Globally Convergent Parallel MAP LP Relaxation Solver using the Frank-Wolfe Algorithm

Alexander G. Schwing ASCHWING@CS.TORONTO.EDU
University of Toronto, 10 King's College Rd., Toronto, Canada
Tamir Hazan TAMIR@CS.HAIFA.AC.IL
University of Haifa, Haifa, Israel
Marc Pollefeys MARC.POLLEFEYS@INF.ETHZ.CH
ETH Zurich, Universitätstrasse 6, Zurich, Switzerland
Raquel Urtasun URTASUN@CS.TORONTO.EDU
University of Toronto, 10 King's College Rd., Toronto, Canada

Abstract

Estimating the most likely configuration (MAP) is one of the fundamental tasks in probabilistic models. While MAP inference is typically intractable for many real-world applications, linear programming relaxations have been proven very effective. Dual block-coordinate descent methods are among the most efficient solvers, however, they are prone to get stuck in sub-optimal points. Although subgradient approaches achieve global convergence, they are typically slower in practice. To improve convergence speed, algorithms which compute the steepest ϵ-descent direction by solving a quadratic program have been proposed. In this paper we suggest to decouple the quadratic program based on the Frank-Wolfe approach. This allows us to obtain an efficient and easy to parallelize algorithm while retaining the global convergence properties. Our method proves superior when compared to existing algorithms on a set of spin-glass models and protein design tasks.

1. Introduction

Graphical models are typically employed to describe the dependencies between variables involved in a joint probability distribution. Finding the most likely configuration, *i.e.*, the maximum a-posteriori (MAP) assignment, is one of the most fundamental inference tasks. Unfortunately computing the MAP is NP-hard for many applications.

In recent years linear programming (LP) relaxations have been shown to retrieve globally optimal configurations in many cases. The large amount of variables and constraints involved in practical applications poses significant challenges to standard LP solvers. Development of particularly tailored algorithms has therefore become a vibrant area of research. Since the graphical model structure is encoded in the constraints, algorithms that optimize the dual are employed to take into account the problem setup. Existing solvers, however, have difficulties due to the non-smoothness of the dual. For example, block coordinate descent algorithms monotonically decrease the objective and converge very fast, but, they are not guaranteed to reach the global optimum of the dual program. To overcome this sub-optimality problem, different solutions have been proposed, *e.g.*, smoothing (Johnson, 2008; Jojic et al., 2010; Hazan & Shashua, 2010; Savchynskyy et al., 2012), proximal updates (Ravikumar et al., 2010) and augmented Lagrangian methods (Martins et al., 2011; Meshi & Globerson, 2011). Due to the modifications of the cost function, convergence speed is however reduced.

Globally convergent methods using the original cost function employ subgradients (Komodakis et al., 2010). Bundle methods (Lemaréchal, 1974; Kappes et al., 2012) are alternatively considered. However, applying sub-gradients directly or indirectly is known to converge slowly since arbitrary gradient directions are employed.

Recently Schwing et al. (2012) proposed a steepest ϵ-descent approach that monotonically decreases the dual objective and reaches the global optimum of the dual program. Contrasting arbitrary subgradient directions, their work advertises usage of the steepest ϵ-descent direction, which is found by solving a computationally expensive quadratic program. In this paper we propose to solve the quadratic program efficiently using a Frank-Wolfe approach, also known as conditional gradient. The benefits are twofold: we no longer require a general purpose solver to find the steepest ϵ-descent directions, which results in better efficiency and allows usage of sub-optimal directions that do not point exactly into the steepest descent direction. Furthermore, the task decouples and is easily parallelized.

We demonstrate the effectiveness of our approach on pro-

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

Repeat until convergence, for every region r:

$$\forall s_r, p \in P(r)$$

$$\mu_{p\to r}(s_r) = \max_{s_p\setminus s_r}\left\{\theta_r(s_r) + \sum_{p\in P(r)}\lambda_{r\to p}(s_r) - \sum_{r'\in C(p)\setminus r}\lambda_{r'\to p}(s_{r'})\right\}$$

$$\forall s_r, p \in P(r)$$

$$\lambda_{r\to p}(s_r) = \frac{1}{1+|P(r)|}\left(\theta_r(s_r) + \sum_{c\in C(r)}\lambda_{c\to r}(s_c) + \sum_{p\in P(r)}\mu_{p\to r}(s_r)\right) - \mu_{p\to r}(s_r)$$

Figure 1. Standard convex max-product message passing algorithm.

tein design tasks as well as on spin glass models, and show that it outperforms existing state-of-the-art algorithms. In the remainder of the paper, we first provide some background regarding MAP estimation. We then detail our Frank-Wolfe approach and present experimental results before discussing related work and conclusions.

2. Background

Graphical models encode joint distributions over product spaces $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, where we assume the domains $\mathcal{S}_i$, $i \in \{1, \ldots, n\}$ to be discrete. The joint probability is commonly defined by summing terms $\theta_r(s_r)$ often also referred to as negative energy functions. Each term depends on a restriction of the variables to subsets or regions $r \subseteq \{1, \ldots, n\}$, *i.e.*, $s_r = (s_i)_{i\in r} \in \prod_{i\in r}\mathcal{S}_i$. We use $\mathcal{R}$ to denote the set of all regions. The joint distribution is then given by

$$p(s) \propto \exp\left(\sum_{r\in\mathcal{R}}\theta_r(s_r)\right).$$

We focus on estimating the maximum a-posteriori (MAP) configuration, *i.e.*, we aim at finding the assignment that maximizes the probability $p(s)$. Estimating the MAP configuration is equivalently written as a program of the following form (Koller & Friedman, 2009):

$$\arg\max_{s_1,\ldots,s_n}\sum_{r\in\mathcal{R}}\theta_r(s_r). \tag{1}$$

Due to its combinatorial nature, this problem is NP-hard for general graphical models. It is tractable only in some special cases such as tree structured graphs, where specialized dynamic programming algorithms (*e.g.*, max-product belief propagation) are guaranteed to recover the optimum. Another notable example are graphs which contain only sub-modular second order functions, where graph-cut approaches yield the global optima. In this paper we are, however, interested in the general case, employing arbitrary graphs and energy functions.

