Accelerated Inference in Markov Random Fields via Smooth Riemannian Optimization

Siyi Hu and Luca Carlone

Abstract—Markov Random Fields (MRFs) are a popular model for several pattern recognition and reconstruction problems in robotics and computer vision. Inference in MRFs is intractable in general and related work resorts to approximation algorithms. Among those techniques, semidefinite programming (SDP) relaxations have been shown to provide accurate estimates while scaling poorly with the problem size and being typically slow for practical applications. Our first contribution is to design a dual ascent method to solve standard SDP relaxations that takes advantage of the geometric structure of the problem to speed up computation. This technique, named Dual Ascent Riemannian Staircase (DARS), is able to solve large problem instances in seconds. Since our goal is to enable real-time inference on robotics platforms, we develop a second and faster approach. The backbone of this second approach is a novel SDP relaxation combined with a fast and scalable solver based on smooth Riemannian optimization. We show that this approach, named Fast Unconstrained SEMidfinite Solver (FUSES), can solve large problems in milliseconds. Contrarily to local MRF solvers, e.g., loopy belief propagation, our approaches do not require an initial guess. Moreover, we leverage recent results from optimization theory to provide per-instance sub-optimality guarantees. We demonstrate the proposed approaches in multi-class image segmentation problems. Extensive experimental evidence shows that (i) FUSES and DARS produce near-optimal solutions, attaining an objective within 0.2% of the optimum, (ii) FUSES and DARS are remarkably faster than general-purpose SDP solvers, and FUSES is more than two orders of magnitude faster than DARS while attaining similar solution quality, (iii) FUSES is faster than local search methods while being a global solver, and is a good candidate for real-time robotics applications.

Index Terms—Object Detection, Segmentation and Categorization; Optimization and Optimal Control; Recognition.

I. INTRODUCTION

Markov Random Fields (MRFs) are a popular graphical model for reconstruction and recognition problems in computer vision and robotics, including 2D and 3D semantic segmentation, stereo reconstruction, image restoration and denoising, texture synthesis, object detection, and panorama stitching [1], [2], [3]. An MRF can be understood as a factor graph including only unary and binary factors, and where node variables are discrete labels. The discrete nature of the variables makes maximum a posteriori (MAP) inference in MRFs intractable in general, hence several MRF-based applications remain out of reach for real-time robotics. Our motivating application in this paper is real-time semantic segmentation, which is crucial for the robot to understand the surrounding environment and execute high-level tasks.

This work was partially funded by MIT Lincoln Laboratory under the grant “Fast Semantic Segmentation on Manifold”. Therefore, we are interested in developing real-time MRF solvers that can support online operation at scale.

The literature on MRFs (reviewed in Section IV) is vast and includes methods based on graph cuts, message passing techniques, greedy methods, and convex relaxations, to mention a few. These approaches are typically approximation techniques, in the sense that they attempt to compute near-optimal MAP estimates efficiently (the problem is NP-hard in general, hence we do not expect to compute exact solutions in polynomial time). Among those, semidefinite programming (SDP) relaxations have been recognized to produce accurate approximations [4]. On the other hand, the computational cost of general-purpose SDP solvers prevented widespread use of this technique beyond problems with few hundred variables [5] (semantic segmentation typically involves thousands to millions of variables), and SDPs lost popularity in favor of computationally cheaper alternatives including move-making algorithms (based on graph cut) and message passing. Move-making methods [6] require specific assumptions on the MRF and their performance typically degrades when these assumptions are not satisfied. Message passing methods [7], [8], on the other hand, may not even converge, even though they are observed to work well in practice.

Contribution. Our first contribution, presented in Section III is to design a dual-ascent-based method to solve standard SDP relaxations that takes advantage of the geometric structure of the problem to speed up computation. In particular, we show that each dual ascent iteration can be solved using a fast low-rank SDP solver known as the Riemannian Staircase [9]. Upon convergence of the dual ascent iterations this technique attains the same objective as the standard SDP relaxation while being more scalable. This technique, named Dual Ascent Riemannian Staircase (DARS), is able to solve MRF instances with thousands of variables in seconds, while general-purpose SDP solvers (e.g., cvx [10]) are not able to provide an answer in reasonable time (hours) at that scale.

