub-problems with favourable structures (e.g. submodular or acyclic). The sub-problems arising from the dual decomposition are subsequently combined using the corresponding Lagrange multipliers (dual variables). The dual and the initial primal graph-structured optimization problems are equivalent by convex duality. The focus of this section is the solution of the dual. This is a large scale nondifferentiable optimization problem (1). This is solved by utilizing the exact solution of each sub-problem to update the dual variables. A complete treatment of dual decomposition is beyond the scope of this paper and can be found in [12,17,20] and references therein. Let us assume there exist a set T of sub-graphs that covers (at least once) every node and edge of the original graph G. These sub-graphs should have a favourable graph-structure, e.g. submodular graph [5], acyclic graph [21]. For each sub-graph $t \in T$, the corresponding MRF sub-problem can be solved exactly using a dynamic programming algorithm (max-product belief propagation for acyclic graphs or graph-cut for submodular graphs). The basic idea for decomposing the LP problem (23) is to make copies θ^t (data cost) and ξ^t (decision variables) of the original LP data cost θ and decision variables ξ so that each copy of the pairs (θ^t, ξ^t) forms a MRF sub-problem:

$$\min_{\xi^t \in \Xi^t} \langle \theta^t, \xi^t \rangle$$

where the set Ξ^t has similar form of Ξ^G, but only applies on the copy ξ^t. The copied vector ξ^t corresponds to a sub-graph, it does not need to cover every node or edge of the original graph G. The collection $\{\xi^t\}_{\forall t \in T}$ must cover all nodes and edges of the original graph G. Each copy ξ^t is required to be consistent with the corresponding partition $\xi_{|t}$ of the original decision variables ξ, i.e.

$$\xi^t = \xi_{|t} \ , \ \forall t \in T \quad .$$

Various choices for decomposing (i.e. making copies) the original graph G are discussed in [12]. For the ease of presentation, let us assume each sub-graph $t \in T$ covers all nodes and edges of G, then the above consistency requirement can be written as:

$$\xi^t = \xi \ , \ \forall t \in T \quad .$$

In addition, the sum of copied data-vectors must equal to the original data-vector:

$$\sum_{t \in T} \theta^t = \theta \quad . \tag{24}$$

The equivalent copied problem of the LP (23) is given by:

$$\min_{\xi \in \Xi^G} \langle \, \theta, \xi \, \rangle = \left\{ \min_{\xi \in \Xi^G} \sum \min_{\xi^t \in \Xi^t} \langle \, \theta^t, \xi^t \, \rangle \mid \xi^t = \xi \ , \ \forall t \in T \right\}$$

where $\{\theta^t\}_{\forall t \in T}$ are initialized to satisfy (24). The simplest setting is $\theta^t = \frac{\theta}{\mathcal{T}}$, where $\mathcal{T}$ denotes the cardinality of the set T (the number of sub-graphs in the set T). The copied problem is equivalently difficult to solve as the LP (23) because all decision variables are simultaneously coupled by the constraint $\xi^t = \xi$. Applying the dual decomposition technique to the copied problem directly yields the dual problem:

$$\max_{\lambda \in \Lambda} \sum_{t \in T} \min_{\xi^t \in \Xi^t} \langle\, \theta^t + \lambda^t, \xi^t \,\rangle \tag{25}$$

where $\Lambda \overset{\text{def}}{=} \left\{ \sum_{t \in T} \lambda^t = \mathbf{0} \right\}$. The domain Λ is a Cartesian product of subsets $\{\Lambda_i\}_{\forall i \in I}$, where

$$I \overset{\text{def}}{=} \{(a,l)\}_{\forall a \in V, \forall l \in L} \bigcup \{(ab, lk)\}_{\forall ab \in E, \forall l,k \in L}$$

Each subset is defined as $\Lambda_i \overset{\text{def}}{=} \left\{ \sum_{t \in T} \lambda^t_i = 0 \right\}$, $\forall i \in I$. As a result, $\Lambda = \Lambda_1 \times \Lambda_2 \times ... \times \Lambda_{\mathcal{I}}$, where $\mathcal{I}$ is the cardinality of I. It is well-known that the solution of (25) is the lower bound of the LP problem (23). By strong duality properties, the solution of (25) becomes the solution of the LP (23). Problem (25) is a nonsmooth convex optimization problem over the Cartesian product of convex subsets, it can be written in the exact form of (1):