The MAP program in Eq. (1) has a linear form, thus it is naturally represented as an integer linear program (ILP). Many researchers (*e.g.*, Schlesinger (1976)) have proposed a relaxation of the ILP by replacing the integrality constraints with non-negativity constraints, while enforcing marginalization to hold between region r and some supersets referred to as its parents $p \in P(r)$. Using this definition, the LP relaxation reads as follows:

$$\max_{b_r}\sum_{r,s_r} b_r(s_r)\theta_r(s_r) \tag{2}$$

$$\text{s.t.}\quad\left\{\begin{array}{ll}\forall r, s_r & b_r(s_r) \geq 0\\ \forall r & \sum_{s_r} b_r(s_r) = 1\\ \forall r, s_r, p \in P(r) & \sum_{s_p\setminus s_r} b_p(s_p) = b_r(s_r).\end{array}\right.$$

This program has the interesting property that whenever its maximizing argument happens to be integral, *i.e.*, the optimal beliefs satisfy $b_r(s_r) \in \{0, 1\}$, the program value equals the MAP value. Moreover, the maximum arguments of the optimal beliefs point toward the MAP assignment (Weiss et al., 2007).

Similar to the parent set we introduce the set of children $c \in C(r) = \{c : r \in P(c)\}$ of region r. Following Sontag & Jaakkola (2009); Werner (2010) we consider the minimization of the following re-parameterization dual

$$q(\lambda) = \sum_r \max_{s_r}\hat{\theta}_r(s_r), \tag{3}$$

with $\hat{\theta}_r(s_r) = \theta_r(s_r) + \sum\limits_{p\in P(r)}\lambda_{r\to p}(s_r) - \sum\limits_{c\in C(r)}\lambda_{c\to r}(s_c)$. The Lagrange multipliers $\lambda_{r\to p}(s_r)$ are introduced for the marginalization constraints, *i.e.*, for $\sum_{s_p\setminus s_r} b_p(s_p) = b_r(s_r)$ which is required to hold $\forall r, s_r, p \in P(r)$.

The dual program value upper bounds the primal program described in Eq. (2). Therefore, to compute the primal optimal value one can minimize the dual upper bound. Using block coordinate descent on the dual objective amounts to optimizing blocks of dual variables while holding the remaining ones fixed. This results in the convex max-product message-passing update rules presented by Hazan & Shashua (2010); Meltzer et al. (2009) and summarized in Fig. 1.

The convex max-product algorithm is guaranteed to converge in value since it minimizes the dual function, which is lower bounded by the primal program. Interestingly, it shares the same complexity as max-product belief propagation, which is attained by replacing the coefficient $1/(1+|P(r)|)$ by 1. It has, however, two fundamental problems. First, it can get stuck in non-optimal stationary points. This happens since the dual objective is non-smooth, thus the algorithm can reach a corner, for which the dual objective stays fixed when changing only a few variables. For example, consider the case of a minimization problem where we try to descend from a pyramid while taking only horizontal and vertical paths. We eventually stay at the same height. The second drawback of convex max-product is that it does not always produce a primal optimal solution $b_r(s_r)$, even when it reaches a dual optimal solution. This happens if the primal solution cannot be consistently reconstructed from the dual variables.

To bypass both of the aforementioned restrictions, Schwing et al. (2012) recently proposed to employ steepest ϵ-descent directions that can be computed by solving the following quadratic program:

$$\min_{b_r} \sum_{r,s_r,p\in P(r)} \left(\sum_{s_p\backslash s_r} b_p(s_p) - b_r(s_r) \right)^2 \tag{4}$$
$$\text{s.t.} \quad \mathcal{C} = \begin{cases} \forall r, s_r \;\; b_r(s_r) \geq 0 \\ \forall r \quad \sum_{s_r} b_r(s_r) = 1 \\ \forall r \quad \sum_{s_r} b_r(s_r)\hat{\theta}_r(s_r) \geq \max\limits_{s_r} \hat{\theta}_r(s_r) - \epsilon. \end{cases}$$

This program is obtained when considering the set of ϵ-subdifferentials for $q(\lambda)$ and searching for the direction of steepest descent. Summing the last constraint over all regions $|\mathcal{R}|$ we observe that the cost function value of 0 ensures $|\mathcal{R}|\epsilon$ optimality while being primal feasible for the original LP relaxation given in Eq. (2), *i.e.*, fulfilling the marginalization constraint.

We note that the constraint set utilized to compute the ϵ-steepest direction nicely decouples. However, solving this quadratic program is computationally challenging due to the cost function which couples the individual beliefs b_r.

3. Steepest Descent Direction with Frank-Wolfe

In what follows, we propose to employ the Frank-Wolfe schema, also known as conditional gradients, to decouple the program given in Eq. (4) and solve it more efficiently. Our intuition is based on the fact that programs with nonlinear cost functions and independent linear constraints can typically be solved via an efficient sequence of much simpler linear programs (Bertsekas et al., 2003). To this end we adopt a Frank-Wolfe schema (Frank & Wolfe, 1956) to compute the steepest ϵ-descent direction, *i.e.*, to solve the quadratic program given in Eq. (4) which is summarized in Fig. 2. Similar to the standard Frank-Wolfe schema it proceeds by iterating three steps. Firstly, a descent direction of the cost function $f(b) = \sum_{r,s_r,p\in P(r)} \left(\sum_{s_p\backslash s_r} b_p(s_p) - b_r(s_r) \right)^2$, linearized at the current iterate b_r is found by solving the following program:

$$u^* = \arg\min_u \sum_{r,s_r} u_r(s_r) \nabla_{b_r(s_r)} f \quad \text{s.t.} \quad u \in \mathcal{C}. \tag{5}$$

Secondly, we compute the optimal step length γ^* before updating the beliefs while making sure not to leave the constraint set, *i.e.*, $0 \leq \gamma^* \leq 1$. The latter two operations, *i.e.*, finding the minimum of a quadratic function in one variable γ and the update step, can be efficiently computed analytically in closed form and involve only simple arithmetic operations.