Our second contribution, presented in Section IV is an even faster SDP relaxation. Despite being remarkably faster than general-purpose SDP solvers, DARS is currently slow for real-world robotics applications, hence, we develop a...
Fast Unconstrained SEMidefinite Solver (FUSES) that can solve large problems in milliseconds. The backbone of this second approach is a novel SDP relaxation combined with the Riemannian Staircase method [2]. The novel formulation uses a more intuitive binary matrix (with entries in \( \{0, 1\} \)), contrarily to related work that parametrizes the problem using a vector with entries in \( \{-1, +1\} \).

Our third contribution is an extensive experimental evaluation. We test the proposed SDP solvers in semantic image segmentation problems and evaluate the corresponding results in terms of accuracy and runtime. We compare the proposed techniques against several related approaches, including move-making methods (\( \alpha \)-expansion [6]) and message passing (Loopy Belief Propagation [8] and Tree-Reweighted Message Passing [7]). The results show that our MRF solver retains all the advantages of SDP relaxations (accuracy, no need for initial guess, no assumption on the objective function), while being fast and scalable. More specifically, our results show that (i) FUSES and DARS produce near-optimal solutions, attaining an objective within 0.2% of the optimum, (ii) FUSES and DARS are remarkably faster than general-purpose SDP solvers, and FUSES is more than two orders of magnitude faster than DARS while retaining similar solution quality, (iii) FUSES is faster than local search methods while being a global solver.

Before delving into the contribution of this paper, Section II provides preliminary notions on inference in MRFs, while we postpone the review of related work to Section VII. All proofs are given in the supplemental material [11].

II. PRELIMINARIES

This section introduces standard notation for MRFs (Section II-A) and provides necessary background on semidefinite relaxations (Section II-B).

A. Markov Random Fields: Models and Inference

A Markov Random Field (MRF) is a graphical model in which nodes are associated with discrete labels we want to estimate, and edges (or potentials) represent given probabilistic constraints relating the labels of a subset of nodes. Formally, for each node \( i \) in the node set \( V = \{1, \ldots, N\} \) (where \( N \) is the number of nodes), we need to assign a label \( x_i \in \mathcal{X} \), where \( \mathcal{X} = \{1, \ldots, K\} \) is the set of \( K \) possible labels. If \( K = 2 \) (i.e., nodes are classified into two classes) the corresponding model is called a binary MRF. Here we consider \( K \geq 2 \) possible labels, a setup generally referred to as a multi-label MRF.

Maximum a posteriori (MAP) inference. The MAP estimate is the most likely assignment of labels, i.e., the assignment of the node labels that attains the maximum of the posterior distribution of an MRF, or, equivalently, the minimum of the negative log-posterior. MAP estimation can be formulated as a discrete optimization problem over the labels \( x_i \in \mathcal{X} \) with \( i = 1, \ldots, N \):

\[
\min_{x_i \in \mathcal{X}} \sum_{i \in \mathcal{U}} E_i(x_i) + \sum_{(i,j) \in \mathcal{B}} E_{ij}(x_i, x_j) \quad (P0)
\]

where \( \mathcal{U} \subseteq V \) is the set of unary potentials (probabilistic constraints involving a single node), \( \mathcal{B} \subseteq V \times V \) is the set of binary potentials (involving a pair of nodes), and \( E_i(\cdot) \) and \( E_{ij}(\cdot, \cdot) \) represent the negative log-distribution for each unary and binary potential, respectively (described below). For instance, in semantic segmentation each node in the MRF corresponds to a pixel (or superpixel) in the image, the unary potentials are dictated by pixel-wise classification from a classifier applied to the image, and the binary potentials enforce smoothness of the resulting segmentation [12]. The binary potentials (often referred to as smoothness priors) are typically enforced between nearby (adjacent) pixels.