$$\max_{\lambda \in \Lambda} f(\lambda)$$

There have been several approaches for solving the nonsmooth problem (25). One approach is by Savchynskyy et. al. [18] using Nesterov's smoothing technique. This approach relaxes the nonsmooth objective function by a smoothing parameter. As a result, the algorithm only computes a suboptimal solution of the dual problem, which in turn, does not yield the optimal solution for the LP problem (23). In addition, this algorithm requires computations for all dual variables at every iteration, whilst the weighted MD requires fewer dual updates as the algorithm converges (as we will see in *Remark 3*). Schmidt et. al. [19] proposed a primal-dual method for solving the LP (23), however, their paper shows that the primal-dual method is inferior to the dual decomposition technique for large scale problem. The weighted MD algorithm is a generalization of the projected subgradient algorithm which was also proposed for solving the dual (25) by Komodakis et. al. [12] and Jancsary et. al. [9].

4.3 Weighted MD algorithm for the MRF problem

Problem (25) requires an initialization of θ^t that satisfies (24). The standard initialization $\theta^t = \frac{\theta}{\mathcal{T}}$ might not give a good starting point for subgradient-typed methods. A better initialization is an initialization such that the objective function value is closer to the optimal objective value. Suppose we have a better initialization θ^{t*}, we can reduce the computational efforts for solving λ

significantly. To this end, let us introduce the following optimization problem to find a better initialization for θ^t:

$$\max_{\rho\in\varDelta} f(\rho) \overset{\text{def}}{=} \max_{\rho\in\varDelta}\sum_{t\in T}\min_{\xi^t\in\varXi^t}\langle\,\rho^t\circ\theta\,,\,\xi^t\,\rangle \tag{26}$$

where $\circ$ is a Hadamard product notation, $\varDelta = \varDelta_1 \times \varDelta_2 \times \ldots \times \varDelta_{\mathcal{I}}$ is the product set of simplices:

$$\varDelta_i \overset{\text{def}}{=} \left\{\rho_i \,\middle|\, \sum_{t\in T}\rho_i^t = 1\ ;\ \rho_i^t \geq 0\,,\,\forall t\in T\right\}\ ,\ \forall i\in I \quad . \tag{27}$$

Problem (26) also has the same form as (1) and can be solved using the weighted MD algorithm. After obtaining the optimal initialization $\{\rho^{t*}\circ\theta\,,\,\forall t\in T\}$, where $\rho^* = \text{argmax}_{\rho\in\varDelta} f(\rho)$, we can proceed to solve for λ:

$$\max_{\lambda\in\varLambda} f(\lambda) \overset{\text{def}}{=} \max_{\lambda\in\varLambda}\sum_{t\in T}\min_{\xi^t\in\varXi^t}\langle\,\rho^{t*}\circ\theta+\lambda^t,\xi^t\,\rangle \tag{28}$$

where $\varLambda = \varLambda \times \varLambda \times \ldots \times \varLambda_{\mathcal{I}}$ is the product set of linear subsets:

$$\varLambda_i \overset{\text{def}}{=} \left\{\lambda_i \,\middle|\, \sum_{t\in T}\lambda_i^t = 0\right\}\ ,\ \forall i\in I \tag{29}$$

The two problems (26) and (28) can be combined into one problem:

$$\max_{\rho\in\varDelta,\lambda\in\varLambda} f(\rho,\lambda) \overset{\text{def}}{=} \max_{\rho\in\varDelta,\lambda\in\varLambda}\sum_{t\in T}\min_{\xi^t\in\varXi^t}\langle\,\rho^t\circ\theta+\lambda^t,\xi^t\,\rangle \tag{30}$$

By setting $\lambda = 0$, we have (30) $\equiv$ (26). Similarly, if we set $\rho^{t*} = \text{argmax}_{\rho\in\varDelta} f(\rho)$ then we have (30) $\equiv$ (28).

Algorithm 1: Weighted Mirror Descent for the MRF problem

```
Choose two nonnegative numbers $K_1, K_2$;
Initialize $\rho^1 = \frac{1}{T}.\mathbf{1}$ and $\lambda^1 = \mathbf{0}$;
for $k = 1, 2, ..., K_1 - 1$ do
    $\rho^{k+1} = \underset{\rho\in\varDelta}{\text{argmax}}\,\langle\, f'_{\rho^k}, \rho\,\rangle - \frac{1}{\tau}\,\mathrm{D}_\Delta(\rho, \rho^k)$ .    (31a)
Set $\bar{\rho} = \underset{\rho}{\text{argmax}}\,\{f(\rho, \lambda^1)\,|\rho = \rho^1, \rho^2, ..., \rho^{K_1}\,\}$;
for $k = 1, 2, ..., K_2 - 1$ do
    $\lambda^{k+1} = \underset{\lambda\in\varLambda}{\text{argmax}}\,\langle\, f'_{\lambda^k}, \lambda\,\rangle - \frac{1}{\eta}\,\mathrm{D}_\Lambda(\lambda, \lambda^k)$ .    (31b)
Set $\bar{\lambda} = \underset{\lambda}{\text{argmax}}\,\{f(\bar{\rho}, \lambda)\,|\lambda = \lambda^1, \lambda^2, ..., \lambda^{K_2}\,\}$;
```