The conditional gradient method employs the gradient of the cost function. Assuming beliefs b to lie within the constraint set $\mathcal{C}$, the gradient of the cost function $f(b)$ w.r.t. $b_r(s_r)$ is given by

$$\nabla_{b_r(s_r)} f = 2 \left(\sum_{c\in C(r)} d_{c\to r}(s_c) - \sum_{p\in P(r)} d_{r\to p}(s_r) \right),$$

with $d_{r\to p}(s_r) = \sum_{s_p\backslash s_r} b_p(s_p) - b_r(s_r)$ denoting the marginalization disagreements.

The LP given in Eq. (5) decomposes due to the structure within the constraint set $\mathcal{C}$. Hence the optimum is found by considering each region r independently. We therefore solve for all regions r in parallel and refer to the corresponding local constraint set via $\mathcal{C}_r$. The program given in Eq. (5) is hence replaced by small programs, one for every region, which have the following form $\forall r$:

$$u_r^* = \arg\min_{u_r} \sum_{s_r} u_r(s_r) \nabla_{b_r(s_r)} f \tag{6}$$
$$\text{s.t. } u_r \in \mathcal{C}_r = \begin{cases} \forall s_r \;\; u_r(s_r) \geq 0 \\ \quad \sum_{s_r} u_r(s_r) = 1 \\ \quad \sum_{s_r} u_r(s_r)\hat{\theta}_r(s_r) \geq \max_{s_r} \hat{\theta}_r(s_r) - \epsilon. \end{cases}$$

While solving many small decoupled linear programs is arguably faster than a single large program, we are interested in faster algorithms that do not use general LP solvers. To this end, we show existence and availability of a construction for the primal optimal solution of the program given in Eq. (6). The method first identifies the non-zero domain for a feasible solution which is not larger than 2. A primal optimal solution is then analytically computed on this domain by setting a single state to equal one in case the non-zero domain points to a single state. Otherwise we solve a

Algorithm: Conditional Gradient to find the steepest ϵ-descent direction
Iterate until convergence:

1. Solve the LP
$$u^* = \arg\min_u \sum_{r,s_r} u_r(s_r)\nabla_{b_r(s_r)} f \quad \text{s.t.} \quad u \in \mathcal{C}.$$
2. Compute the optimal step size γ^* in closed form by solving
$$\gamma^* = \arg\min_{0\leq\gamma\leq 1} \sum_{r,s_r,p\in P(r)} \left(\sum_{s_p\setminus s_r} \left(b_p(s_p) + \gamma\left(u_p^*(s_p) - b_p(s_p)\right)\right) - \left(b_r(s_r) + \gamma\left(u_r^*(s_r) - b_r(s_r)\right)\right)\right)^2.$$
3. Update the beliefs via
$$b_r(s_r) \leftarrow b_r(s_r) + \gamma^*(u_r^*(s_r) - b_r(s_r)) \quad \forall r, s_r.$$

Figure 2. Frank-Wolfe algorithm for finding the solution of the program given in Eq. (4).

system of linear equations of size 2 × 2 to obtain the two non-zero values.

Theorem 1 *There exists a non-zero domain $\mathcal{S}_r^*$ with $1 \leq |\mathcal{S}_r^*| \leq 2$, and a distribution $u_r(s_r)$ which is primal feasible and primal optimal for the program given in Eq. (6), such that $u_r(s_r) > 0$ only if $s_r \in \mathcal{S}_r^*$.*

Proof: To prove the theorem we consider the Karush-Kuhn-Tucker (KKT) conditions of the program given in Eq. (6). We introduce Lagrange multipliers μ for the constraint encoding that the weighted sum exceeds the constant $\max_{s_r}\hat{\theta}_r(s_r) - \epsilon$, multipliers $\sigma_r(s_r)$ for the positivity constraint and multiplier γ for the summation over states being equal to one. The stationarity, primal feasibility, dual feasibility and complementary slackness requirements are then:

$$\nabla_{b_r(s_r)} f - \mu\hat{\theta}_r(s_r) - \sigma_r(s_r) + \gamma = 0 \;\; \forall s_r \tag{7}$$
$$u_r(s_r) \geq 0 \;\; \forall s_r \tag{8}$$
$$\sum_{s_r} u_r(s_r) = 1 \tag{9}$$
$$\max_{s_r}\hat{\theta}_r(s_r) - \epsilon - \sum_{s_r} u_r(s_r)\hat{\theta}_r(s_r) \leq 0 \tag{10}$$
$$\mu \geq 0 \tag{11}$$
$$\sigma_r(s_r) \geq 0 \;\; \forall s_r \tag{12}$$
$$\mu\left(\max_{s_r}\hat{\theta}_r(s_r) - \epsilon - \sum_{s_r} u_r(s_r)\hat{\theta}_r(s_r)\right) = 0 \tag{13}$$
$$\sigma_r(s_r)u_r(s_r) = 0 \;\; \forall s_r \tag{14}$$

Plugging the stationarity requirement given in Eq. (7) into the complementary slackness constraint provided in Eq. (14) yields $u_r(s_r)\left(\nabla_{b_r(s_r)} f - \mu\hat{\theta}_r(s_r) + \gamma\right) = 0 \, \forall s_r$ with $\sigma_r(s_r) = \nabla_{b_r(s_r)} f - \mu\hat{\theta}_r(s_r) + \gamma \geq 0 \, \forall s_r$ following from Eq. (12).