MRF Potentials. A typical form for the unary and binary potentials is:

\[
E_i(x_i) = \begin{cases} 0 & \text{if } x_i = \bar{x}_i \\ \delta_i & \text{otherwise} \end{cases}
\]

\[
E_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ \delta_{ij} & \text{otherwise} \end{cases}
\]

\[(1)\]

where \( \bar{x}_i \) is a data-driven noisy measurement of the label of node \( i \) (typically from a classifier), and \( \delta_i \) and \( \delta_{ij} \) are given scalars. Typically, it is assumed \( \delta_i \geq 0 \), i.e., choosing a label different from the measured one incurs a cost \( \delta_i \) in \( P0 \). Similarly, for the binary potentials \( E_{ij}(\cdot) \) it is typically assumed \( \delta_{ij} \geq 0 \), i.e., label mismatch (\( x_i \neq x_j \)) incurs a cost of \( \delta_{ij} \) in the objective \( P0 \). In this case the binary potentials are called attractive, while they are referred to as repulsive when \( \delta_{ij} < 0 \) (i.e., the potentials encourage label mismatches) [13].

The MRF resulting from the choice of potentials in eq. (1) is known as the Potts model [14], which was first proposed in statistical mechanics to model interacting spins in ferromagnetic materials. When \( K = 2 \) (binary MRFs) the resulting model is known as the Ising model [2] Section I.4.1.

B. Standard Semidefinite Relaxation

Semidefinite programming (SDP) relaxation has been shown to provide an effective approach to compute a good approximation of the global minimizer of \( P0 \) [8], [15], [16]. In this section we introduce a standard approach to obtain an SDP relaxation, for which we design a fast solver in Section III.

In order to obtain an SDP relaxation, related works rewrite each node variable \( x_i \in \mathcal{X} \) as a vector \( w_i \in \{-1, +1\}^K \), such that \( w_i \) has a single entry equal to +1 (all the others are −1), and if the \( j \)-th entry of \( w_i \) is +1, then the corresponding node has label \( j \). Moreover, they stack all vectors \( w_i \), \( i = 1, \ldots, N \), in a single \( NK \)-vector \( w = [w_1^T \ w_2^T \ \ldots \ \ w_N^T]^T \). Using this reparametrization, the inference problem \( P0 \) can be written in terms of the vector \( w \) as follows (full derivation in Appendix A):

\[
\min_w \quad \mathbf{u}^\top A \mathbf{w} + 2b^\top \mathbf{w}
\]

subject to \( \text{diag} (ww^\top) = 1_{NK} \),

\[
\mathbf{u}_i^\top \mathbf{w} = 2 - K, \quad i = 1, \ldots, N
\]

where \( A \) and \( b \) are a suitable symmetric matrix and a suitable vector collecting the coefficients of the binary terms and the unary terms in \( P0 \), respectively; \( \text{diag} (ww^\top) \) is the diagonal of the matrix \( ww^\top \), and \( \mathbf{u}_i = e_i^\top \otimes 1_K \), where \( e_i \) is an \( N \)-vector which is all zero, except the \( i \)-th entry which is one, \( 1_K \) is a \( K \)-vector of ones, and \( \otimes \) is the Kronecker product.

Intuitively, \( \text{diag} (ww^\top) \) contains the square of each entry of
we, hence $\text{diag}(ww^T) = 1_N$ imposes that every entry of $w$ has norm 1, i.e., it belongs to $\{-1,+1\}$; the constraint $u_i^Tw = 2 - K$ writes in compact form $1^T w_i = 2 - K$, which enforces each node to have a unique label (i.e., a single entry in $w_i$ can be +1, while all the others are −1).

Before relaxing problem (2), it is convenient to homogenize the objective by reparametrizing the problem in terms of an extended vector $y = [u^T; v^T]^T$, where an entry equal to 1 is concatenated to $w$. We can now rewrite (2) in terms of $y$:

$$\begin{array}{l}
    f_{P1} = \min_y \\    \text{subject to} \quad \text{tr} (Lyy^T) + \text{diag}(yy^T) = 1_{NK+1} \\
    \quad \text{tr} (U_iyy^T) = 2 - K, \quad i = 1, \ldots, N
\end{array}$$

where $L = \begin{bmatrix} A & b \\ b^T & 0 \end{bmatrix}$ and $U_i = \begin{bmatrix} 0 & 0 \\ w_i^T & w_i \end{bmatrix}$. In (P1), we used the equality $yy^T = \text{tr} (Lyy^T)$, and noted that since