The weighted MD approach for solving the MRF problem is described in Algorithm 1. As we will see later (equation (39)), exact subset-dependent step-sizes can be computed for the recurrence (31a). However, recurrence (31b) can only use estimate step-sizes. A step-size estimation is based on the difference between the current objective value and the optimal objective value. The smaller this difference is, the more accurate estimation can be made. Clearly, if the number of iterations K_1 is large enough, we will obtain an objective value that is better (closer to optimal objective value) than a random initialization.

We clarify the various aspects of the vector ρ (similarly applies for λ):

- ρ denotes a full vector corresponding to all sub-graphs of the set T, and $\rho \in \varDelta$.
- With superscipt t, ρ^t denotes a vector corresponding to sub-graph $t \in T$.
- With subscript i, ρ_i denote a collection of scalars ρ_i^t across all sub-graphs that cover the index i, and $\rho_i \in \varDelta_i$.
- With numeric superscipts, such as $\rho^1, \rho^2, .., \rho^K$, or ρ^k, ρ_i^k, denote the corresponding iterate of the vector.
- When superscipts t and k are used together, we separate them by a comma: $\rho^{t,k}$ is a vector, or $\rho_i^{t,k}$ is a scalar.

Recurrence (31a) and (31b) seek for feasible points in the domains that are intersections of subsets. As a result, we employ weighted distances for both sequences. The subset-projections (also see (9)) for these recurrences can be written as:

$$\forall i \in I: \quad \rho_i^{k+1} = \underset{\rho_i \in \varDelta_i}{\operatorname{argmax}} \left\langle f'_{\rho_i^k}, \rho_i \right\rangle - \frac{\alpha_{\Delta_\mathrm{i}}}{\tau} \mathrm{D}_{\Delta_\mathrm{i}}(\rho_i, \rho_i^k) \quad . \tag{32a}$$

$$\forall i \in I: \quad \lambda_i^{k+1} = \underset{\lambda_i \in \varLambda_i}{\operatorname{argmax}} \left\langle f'_{\lambda_i^k}, \lambda_i \right\rangle - \frac{\alpha_{\Lambda_\mathrm{i}}}{\eta} \mathrm{D}_{\Lambda_\mathrm{i}}(\lambda_i, \lambda_i^k) \quad . \tag{32b}$$

To this end, we choose the log-entropy distance function for each subset $\varDelta_i$ and the Euclidean distance function for each subset $\varLambda_i$. In particular, let us consider:

- For each $\varDelta_i$, let:

$$\psi_\varDelta^i(\rho_i) = \sum_{t \in T} \rho_i^t \log \rho_i^t, \text{ if } \rho_i \in \varDelta_i;\ else, +\infty \quad ,$$

  then $\psi_\varDelta^i$ is 1-strongly convex [1, Proposition 5.1] w.r.t. $\|.\|_1$. The dual norm of $\|.\|_1$ is $\|.\|_\infty$ [4].
- For each $\varLambda_i$, let:

$$\psi_\varLambda^i(\lambda_i) = \frac{1}{2} \sum_{t \in T} (\lambda_i^t)^2, \text{ if } \lambda_i \in \varLambda_i;\ else, +\infty$$

  then $\psi_\varLambda^i$ is 1-strongly convex w.r.t. $\|.\|_2$. The dual norm of $\|.\|_2$ is itself.