Assuming $\mu = 0$ requires $\sigma_r(s_r) = \nabla_{b_r(s_r)} f + \gamma \geq 0$. To ensure non-negativity and a minimum cost function value for the program given in Eq. (6), we set γ to be the negative value of the smallest gradient, *i.e.*, $\gamma = -\min_{s_r} \nabla_{b_r(s_r)} f$, and choose one state $\mathcal{S}_r^* \in \arg\max_{\{s_r:\nabla_{b_r(s_r)} f = -\gamma\}} \hat{\theta}_r(s_r)$ to obtain $|\mathcal{S}_r^*| = 1$. Setting $u_r(s_r) = 1$ if $s_r \in \mathcal{S}_r^*$ and to zero otherwise fulfills dual feasibility and complementary slackness constraints. All primal feasibility constraints are fulfilled if $\max_{s_r}\hat{\theta}_r(s_r) - \epsilon - \sum_{s_r} u_r(s_r)\hat{\theta}_r(s_r) \leq 0$ for $s_r \in \mathcal{S}_r^*$.

Thus, we take a single state $\mathcal{S}_r^*$ that has the largest $\hat{\theta}_r(s_r)$ among the minimizing elements of the cost function in Eq. (6). If the distribution $u_r(s_r)$ placing all its mass on that state fulfills the primal feasibility constraint given in Eq. (10) we have found the optimal feasible solution with $|\mathcal{S}_r^*| = 1$.

Let us for now assume existence of a solution for the distribution $u_r(s_r)$ that has at most two non-zero entries. Using the condition obtained earlier by combination of stationarity with complementary slackness, *i.e.*, $u_r(s_r)\left(\nabla_{b_r(s_r)} f - \mu\hat{\theta}_r(s_r) + \gamma\right) = 0$, for the two non-zero states enables computation of μ and γ by solving a linear system analytically. Assuming for now that dual feasibility holds, we construct a primal feasible solution by solving the 2 × 2 linear system arising from Eq. (9) and by enforcing Eq. (10) to hold with equality. We observe primal feasibility, and in particular also Eq. (8) to hold, if the set $\mathcal{S}_r^* = \{s_1, s_2\}$ contains one state s_1 with $\hat{\theta}_r(s_1) > \max_{\hat{s}_r}\hat{\theta}_r(\hat{s}_r) - \epsilon$ and another one with $\hat{\theta}_r(s_2) \leq \max_{\hat{s}_r}\hat{\theta}_r(\hat{s}_r) - \epsilon$.

It remains to be shown that we can find two such states s_1, s_2 which also fulfill dual feasibility. For dual feasibility to hold we require $\mu \geq 0$ and $\sigma_r(s_r) = \nabla_{b_r(s_r)} f - \mu\hat{\theta}_r(s_r) + \gamma \geq 0 \; \forall s_r$. To interpret this program we refer the reader to Fig. 3. Every state s_r defines the linear function $\nabla_{b_r(s_r)} f - \mu\hat{\theta}_r(s_r)$ which depends on the La-

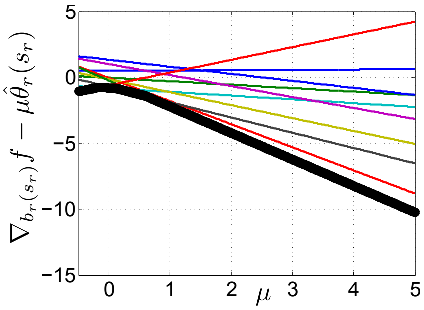


Figure 3. Linear functions for every state s_r with the pointwise minimum given a set of lines being illustrated in black color.

grange multiplier μ. We emphasize that there exists at least one linear function with slope strictly smaller than $-\max_{\hat{s}_r} \hat{\theta}_r(\hat{s}_r) + \epsilon$ due to $\epsilon > 0$. Otherwise the problem would be unbounded. Note the change of sign, *i.e.*, the slope is $-\hat{\theta}_r(s_r)$. Intersecting every line s_1 with slope strictly smaller than $-\max_{\hat{s}_r} \hat{\theta}_r(\hat{s}_r) + \epsilon$ with every line s_2 having a slope larger or equal to this constant gives the point $(\mu', -\gamma')$. Importantly dual feasibility only holds if $\sigma_r(s_r) = \nabla_{b_r(s_r)} f - \mu'\hat{\theta}_r(s_r) + \gamma' \geq 0 \; \forall s_r$, *i.e.*, if $\nabla_{b_r(s_r)} f - \mu'\hat{\theta}_r(s_r) \geq -\gamma' \; \forall s_r$ while $\mu' \geq 0$. Thus every line has to pass above the point of intersection being $(\mu', -\gamma')$. Existence of at least one combination $\mathcal{S}_r^* = \{s_1, s_2\}$ is guaranteed if we found the case $\mu = 0$ not to yield a valid solution.

This concludes the proof of the theorem where we showed that we can construct a primal feasible solution $u_r(s_r)$ with the non-zero domain being at most two. □

Summarizing the constructive proof, we first investigate whether a feasible primal optimal solution can be found for $\mu = 0$, which can be done in linear time. If no such solution is found, we proceed by successively walking along the lower envelope highlighted with black color in Fig. 3. In the following lemma we provide a statement regarding the computational complexity of constructing a primal optimal and primal feasible solution when following the procedure designed when proving Theorem 1.