$$yy^T = \begin{bmatrix} w^T \\ 0 \\ \vdots \\ w^T \end{bmatrix},$$

then $\text{tr} (U_iyy^T) = u_i^T w$. So far we have only reparameterized problem (P0), hence (P1) is still a MAP estimator. We can now introduce the SDP relaxation: problem (P1) only includes terms in the form $yy^T$, hence we can reparametrize it using a matrix $Y = yy^T$. Moreover, we note that the set of matrices $Y$ that satisfy $Y = yy^T$ is the set of positive semidefinite ($Y \succeq 0$) rank-1 matrices ($\text{rank}(Y) = 1$). Rewriting (P1) using $Y$ and dropping the non-convex rank-1 constraint, we obtain:

$$\begin{array}{l}
    f_{S1} = \min_Y \\    \text{subject to} \quad \text{tr} (LY) + \text{diag}(Y) = 1_{NK+1} \\
    \quad \text{tr} (U_iY) = 2 - K, \quad i = 1, \ldots, N
\end{array}$$

which is a (convex) semidefinite program and can be solved globally in polynomial time using interior-point methods. While the SDP relaxation (S1) is known to provide near-optimal approximations of the MAP estimate, interior-point methods are typically slow in practice and cannot solve problems with more than few hundred nodes in a reasonable time.

III. DARS: DUAL ASCENT Riemannian Staircase

This section presents the first contribution of this paper: a dual ascent approach to efficiently solve large instances of the standard SDP relaxation (S1).

A. Dual Ascent Approach

The main goal of this section is to design a dual ascent method, where the subproblem to be solved at each iteration has a more favorable geometry, and can be solved quickly using the Riemannian Staircase method introduced in Section III-B. Towards this goal, we rewrite (S1) equivalently as:

$$\begin{array}{l}
    \min_Y g(Y) \\    \text{subject to} \quad \text{tr} (U_iY) = 2 - K, \quad i = 1, \ldots, N
\end{array}$$

where the objective function is now $g(Y) = \text{tr} (LY) + \mathcal{I}(\text{diag}(Y) = 1_{NK+1}) + \mathcal{I}(Y \succeq 0)$, where $\mathcal{I}(\cdot)$ is the indicator function which is zero when the constraint inside the parenthesis is satisfied and plus infinity otherwise.

Under constraints qualification (e.g., the Slater’s condition for convex programs), we can obtain an optimal solution to (3) by computing a saddle-point of the Lagrangian function $\mathcal{L}(Y, \lambda)$:

$$\max_{\lambda} \min_{Y} \mathcal{L}(Y, \lambda) = \max_{\lambda} g(Y) + \sum_{i=1}^{N} \lambda_i (\text{tr} (U_i Y) + K - 2)$$

where $\lambda \in \mathbb{R}^N$ is the vector of dual variables and $Y$ is the primal variable.

The basic idea behind dual ascent [19 Section 2.1] is to solve the saddle-point problem by alternating maximization steps with respect to the dual variables $\lambda$ and minimization steps with respect to the primal variable $Y$.

**Dual Maximization.** The maximization of the dual variable $\lambda$ is carried out via gradient ascent. In particular, at each iteration $t = 1, \ldots, T$ ($T$ is the maximum number of iterations), the dual ascent method fixes the primal variable and updates the dual variable $\lambda$ as:

$$\lambda^{(t)} = \lambda^{(t-1)} + \alpha \nabla \mathcal{L}_Y(Y^{(t-1)}, \lambda^{(t-1)})$$

where $\nabla \mathcal{L}(Y^{(t-1)}, \lambda^{(t-1)})$ is the gradient of the Lagrangian with respect to the dual variables, evaluated at the latest estimate of the primal-dual variables $(Y^{(t-1)}, \lambda^{(t-1)})$, and $\alpha$ is a suitable stepsize. It is straightforward to compute the gradient with respect to the $i$-th dual variable $\lambda_i$ as $\nabla \mathcal{L}_{Y}(Y, \lambda) = \text{tr} (U_i Y) + K - 2$. Intuitively, the second summand in (4) penalizes the violation of the constraint $\text{tr} (U_i Y) = 2 - K$ (for all $i$). Moreover, since the gradient in (5) grows with the amount of violation $\text{tr} (U_i Y) - K + 2$, the dual update increases the penalty for constraints with large violation.