By using the Bregman distance: $\mathrm{D}(x,y) = \psi(x) - \psi(y) - \langle \nabla\psi_y, x-y \rangle$, we can obtain the log-entropy distance function and the Euclidean distance function for the corresponding subset. As a result, each iteration of the recurrences (32) can be solved in a closed form:

$$\forall i \in I: \quad \rho_i^{t,k+1} = \frac{\rho_i^{t,k}.\exp\left(\frac{\tau}{\alpha_{\Delta_\mathrm{i}}}.f'_{\rho_i^{t,k}}\right)}{\sum_{t\in T}\left(\rho_i^{t,k}.\exp\left(\frac{\tau}{\alpha_{\Delta_\mathrm{i}}}.f'_{\rho_i^{t,k}}\right)\right)} \quad . \tag{34a}$$

$$\forall i \in I: \quad \lambda_i^{t,k+1} = \frac{\eta}{\alpha_{\Lambda_\mathrm{i}}}\left(f'_{\lambda_i^{t,k}} - \frac{\sum_{t\in T} f'_{\lambda_i^{t,k}}}{\mathcal{T}}\right) \quad . \tag{34b}$$

We note that the basic MD algorithm also uses the above recurrences with the constant choice $\alpha_{\Delta_\mathrm{i}} = \alpha_{\Lambda_\mathrm{i}} = 1$, $\forall i \in I$. Using the definitions of optimal step-size (16) and weighting pararmeters (15), the two subset-dependent step-sizes $\frac{\tau}{\alpha_{\Delta_\mathrm{i}}}$ and $\frac{\eta}{\alpha_{\Lambda_\mathrm{i}}}$ can be written as:

$$\frac{\tau}{\alpha_{\Delta_\mathrm{i}}} = \frac{\sqrt{2\,\Omega_{\Delta_\mathrm{i}}}}{\mathcal{L}_{\Delta_\mathrm{i}}\sqrt{k}} \quad \text{and} \quad \frac{\eta}{\alpha_{\Lambda_\mathrm{i}}} = \frac{\sqrt{2\,\Omega_{\Lambda_\mathrm{i}}}}{\mathcal{L}_{\Lambda_\mathrm{i}}\sqrt{k}} \quad . \tag{35}$$

The above subset-dependent step-sizes improve the performance of the weighted MD because they use optimal values of $\alpha_{\Delta_\mathrm{i}}$ and $\alpha_{\Lambda_\mathrm{i}}$ instead of the constant 1. It thus remains to show how to compute the subgradients f'_ρ and f'_λ at any feasible $\rho \in \varDelta$ and $\lambda \in \varLambda$.

Lemma 3 *Let* $\bar{\xi}^t = \operatorname*{argmin}_{\xi^t\in\varXi^t}\langle\rho^t\circ\theta+\lambda^t,\xi^t\rangle$ *be the optimal solution for the MRF sub-problem of the corresponding sub-graph* $t \in T$*. Then the subgradients of* $f(\rho,\lambda)$ *w.r.t. the corresponding decision vector are given by:*

$$f'_{\rho^t} = \theta\circ\bar{\xi}^t \quad \textit{and} \quad f'_{\lambda^t} = \bar{\xi}^t \quad .$$

Proof Let x, y be arbitrary vectors such that $x \in \varDelta$ and $y \in \varLambda$. By definition $\bar{\xi}^t$ is not necessarily optimal for $\min_{\xi^t\in\varXi^t}\langle x^t\circ\theta + y^t, \xi^t\rangle$, i.e.

$$\forall t \in T: \quad \min_{\xi^t\in\varXi^t}\langle x^t\circ\theta + y^t, \xi^t\rangle \le \langle x^t\circ\theta + y^t, \bar{\xi}^t\rangle\,.$$

In addition,

$$\begin{aligned} f(x,y) &= \sum_{t\in T}\min_{\xi^t\in\varXi^t}\langle x^t\circ\theta + y^t, \xi^t\rangle \\ &\le \sum_{t\in T}\langle x^t\circ\theta + y^t, \bar{\xi}^t\rangle \\ &= \sum_{t\in T}\langle\rho^t\circ\theta + \lambda^t, \bar{\xi}^t\rangle + \langle\theta\circ\bar{\xi}^t, x^t - \rho^t\rangle + \langle\bar{\xi}^t, y^t - \lambda^t\rangle \\ &= F(\rho,\lambda) + \langle\theta\circ\bar{\xi}, x-\rho\rangle + \langle\bar{\xi}, y-\lambda\rangle. \end{aligned}$$

□

Remark 3. The above choices of subgradient rely on the exact solution $\bar{\xi}^t \in \Xi^{\mathcal{I}}$ for each sub-graph t (that can be computed very efficiently by a dynamic programming algorithm, e.g. max-product belief propagation or graph-cut). Using these subgradients, we can verify that updates (34) are only needed at *disagreement nodes* [2]. As a result, we can utilize this property to define a stopping criteria by counting the number of disagreement nodes. Let L_k be the number of disagreement nodes at iteration k. Essentially, as $L_k \to 0$, the algorithm converges to a stationary point, i.e. the optimal solution.