Lemma 1 *Finding the feasible primal optimal solution $u_r(s_r)$ for the program in Eq.* (6) *has complexity at most $O(|\mathcal{S}_r|^2)$.*

Proof: To prove the lemma we note that intersecting the line defined by $\nabla_{b_r(s_r)} f - \mu\hat{\theta}_r(s_r)$ for every s_r with every other state s_r as described in Theorem 1 is of quadratic complexity. This proves the claim. □

Although the worst complexity of our approach is quadratic in the number of states of a region, the practical performance is often much better as shown by our experimental evaluation. This is also illustrated in Fig. 3, since only 3 out of 10 lines need to be intersected. Note that a better bound of $O(|\mathcal{S}_r| \log |\mathcal{S}_r|)$ on the worst complexity can be obtained using search techniques described by Yu et al. (2010).

```
Algorithm: Efficient Globally Convergent Parallel MAP LP Relaxation Solver
Let $\hat{\theta}_r(s_r) = \theta_r(s_r) + \sum_{p \in P(r)} \lambda_{r\to p}(s_r) - \sum_{c \in C(r)} \lambda_{c\to r}(s_c)$
Iterate until convergence:
  1. For a fixed or variable number of iterations:
     (a) $\forall r$ in parallel: construct a primal feasible solution $u_r(s_r)$
     (b) compute the optimal step size $\gamma^*$ and update the region beliefs $b_r(s_r)$ as detailed in Fig. 2
  2. Compute the disagreement:
         $d_{r\to p}(s_r) = \sum_{s_p \setminus s_r} b_p(s_p) - b_r(s_r)$
  3. Update messages with stepsize $\eta$ obtained through line search to improve $q(\lambda)$:
         $\lambda_{r\to p}(s_r) \leftarrow \lambda_{r\to p}(s_r) + \eta d_{r\to p}(s_r)$
  4. Update potentials
         $\hat{\theta}_r(s_r) \leftarrow \theta_r(s_r) + \sum_{c \in C(r)} \lambda_{c\to r}(s_c) - \sum_{p \in P(r)} \lambda_{r\to p}(s_r)$
```

Figure 4. Our efficient, parallel and provably convergent MAP LP relaxation solver.

It is important to note that it is generally beneficial to interleave the optimization for finding the steepest ϵ-descent direction with the update of the dual variables, *i.e.*, we blend the conditional gradient procedure outlined in Fig. 2 with an update of the dual variables λ. However, there is a trade-off between fewer conditional gradient iterations and more verifications of the resulting ϵ-descent directions. After some conditional gradient iterations we need to verify whether a sufficiently large ($\geq \epsilon$) improvement of the dual $q(\lambda)$ is possible. Hence, a single Frank-Wolfe step might require frequent verifications of the dual improvement.

The obtained globally optimal maximum a-posteriori (MAP) LP relaxation solver is outlined in Fig. 4. We compute the ϵ-steepest descent direction by iterating step 1(a) and step 1(b) for a fixed number of times, verifying a possible ϵ improvement with step 2 and step 3 and continue to iterate after the reparameterization in step 4 if a sufficiently large descent direction was found.

4. Experimental Evaluation

We compare our approach to a wide variety of state-of-the-art baselines using spin-glass models of size 10×10 with variable state-space size and energy functions aris-

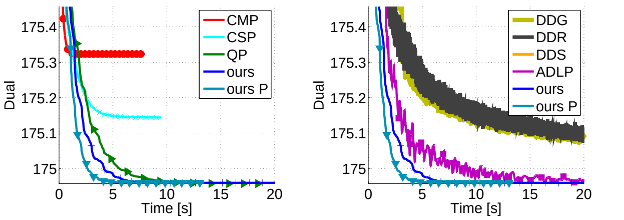


Figure 5. **Dual over time** comparison of our method with monotonically (left) and non-monotonically (right) converging approaches on a 3-state spin-glass model. Baselines and "ours" are single core, "ours P" is parallelized.

ing from a protein design task. As baselines, we employ the alternating direction method for dual MAP LP relaxations (ADLP) (Meshi & Globerson, 2011), the quadratic programming (QP) formulation given by Schwing et al. (2012), convex max-product (CMP) and convex sum-product (CSP) (Hazan & Shashua, 2010) as well as the dual-decomposition work of Komodakis et al. (2010) provided in a generic (DDG), a re-weighted (DDR) and a smoothed (DDS) version in the STAIR library by Gould et al. (2011). Note that ADLP is also implemented in this library. All algorithms are restricted to at most $5,000$ iterations and all baselines utilize a single core.

Convex max-product and ϵ-steepest descent optimize the same cost function. Following Schwing et al. (2012), we start with efficient block-coordinate descent steps before switching to computing the ϵ-steepest descent directions via the conditional gradient procedure. We start from $\epsilon = 0.01$ and successively decrease its value if the model is sufficiently close to $|\mathcal{R}|\epsilon$ optimality, *i.e.*, if ϵ is larger than $f(b)/1000$. We denote the single-core approach as "ours" and refer to the parallel implementation employing 16 cores as "ours P."

4.1. Spin Glasses

We first consider spin glass models that consist of local factors, each having 3 states with values randomly chosen according to a zero mean, unit variance normal distribution $\mathcal{N}(0,1)$. The pairwise factors of the regular grid are weighted potentials with $+1$ on the diagonal, and off-diagonal entries being -1. The weights are independently drawn from $\mathcal{N}(0,1)$. As shown by Schwing et al. (2012), convex max-product does not achieve optimality for about 20% of the models in this setting, illustrating the need for globally convergent algorithms.

Dual over time: Fig. 5 shows the convergence behavior for a spin glass model and illustrates the dual value obtained after a certain amount of time measured in seconds. As shown in the figure, CMP is monotonically decreasing but gets stuck in a corner. Convex sum product (CSD) is guaranteed to monotonically converge to the optimum of a problem modified by entropy terms which could be far from the MAP solution even if the corresponding temperature is as low as 0.001. It is important to note that our ϵ-descent approach is monotonically decreasing just like the QP formulation, which contrasts the other baselines (ADLP, DDG, DDR, DDS). Our single core approach using 400 Frank-Wolfe iterations is slightly faster than the QP method but significantly easier to parallelize efficiently.