**Primal Minimization.** The minimization step fixes the dual variable to the latest estimate $\lambda^{(t-1)}$ and minimizes (4) with respect to the primal variable $Y$:

$$\min_Y g(Y) + \sum_{i=1}^{N} \lambda_i^{(t-1)} (\text{tr} (U_i Y) + K - 2)$$

where we substituted “inf” for “min” since the objective cannot drift to minus infinity due to the implicit constraints imposed by the indicator functions in $g(Y)$. Recalling the expression of $g(Y)$, defining $L_{\lambda} = L + \sum_{i=1}^{N} \lambda_i^{(t-1)} U_i$, and moving again the indicator functions to the constraints we write (6) more explicitly as:

$$Y^{(t)} = \arg\min_Y \mathcal{L}(Y) \quad \text{subject to} \quad \text{tr} (L_{\lambda} Y) + \mathcal{I}(\text{diag}(Y) = 1_{NK+1}) + \mathcal{I}(Y \succeq 0)$$

where we dropped the constant terms $\sum_{i=1}^{N} \lambda_i^{(t-1)} (K - 2)$ from the objective since they are irrelevant for the optimization. The minimization step in the dual ascent is again an SDP, but contrarily to the standard SDP (S1), problem (7) can be solved quickly using the Riemannian Staircase, as discussed in the following.

B. A Riemannian Staircase for the Dual Ascent Iterations

This section provides a fast solver to compute a solution for the SDP (7), that needs to be solved at each iteration of the dual ascent method of Section III-A.

We use of the Burer-Monteiro method, which replaces the matrix $Y \in \mathbb{R}^{(NK+1) \times (NK+1)}$ in (7) with a rank-$r$ product $RR^T$ with $R \in \mathbb{R}^{(NK+1) \times r}$:

$$\begin{array}{l}
    \min_R \\    \text{subject to} \quad \text{tr} (L_{\lambda} RR^T) + \mathcal{I}(\text{diag}(RR^T) = 1_{NK+1})
\end{array}$$
Note that the constraint $Y \succeq 0$ in (7) becomes redundant after the substitution, since $RR^T$ is always positive semidefinite, hence it is dropped.

Following Boumal et al. [9] we note that the constraint set in (8) describes a smooth manifold, and in particular a product of Stiefel manifolds. To make this apparent, we recall that the (transposed) Stiefel manifold is defined as [9]:

$$\text{St}(d, p) = \{M \in \mathbb{R}^{d \times p} : MM^T = I_d\}$$  \hspace{1cm} (9)

Then, we observe that $\text{diag}(RR^T) = 1_{N^2K+1}$ can be written as $R_iR_i^T = 1, i = 1, \ldots, NK + 1$ (where $R_i$ is the $i$-th row of $R$), which is equivalent to saying that $R_i \in \text{St}(1, r)$ for $i = 1, \ldots, NK + 1$. This observation allows concluding that the matrix $R$ belongs to the product manifold $\text{St}(1, r)^{NK+1}$.

Therefore, we can rewrite (8) as an unconstrained optimization on manifold:

$$\min_{R \in \text{St}(1, r)^{NK+1}} \text{tr}(L_iR R^T)$$  \hspace{1cm} (R1)

The formulation (R1) is non-convex (the product of Stiefel manifolds describes a non-convex set), but one can find local minima efficiently using iterative methods [9], [21]. While it might seem that little was gained (we started with an intractable problem and we ended up with another non-convex problem), the following remarkable result from Boumal et al. [9] ties back the local solutions of (R1) to globally optimal solutions of the SDP (7).

**Proposition 1 (Optimality Conditions for (R1), Corollary 8 in [9]):** If $R \in \text{St}(1, r)^{NK+1}$ is a (column) rank-deficient second-order critical point of problem (R1), then $R$ is a global optimizer of (R1), and $Y^* = RR^T$ is a solution of the semidefinite relaxation (7).

The previous proposition ensures that when local solutions (second-order critical points) of (R1) are rank deficient, then they can be mapped back to global solutions of (7), hence providing a way to solve (7) efficiently via (R1).