By using the above subgradients and the fact that $\bar{\xi}_i^t \in [0,1]$, we can derive the local Lipschitz constants corresponding to their subsets, $\forall i \in I$:

$$\mathcal{L}_{\Delta_\mathrm{i}} = \sup_{\rho_i \in \varDelta_i} \|f'_{\rho_i}\|_\infty = |\theta_i| \quad \text{and} \quad \mathcal{L}_{\Lambda_\mathrm{i}} = \sup_{\lambda_i \in \varLambda_i} \|f'_{\lambda_i}\|_2 = \sqrt{\mathcal{T}} \tag{36}$$

To specify the maximum subset distances, we need to find an upper bound for the distance between any feasible point to starting points ρ_i^1 and λ_i^1.

Lemma 4 *Let all elements of starting point $\rho_i^{t,1} = \frac{1}{\mathcal{T}}$, the upper bound of the distance between any feasible vector and ρ_i^1 is given by:*

$$\Omega_{\Delta_\mathrm{i}} = \log \mathcal{T} \tag{37}$$

Proof Using the Bregman distance (7) with log-entropy function $\psi_\varDelta^i(\rho_i) = \sum_{t\in T} \rho_i^t \log \rho_i^t$ for every subset $\varDelta_i$, $i \in \mathcal{I}$, we have:

$$\begin{aligned}\mathrm{D}_{\Delta_\mathrm{i}}(\rho_i, \rho_i^1) &= \sum_{t\in T} \rho_i . \log \frac{\rho_i}{\rho_i^{t,1}} = \sum_{t\in T} \rho_i^t \log \rho_i^t + \left(\sum_{t\in T} \rho_i^t\right) \log \mathcal{T} \\ &\leq \left(\sum_{t\in T} \rho_i^t\right) \log \mathcal{T} \leq \log \mathcal{T}\end{aligned}$$

The last two inequalities follow from the facts that $0 \leq \rho_i^t \leq 1$, therefore $\log \rho_i^t \leq 0$; and $\sum_{t\in T} \rho_i^t = 1$. □

Similarly, the Bregman distance with $\psi_\varLambda^i(\lambda_i) = \frac{1}{2}\sum_{t\in T}(\lambda_i^t)^2$ yields the Euclidean distance corresponding to subset $\varLambda_i$, thus the quantity $\Omega_{\Lambda_\mathrm{i}}$ is given by (with $\lambda_i^1 = \mathbf{0}$):

$$\Omega_{\Lambda_\mathrm{i}} = \max_{\lambda_i \in \varLambda_i} \frac{1}{2}\|\lambda_i - \lambda_i^1\|_2^2 = \max_{\lambda_i \in \varLambda_i} \frac{1}{2}\|\lambda_i\|_2^2$$

The subset $\varLambda_i$ defined in (29) does not allow exact computation for $\Omega_{\Lambda_\mathrm{i}}$. For example, assume the index $i \in I$ is covered by two sub-graphs $t_1, t_2 \in T$, then

$$2\,\Omega_{\Lambda_\mathrm{i}} = \max_{\lambda_i^{t_1}+\lambda_i^{t_2}=0} \|\lambda_i\|_2^2 = \max_{\lambda_i^{t_1}+\lambda_i^{t_2}=0} (\lambda_i^{t_1})^2 + (\lambda_i^{t_2})^2$$

[2] A node $a \in V$ is a *disagreement node* if all sub-graphs do not assign the same label to a, i.e. for any two sub-graphs $t_1, t_2 \in T$, there exists $l \in L$ such that $\bar{\xi}_{a,l}^{t_1} \neq \bar{\xi}_{a,l}^{t_2}$.

In theory, $2\,\Omega_{\Lambda_{\rm i}}$ can be infinitely large, thus, the step-size $\frac{\eta}{\alpha_{\Lambda_{\rm i}}}$ also becomes infinitely large. In this problem, we assume subset Λ_i is bounded and solutions exist. Therefore, we estimate $\Omega_{\Lambda_{\rm i}}$ by a quantity that is proportional to the distance between the solution λ_i^* and the starting point $\lambda_i^1 = \mathbf{0}$. The approximate duality gap is a good heuristic to estimate how far the current iterate is from the optimal solution.