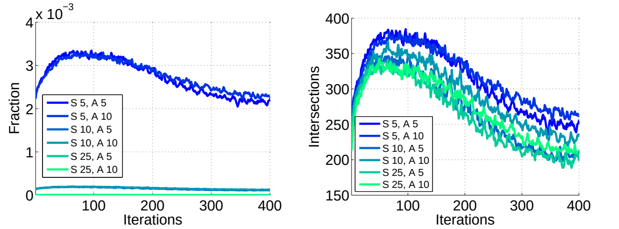


Figure 6. **Performance scaling:** Impact of state-space size (S) and interaction strength (A) on the fraction (left) and absolute number of line intersections (right).

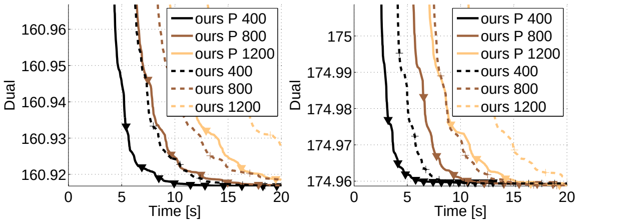
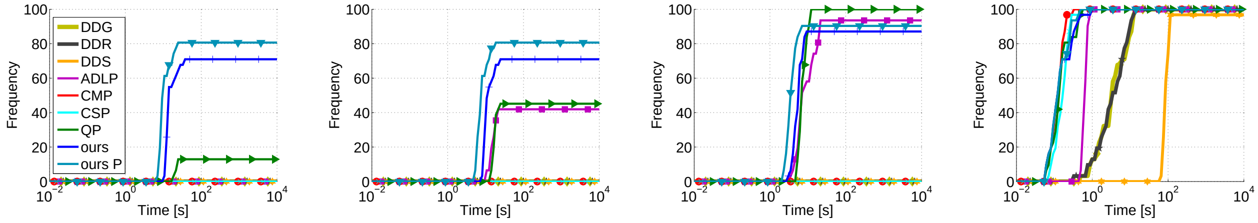


Figure 7. **Frank-Wolfe iterations:** 400 conditional gradient iterations result in fast convergence for the spin glass given on the left, 800 are about as suitable for the case visualized on the right.

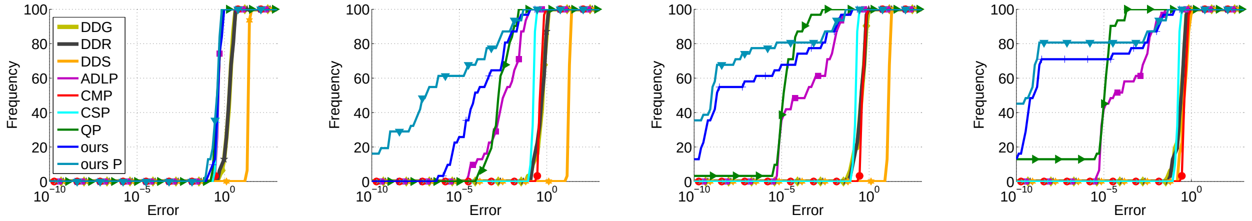
Performance scaling: We next investigate how our approach scales with the state-space size $|\mathcal{S}_r|$. We therefore measure the fraction of line intersections (Fig. 6 left) and the actual number of intersections (Fig. 6 right) as a function of the number of iterations for "hard" spin-glass models of size 10×10 with state space sizes 5, 10 and 25. We defined **"hard" spin glass models** as those where the difference between convex max-product and the ϵ-descent method is larger than 0.2. As shown in Fig. 6, where results are averaged over 30 models, the fraction of actually intersected lines compared to the maximally possible number (worst case complexity in Lemma 1) is very small. Further, there is no negative influence when increasing the strengths of the pairwise interaction. Thus the increase of wall-clock time for larger sized models is not due to the suggested line-intersection method but rather due to additional operations required for computation of the dual, gradient, *etc*.

Frank-Wolfe iterations: We evaluate how many conditional gradient iterations are required before checking whether we can find an ϵ-descent direction. To this end, we compare our approach using 400, 800 and 1200 Frank-Wolfe iterations. The results for two spin glass models having a state-space size of 3 are given in Fig. 7. In the first case 400 conditional gradient iterations are sufficient. Note that choosing the number of iterations too low might result in slow convergence due to frequent ϵ-descent checks that are not successful, as illustrated on the right hand side of the figure. In the following we set the number of it-



(a) within 10^{-7} (b) within 10^{-5} (c) within 10^{-2} (d) within 10^{0}

Figure 8. **Time for accuracy:** Time it requires to get a fraction of the samples to within the indicated optimality.



(a) after 1s (b) after 10s (c) after 20s (d) after 200s

Figure 9. **Error after time:** Percentage of samples that achieved a smaller error after the indicated time.

erations to 400 and note that an adaptive approach which successively increases the number of iterations might be preferred, as the ϵ-descent directions are more easily found during the first few rounds.

Time for a given accuracy: The percentage of tasks that reaches a given accuracy after a given time, averaged over 30 "hard" spin glass models, is shown in Fig. 8. We observe that significantly more samples achieve a lower deviation from the optimum in a smaller amount of time when using the proposed approach.

Error for a given time: We evaluate the fraction of problems that achieved a deviation from the optimum smaller than a certain error given a specific amount of time. As shown in Fig. 9 our approach outperforms the baselines most of the time.

4.2. Protein Design

Next we consider a protein design task and compare our proposed conditional gradient method to the performance of the most competitive algorithms, *i.e.*, ADLP, QP and CMP. To this end, we make use of the eight problems from the probabilistic inference challenge[1]. For all algorithms we use the same setting as before, except that we increase the number of the maximally possible iterations to $50,000$ and the Frank-Wolfe iterations to $2,000$ to increase the chances of finding a descent direction and hence decrease the number of evaluations of the dual.