The catch is that one has to choose the rank $r$ large enough to obtain rank-deficient solutions. Related work [9] therefore proposes the Riemannian staircase method, where one solves (R1) for increasing values of $r$ till a rank-deficient solution is found. Boumal et al. [9] also provide theoretical results ensuring that rank-deficient solutions are found for small $r$ (more details in Section V).

**C. DARS: Summary, Convergence, and Guarantees**

We name DARS (Dual Ascent Riemannian Staircase) the approach resulting from the combination of dual ascent and the Riemannian Staircase. DARS starts with an initial guess for the dual variables (we use $\lambda = 0_N$), and then alternates two steps: (i) the primal minimization where a solution for (7) is obtained using the Riemannian Staircase (R1) (in practice this is solved using iterative methods, such as the Truncated Newton method); (ii) the dual maximization were the dual variables are updated using the gradient ascent update (5).

**Rounding.** Upon convergence, DARS produces a matrix $Y^* = RR^T$. When deriving the standard SDP relaxation (S1) we dropped the rank-1 constraint, hence $Y^*$ cannot be written in general as $Y^* = yy^*$. The process of computing a feasible solution $\hat{y}$ for the original problem (P1) is called rounding. A standard approach for rounding consists in computing a rank-1 approximation of $Y$ (which can be done via singular value decomposition) and rounding the entries of the resulting vector in $\{-1, +1\}$. We refer to $\hat{y}$ as the rounded estimate and we call $f_{S1}$ the objective value attained by $\hat{y}$ in (P1).

**Convergence.** While dual ascent is a popular optimization technique, few convergence results are available in the literature. For instance, dual ascent is known to converge when the original objective is strictly convex [22]. Currently, we observe that DARS converges when the stepsize $\alpha$ in (5) is sufficiently small. We prove the following per-instance performance guarantees.

**Proposition 2 (Guarantees in DARS):** If the dual ascent iterations converge to a value $\hat{X}^*$ (i.e., the dual iterations reach a solution where the gradient in (5) is zero) then the following properties hold:

- let $R^*$ be a (column) rank-deficient second-order critical point of problem (R1) with $L_i = L + \sum_{i=1}^N \lambda_i U_i$, then the matrix $Y^* = (R^*)R^*$ is an optimal solution for the standard SDP relaxation (S1);
- let $f^*_{S1}$ be the (optimal) objective value attained by $Y^*$ in the standard SDP relaxation (S1), $f^*_{P1}$ be the optimal objective of (P1), and $\hat{f}_{S1}$ the objective attained by the rounded solution $\hat{y}$, then it holds $\hat{f}_{S1} - f^*_{P1} \leq f^*_{S1} - \hat{f}_{S1}$.

The proof of Proposition 2 is given in Appendix B. The first claim in Proposition 2 ensures that when the dual ascent method converges, it produces an optimal solution for the standard SDP relaxation (S1). The second claim states that we can compute an upper-bound on how far the DARS’ solution is from optimality ($f_{S1} - f^*_{P1}$) using the rounded objective $\hat{f}_{S1}$ and the relaxed objectives $f^*_{S1}$.

**IV. FUSE: Fast Unconstrained Semidefinite Solver**

In this section we propose a more direct way to obtain a semidefinite relaxation and a remarkably faster solver. While DARS is already able to compute an approximate MAP estimate in seconds for large problems, the approach presented in this section requires two orders of magnitude less time to compute a solution of comparable quality. We first present a binary $\{0, 1\}$ (rather than $\{-1, +1\}$) matrix formulation (Section IV-A) and derive an SDP relaxation (Section IV-B). We then present a Riemannian staircase approach to solve the resulting SDP in real time (Section IV-C) and discuss performance guarantees (Section IV-D).