In order to estimate the duality gap at iteration k, we need to compute (approximately) the primal value $P(\xi^k) = \langle\, \theta, \xi^k \,\rangle$. Several approaches to estimate the primal variables are discussed in [12]. We employ the ergodic sequence of dual subgradients f'_{λ^k} to estimate the primal variables. Ergodic convergence analysis [14] has been used by many authors to bridge the primal-dual gap in convex optimisation. In the approach, primal variables ξ^k are estimated by considering the weighted average of the dual subgradients over all iterations:

$$\xi^K = \frac{\sum_{k=1}^{K} \sum_{t\in T} f'_{\lambda^{t,k}}}{K} = \frac{\sum_{k=1}^{K} \sum_{t\in T} \bar{\xi}^{t,k}}{K} \quad .$$

The approximate duality gap is given by:

$$|P(\xi^K) - f(\bar{\rho}, \lambda^K)| \quad ,$$

which can be used as a heuristic to estimate $\Omega_{\Lambda_{\rm i}}$ at iteration k:

$$\Omega_{\Lambda_{\rm i}} = \frac{|P(\xi^k) - f(\bar{\rho}, \lambda^k)|}{2L_k} \quad . \tag{38}$$

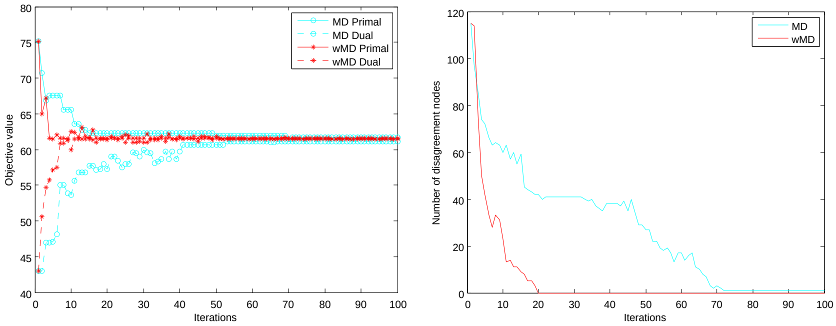
where L_k is the number of disagreement nodes (see *Remark 3*). Substituting local Lipschitz constants (36) and subset distances (37),(38) into the subset-dependent step-sizes (35) yields:

$$\frac{\tau}{\alpha_{\Delta_{\rm i}}} = \frac{\sqrt{2\log(\mathcal{T})}}{|\theta_i|\sqrt{k}} \quad \text{and} \quad \frac{\eta}{\alpha_{\Lambda_{\rm i}}} = \sqrt{\frac{|P(\xi^k) - f(\bar{\rho}, \lambda^k)|}{L_k\,\mathcal{T}\,k}} \quad . \tag{39}$$

Relating the step-size $\frac{\eta}{\alpha_{\Lambda_{\rm i}}}$ to the duality gap allows the algorithm to admits large step-sizes (as a result, applies large changes) when the duality gap is large (far from the optimum). As the duality gap reduces, so does the step-size. This choice of step-size is consistent with the diminishing step-size approach that guarantees convergence for subgradient methods [3].

4.4 Experiments

In order to demonstrate the effectiveness of our method, we present experimental results for two MRF problems. The first is a graph structure optimisation problem with synthetic data. The second is an image segmentation problem. In the first experiment, we apply the weighted Mirror Descent (wMD), and the



(a) Convergence rate

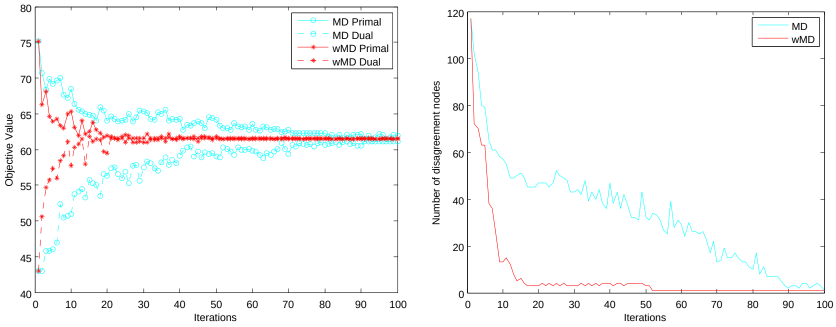
(b) Number of disagreement nodes

Fig. 2 Synthetic data: Potts model

standard Mirror Descent algorithm (MD). Both methods performs 20 iterations of log-entropy projections (i.e. $K_1 = 20$ in Algorithm 1) before switching to Euclidean projections. The standard MD algorithm employs unweighted distance function, i.e. assign all weighting parameters $\alpha_i = 1$. In the second, we use the image segmentation implementation from the UGM Matlab package [19]. The provided package also implements the Tree Reweighted Belief Propagation (TRBP) which is one of the state-of-the-art dynamic programming algorithms for MRF opitimisation (however, its global convergence is not guaranteed [12]). TRBP only returns the primal objective value, therefore, we use the primal objective values of TRBP as reference to compare with the primal objective values of wMD and MD.