Dual over time: Fig. 10 illustrates the dual energy over time. We observe fast convergence when the convex max-product (CMP) algorithm is optimal (Fig. 10 left). Similar to spin glass models, CMP gets stuck in corners as shown in all but the leftmost plot of Fig. 10. Our approach successfully finds the global optimum while monotonically decreasing the dual energy.

Cumulative frequency w.r.t. time/error: We evaluate the time required to achieve a specific accuracy as well as the cumulative error distribution after a given amount of time in Fig. 11 and Fig. 12 respectively. We observe that our approach achieves very good performance.

5. Related Work

Efficient dual solvers were extensively studied in the context of LP relaxations for the MAP problem (Koster et al., 1998; Schlesinger, 1976; Wainwright et al., 2005). Dual block coordinate descent methods, typically referred to as convex max-product algorithms, are monotonically decreasing, and were shown to be very efficient (Globerson & Jaakkola, 2007; Hazan & Shashua, 2010; Kolmogorov, 2006; Meltzer et al., 2009; Sontag & Jaakkola, 2009; Tarlow et al., 2011; Werner, 2010). Since the dual program is non-smooth, these algorithms can however get stuck in non-optimal stationary points and cannot in general recover a primal optimal solution (Weiss et al., 2007).

To fix the convergence issue, two main directions have been pursued, *i.e.*, smoothing or directly optimizing the non-smooth dual. To smooth the dual objective, some methods use the soft-max with low or decreasing temperature in order to avoid corners as well as to recover primal optimal solutions (Hazan & Shashua, 2010; Johnson, 2008; Jojic et al., 2010; Savchynskyy et al., 2012). However, these methods are generally slower, as computation of the exponential and logarithm functions is more expensive than finding the maximum. Ravikumar et al. (2010) applied the proximal method, leveraging a primal strictly concave modification, which results in a smooth dual approximation. This approach converges to the dual optimum and recovers the primal optimal solution. However, it uses a

[1] http://www.cs.huji.ac.il/project/PASCAL/index.php

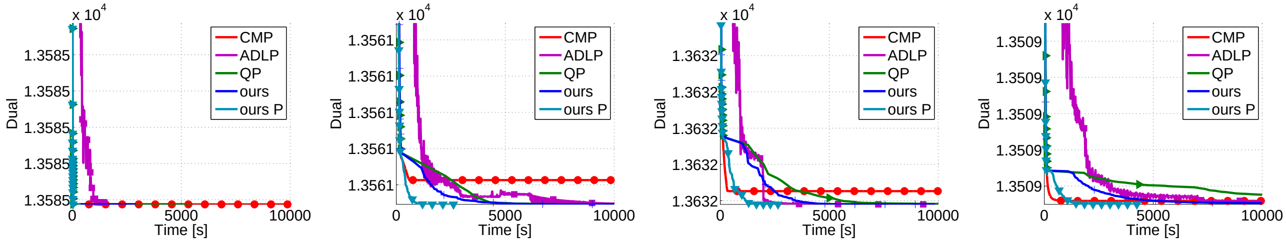
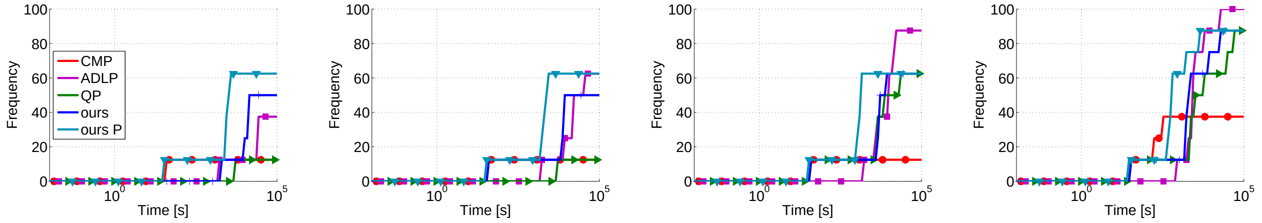
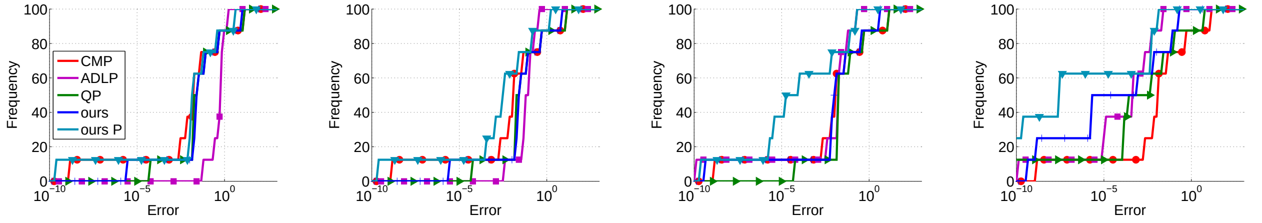


Figure 10. **Dual over time** for four different protein design tasks.



(a) within 10^{-7} (b) within 10^{-5} (c) within 10^{-3} (d) within 10^{-2}

Figure 11. **Cumulative frequency w.r.t. time:** Time it requires to get a fraction of the samples to within the indicated optimality.



(a) after 500s (b) after 1000s (c) after 2000s (d) after 10000s

Figure 12. **Cumulative frequency w.r.t. error:** Percentage of samples that achieved a specific error after the indicated time.

double loop scheme where every update involves executing a convex sum-product algorithm. Other methods apply augmented Lagrangian techniques to the primal (Martins et al., 2011) and the dual programs (Meshi & Globerson, 2011). This guarantees to reach the global optimum and recovers the dual and primal solutions. It is however not monotonically decreasing and thus it cannot be efficiently integrated with convex max-product updates that perform block coordinate descent on the dual of the LP relaxation.