**A. Matrix Formulation**

In this section we rewrite the node variables $x_i \in L = \{0, 1, \ldots, K\}$ as an $N \times K$ binary matrix $X \in \{0, 1\}^{N \times K}$ that is such that if an entry in position $(i, j)$ is equal to 1, then node $i$ has label $j$ and is zero otherwise. In other words, the $i$-th row of $X$ is a binary vector that describes the label of node $i$ and has a single entry equal to 1 in position $j$, where $j$ is the label assigned to the node. This is a more intuitive parametrization of the problem and indeed leads to a more elegant matrix formulation, given as follows.
Let us define $G \in \mathbb{R}^{N \times K}$ and $H \in \mathbb{R}^{N \times N}$ be defined as follows:

\[
G = \begin{cases} 
G_i = -\delta_i e_i^T, & \text{if } i \in \mathcal{U} \\
G_i = 0^T_K, & \text{otherwise}
\end{cases} \quad H = \begin{cases} 
H_{ij} = -\bar{\delta}_{ij}, & \text{if } (i, j) \in \mathcal{B} \\
H_{ij} = 0, & \text{otherwise}
\end{cases} \quad (10)
\]

where $G_i$ is the $i$-th row of $G$, $H_{ij}$ is the entry of $H$ in row $i$ and column $j$. $\delta_i$ and $\bar{\delta}_{ij}$ are the coefficients defining the MRF, cf. eq. (1), and $e_i \in \mathbb{R}$ is a vector with a unique nonzero entry equal to 1 in position $i$ ($\bar{e}_i$ is the measured label for node $i$). Then the MAP estimator (P0) can be equivalently written as:

\[
\begin{align*}
\min_X & \quad \text{tr} \left( X^T H X \right) + \text{tr} \left( G X^T \right) \\
\text{subject to} & \quad \text{diag} \left( X X^T \right) = I_N \\
& \quad X \in \{0, 1\}^{N \times K}
\end{align*} \quad (11)
\]

The equivalence between (P0) and (11) is proven in Appendix C. We note that the constraint $\text{diag} \left( X X^T \right) = I_N$ in (11) (contrarily to the diagonal constraint in (P0)) imposes that each node has a unique label when $X \in \{0, 1\}^{N \times K}$.

### B. Novel Semidefinite Relaxation

This section presents a semidefinite relaxation of (11). Towards this goal, we first homogenize the cost by lifting the problem to work on a larger variable:

\[
V \doteq \begin{bmatrix} X & I_K \end{bmatrix} \quad \text{note: } V V^T = \begin{bmatrix} X X^T & X & I_K \end{bmatrix}
\]

where $I_K$ is the $K \times K$ identity matrix. The reparametrization is given as follows.

**Proposition 3 (Binary Matrix Formulation of MAP-MRF):**

Let us define $Q \doteq \begin{bmatrix} H & \frac{1}{2} G \\ \frac{1}{2} G & 0 \end{bmatrix} \in \mathbb{R}^{(N+K) \times (N+K)}$. Then the MAP estimator (11) can be rewritten as:

\[
\begin{align*}
\min_V & \quad \text{tr} \left( V^T Q V \right) \\
\text{subject to} & \quad \text{diag} \left( V V^T \right)_{\text{tl}} = I_N \\
& \quad V \in \{0, 1\}^{(N+K) \times (K+K)}
\end{align*} \quad (P2)
\]

where $\left[ V V^T \right]_{\text{tl}}$ denotes the $(N \times N)$ top-left block of the matrix $V V^T$, cf. (12), and $\text{diag} \left( V V^T \right)_{\text{br}}$ denotes the $(K \times K)$ bottom-right block of $V V^T$, cf. (12).

At this point it is straightforward to derive a semidefinite relaxation, by noting that $\text{tr} \left( V^T Q V \right) = \text{tr} \left( Q V V^T \right)$ and by observing that $V V^T$ is a $(N + K) \times (N + K)$ symmetric positive semidefinite matrix of rank $K$.

**Proposition 5 (Semidefinite Relaxation):** The following SDP is a convex relaxation of the MAP estimator (P2):

\[
\begin{align*}
\min_Z & \quad \text{tr} \left( Q Z \right) \\
\text{subject to} & \quad \text{diag} \left( Z \right)_{\text{tl}} = I_N \\
& \quad \text{diag} \left( Z \right)_{\text{br}} = I_K \\
& \quad Z \succeq 0
\end{align*} \quad (S2)
\]

where $\left[ Z \right]_{\text{tl}}$ and $\left[ Z \right]_{\text{br}}$ are the $(N \times N)$ top-left block and the $(K \times K)$ bottom-right block of the matrix $Z$, respectively, and we dropped the rank-$K$ constraint for $Z$.