Synthetic data. For synthetic experiments, we use a graph of size 100×100 and 5 labels. Two popular methods to set up a synthetic MRF problem are based on the Potts model and the uniform model. Let $\mathcal{U}(0,1)$ and $\mathcal{N}(0,1)$ denote the uniform distribution and normal distribution respectively. In the Potts model, $\theta_{a,l} \sim \mathcal{U}(0,1)$, while $\theta_{ab,lk} = \omega_{ab} * \mathbb{I}(l = k)$, where $\omega_{ab} \sim \mathcal{N}(0,1)$ and $\mathbb{I}(l = k) = 1 \iff l = k$ and $\mathbb{I}(l = k) = 0 \iff l \neq k$. In the uniform model, $\theta_{a,l} \sim \mathcal{U}(0,1)$ and $\theta_{ab,lk} = \omega_{ab}.\gamma_{ab}$, where $\omega_{ab} \sim \mathcal{N}(0,1)$ and $\gamma_{ab} \sim \mathcal{U}(0,1)$.

Figure 2(a) shows the convergence for the Potts model. The two algorithms compute a pair of dual and primal values at each iteration. The optimal solutions are achieved when the duality gap vanishes. As the duality gap decreases, the number of disagreement nodes reduces, see Figure 2(b). For the uniform model, the corresponding graphs are shown in Figure 3. In addition to the convergence rate, what matters most is the time required to compute the solution. We generate 1000 random simulations for the uniform model with graphs of size 100×100, 500×500 and 1000×1000. All graphs recover a solution for the MRF graph problem with 5 discrete labels. The average computational time is presented in the boxplot Figure 4.



(a) Convergence rate (b) Number of disagreement nodes

Fig. 3 Synthetic data: Uniform model

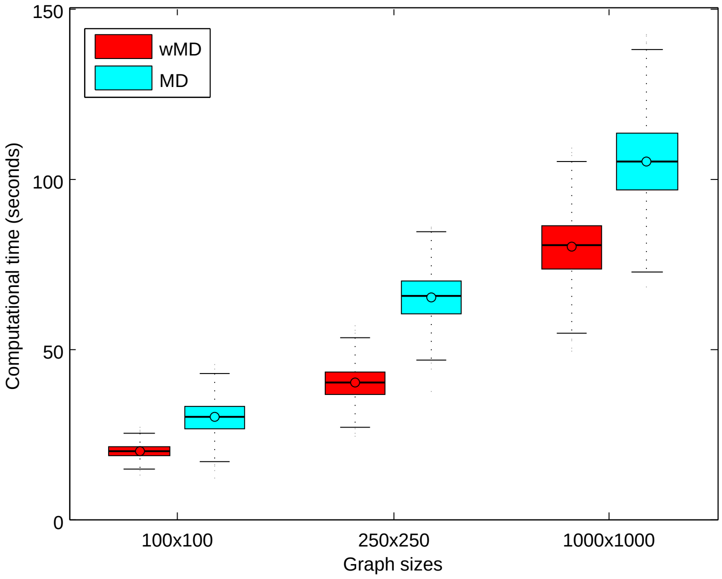


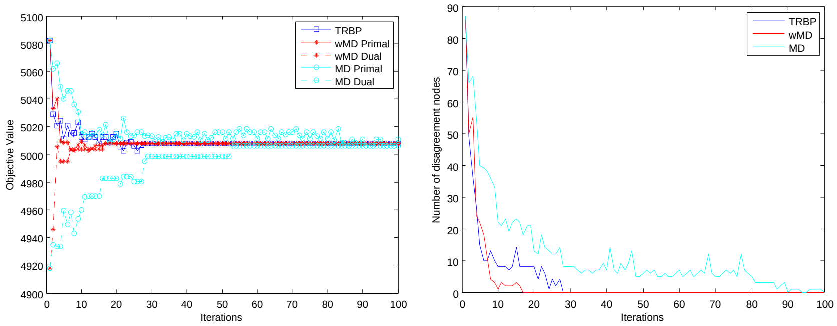
Fig. 4 Computational time for 1000 random simulations of uniform model.