When directly optimizing the original non-smooth dual, subgradient descent algorithms are guaranteed to reach the dual optimum, as well as recover the primal optimum (Komodakis et al., 2010). A bundle approach presented by Lemaréchal (1974); Kappes et al. (2012) also employs subgradients. Despite their theoretical guarantees, methods employing subgradients are slow because arbitrary gradient directions are chosen. This contrasts the approach presented by Schwing et al. (2012) where the steepest, *i.e.*, fastest, descent direction from the set of ϵ-subdifferentials is employed.

Following Schwing et al. (2012), our approach is based on the ϵ-descent algorithm for convex functions (Bertsekas et al., 2003). Similarly, we use the ϵ-margin of the Fenchel-Young duality theorem to obtain the ϵ-subdifferential of the dual objective of the LP relaxation, thus augmenting the convex max-product algorithm with the ability to get out of corners and to recover a primal optimal solution. Finding the steepest descent direction within the set of ϵ-subdifferentials requires solving a quadratic program. Contrasting the work of Schwing et al. (2012) we replace this task by a series of linear programs following the Frank-Wolfe scheme inspired from (Lacoste-Julien et al., 2013; Bach, 2013). Due to the constraint structure, we show that the problem decouples and is hence trivially parallelizable. Further we show how to construct a simple iterative procedure to efficiently solve the small sub-problems.

6. Conclusion

We presented an algorithm to compute the globally optimal solution of the LP relaxation of the MAP problem. To this end, we proposed to employ the conditional gradient (Frank-Wolfe) algorithm to replace the standard quadratic programming solver. This results in an algorithm which is easily parallelized and where each subproblem is efficiently solvable via a search procedure. We showed that our approach outperforms existing solvers on synthetic spin-glass models and on protein design tasks. In the future, we plan to investigate theoretical convergence rates of steepest ϵ-descent algorithms.

Acknowledgements: We gratefully acknowledge support from ONR Grant N00014-14-1-0232.

References

Bach, F. Duality between subgradient and conditional gradient methods. Technical report, INRIA - Sierra project team, 2013.

Bertsekas, D. P., Nedić, A., and Ozdaglar, A. E. *Convex Analysis and Optimization*. Athena Scientific, 2003.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956.

Globerson, A. and Jaakkola, T. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Proc. NIPS*, 2007.

Gould, S., Russakovsky, O., Goodfellow, I., Baumstarck, P., Ng, A. Y., and Koller, D. The STAIR Vision Library (v2.4), 2011. http://ai.stanford.edu/ sgould/svl.

Hazan, T. and Shashua, A. Norm-Product Belief Propagation: Primal-Dual Message-Passing for LP-Relaxation and Approximate-Inference. *Trans. Information Theory*, 2010.

Johnson, J. K. *Convex relaxation methods for graphical models: Lagrangian and maximum entropy approaches*. PhD thesis, MIT, 2008.

Jojic, V., Gould, S., and Koller, D. Accelerated dual decomposition for MAP inference. In *Proc. ICML*, 2010.

Kappes, J. H., Savchynskyy, B., and Schnörr, C. A Bundle Approach To Efficient MAP-Inference by Lagrangian Relaxation. In *Proc. CVPR*, 2012.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 2006.

Komodakis, N., Paragios, N., and Tziritas, G. MRF Energy Minimization & Beyond via Dual Decomposition. *PAMI*, 2010.

Koster, A., van Hoesel, C. P. M., and Kolen, A. W. J. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 1998.

Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *Proc. ICML*, 2013.

Lemaréchal, C. An algorithm for minimizing convex functions. *Information processing*, 1974.

Martins, A. F. T., Figueiredo, M. A. T., Aguiar, P. M. Q., Smith, N. A., and Xing, E. P. An Augmented Lagrangian Approach to Constrained MAP Inference. In *Proc. ICML*, 2011.

Meltzer, T., Globerson, A., and Weiss, Y. Convergent Message Passing Algorithms: a unifying view. In *Proc. UAI*, 2009.

Meshi, O. and Globerson, A. An Alternating Direction Method for Dual MAP LP Relaxation. In *Proc. ECML PKDD*, 2011.

Ravikumar, P., Agarwal, A., and Wainwright, M. J. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *JMLR*, 2010.

Savchynskyy, B., Schmidt, S., Kappes, J. H., and Schnörr, C. Efficient MRF Energy Minimization via Adaptive Diminishing Smoothing. In *Proc. UAI*, 2012.

Schlesinger, M. I. Sintaksicheskiy analiz dvumernykh zritelnikh signalov v usloviyakh pomekh (Syntactic analysis of two-dimensional visual signals in noisy conditions). *Kibernetika*, 1976.

Schwing, A. G., Hazan, T., Pollefeys, M., and Urtasun, R. Globally Convergent Dual MAP LP Relaxation Solvers using Fenchel-Young Margins. In *Proc. NIPS*, 2012.

Sontag, D. and Jaakkola, T. Tree Block Corrdinate Descent for MAP in Graphical Models. In *Proc. AISTATS*, 2009.

Tarlow, D., Batra, D., Kohli, P., and Kolmogorov, V. Dynamic Tree Block Coordinate Ascent. In *Proc. ICML*, 2011.

Wainwright, M. J., Jaakkola, T., and Willsky, A. S. MAP estimation via agreement on trees: message-passing and linear programming. *Trans. Information Theory*, 2005.

Weiss, Y., Yanover, C., and Meltzer, T. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Proc. UAI*, 2007.

Werner, T. Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction. *PAMI*, 2010.

Yu, J., Vishwanathan, S. V. N., Günter, S., and Schraudolph, N. N. A Quasi-Newton Approach to Nonsmooth Convex Optimization Problems in Machine Learning. *JMLR*, 2010.