Image segmentation problem. The image segmentation problem aims to allocate every pixel to the best corresponding label, see Figure 1. There are 4 input labels: white, blue, red and green. The unary potentials are defined by the cost to assign a label $l \in L$ to a pixel $I(a)$, for example, one way of defining this cost is:

$$\theta_{a,l} = -\log p(I(a)|a = l) \quad \forall a \in V,\ \forall l \in L$$

where $p(.)$ is a known probability distribution. The pairwise potentials are computed to penalise the differing label assignment of neighbouring pixels,

$$\theta_{ab,lk} = \exp\left(-\frac{|I(a) - I(b)|}{\sigma^2}\right) \cdot \frac{1}{\|l-k\|} . \mathbb{I}(l = k) \quad \forall ab \in E,\ \forall l, k \in L$$



(a) Convergence rate (b) Number of disagreement nodes

Fig. 5 Image segmentation: convergence properties

where σ corresponds to the level of noise in the image. Figure 5 demonstrates the performance of three methods: TRBP, wMD and MD.

5 Conclusion

An efficient algorithm is presented for solving a large scale nonsmooth convex problem. The method is based on the Mirror Descent algorithm employing a suitable weighted distance function. By assessing the optimality bound of the proposed algorithm, we are able to compute the optimal subset-dependent step-sizes. This yields a convergence rate that is not worse than the standard MD algorithm. The experimental results for MRF optimization problems confirm the improved performance.

Conflict of Interest. We acknowledge a partial support of the EPSRC award EP/I014640/1 for the author Duy. V.N. Luong. We declare that we have no other conflict of interest.

References

1. Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Operations Research Letters (2003)
2. Ben-tal, A., Margalit, T., Nemirovski, A.: The ordered subsets mirror descent optimization method with applications to tomography. SIAM Journal on Optimization **12**, 2001 (2001)
3. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific (1999)
4. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
5. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. International Journal of Computer Vision **70**, 109–131 (2006)
6. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**, 1222 –1239 (2001)
7. Censor, Y., Zenios, S.A.: Proximal minimization algorithm with d-functions. Journal of Optimization Theory and Applications **73** (1992)
8. Chen, G., Teboulle, M.: Convergence analysis of a proximal-like minimization algorithm using bregman functions. SIAM Journal on Optimization **3**, 538–543 (1993)

9. Jancsary, J., Matz, G.: Convergent decomposition solvers for tree-reweighted free energies. Journal of Machine Learning Research
10. Juditsky, A., Nemirovski, A.: First order methods for nonsmooth convex large-scale optimization, i: General purpose methods. Optimization for Machine Learning, chap. 5. The MIT Press (2012)
11. Kiwiel, K.C.: Proximal minimization methods with generalized bregman functions. SIAM J. Control Optim. **35**(4) (1997)
12. Komodakis, N., Paragios, N., Tziritas, G.: Mrf energy minimization and beyond via dual decomposition. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 531–552 (2011)
13. Kumar, M.P., Kolmogorov, V., Torr, P.H.S.: An analysis of convex relaxations for map estimation of discrete mrfs. Journal of Machine Learning Research **10**, 71–106 (2009)
14. Larsson, T., Patriksson, M., Stromberg, A.: Ergodic primal convergence in dual subgradient schemes for convex programming. Mathematical Programming **86** (1999)
15. Li, S.Z.: Markov Random Field Modelling in Image Analysis. Advances in Computer Vision and Pattern Recognition. Springer-Verlag (2009)
16. Nemirovski, A., Yudin, D.: Problem complexity and Method Efficiency in Optimization. Wiley (1983)
17. Rush, A.M., Collins, M.: A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. J. Artif. Int. Res. **45** (2012)
18. Savchynskyy, B., Schmidt, S., Kappes, J., Schnorr, C.: A study of nesterov's scheme for lagrangian decomposition and map labeling. In: Computer Vision and Pattern Recognition, pp. 1817–1823 (2011)
19. Schmidt, M.: Ugm: Matlab code for undirected graphical models (2011). URL http://www.di.ens.fr/ mschmidt/Software/UGM.html
20. Sontag, D., Globerson, A., Jaakkola, T.: Introduction to dual decomposition for inference. In: S. Sra, S. Nowozin, S.J. Wright (eds.) Optimization for Machine Learning. MIT Press (2011)
21. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. In: Exploring artificial intelligence in the new millennium, pp. 239–269 (2003)