### C. Accelerated Inference via the Riemannian Staircase

We now present a fast specialized solver to solve the SDP (S2) in real time and for large problem instances. Similarly to Section III-B, we use the Burer-Monteiro method [20], which replaces the matrix $Z$ in (S2) with a rank-$r$ product $R R^T$:

\[
\begin{align*}
\min_R & \quad \text{tr} \left( Q R R^T \right) \\
\text{subject to} & \quad \text{diag} \left( R R^T \right)_{\text{tl}} = I_N \\
& \quad \left[ R R^T \right]_{\text{br}} = I_K
\end{align*} \quad (13)
\]

where $R \in \mathbb{R}^{N \times r}$ (for a suitable rank $r$), and where the constraint $Z \succeq 0$ in (S2) becomes redundant after the substitution, and is dropped.

Similarly to Section III-B, we note that the constraint set in (13) describes a smooth manifold, and in particular a product of Stiefel manifolds. Specifically, we observe that $\text{diag} \left( \left[ R R^T \right]_{\text{tl}} \right) = I_N$ can be written as $R_i R_i^T = 1$, $i = 1, \ldots, N$, which is equivalent to saying that $R_i \in \text{St}(1, r)$ for $i = 1, \ldots, N$. Moreover, denoting with $\bar{R}_i$ the block matrix including the last $K$ rows of $R$, the constraint $\left[ R R^T \right]_{\text{br}} = I_K$ can be written as $\bar{R}_i \bar{R}_i^T = I_K$, which is equivalent to saying that $\bar{R}_i \in \text{St}(K, r)$. The two observations above allow concluding that the matrix $R$ belongs to the product manifold $\text{St}(1, r)^N \times \text{St}(K, r)$. Therefore, we can rewrite (13) as an unconstrained optimization on manifold:

\[
\min_{R \in \text{St}(1, r)^N \times \text{St}(K, r)} \text{tr} \left( Q R R^T \right) \quad (R2)
\]

The formulation (R2) is non-convex but one can find local minima efficiently using iterative methods [9], [21]. We can again adapt the result from Boumal et al. [9] to conclude that rank-deficient local solutions of (R2) can be mapped back to global solutions of the semidefinite relaxation (S2).

**Proposition 6 (Optimality Conditions for R2):**

**Corollary 8 in [9]:** If $R \in \text{St}(1, r)^N \times \text{St}(K, r)$ is a (column) rank-deficient second-order critical point of problem (R2), then $R$ is a global optimizer of (R2), and $Z^\star = R R^T$ is a solution of the semidefinite relaxation (S2).

Similarly to Section III-B, we can adopt a Riemannian staircase method, where one solves (R2) for increasing values of $r$ till a rank-deficient solution is found. In all tests we performed, a single step of the staircase ($r = K + 1$ in this case) was sufficient to find a rank-deficient solution.

### D. FUSES: Summary, Convergence, and Guarantees

We name FUSES (Fast Unconstrained SEmidefinite Solver) the approach presented in this section. Contrarily to DARS, FUSES is extremely simple and only requires solving the rank-restricted problem (R2), which can be solved using iterative methods, such as the Truncated Newton method. Besides its simplicity, FUSES is guaranteed to converge to the solution of the SDP (S2) for increasing values of the rank $r$ (Proposition 6).

**Rounding.** Upon convergence, FUSES produces a matrix $Z^\star$. Similarly to DARS, we obtain a rounded solution by computing a rank-$K$ approximation of $Z^\star$ and rounding the corresponding matrix in $\{0, 1\}$ (i.e., we assign the largest element in each row to 1 and we zero out all the others).
We denote with \( \hat{X} \) the resulting estimate and we call \( \hat{f}_{S2} \) the objective value attained by \( \hat{X} \) in (1).

Since the SDP (S2) is a relaxation of the MAP estimator (P2), it is straightforward to prove the following proposition.

**Proposition 7 (Guarantees in FUSES):** Let \( f_{S2}^* \) be the optimal objective attained by \( \mathbf{Z}^* = (\mathbf{R})(\mathbf{R})^\top \) in (S2), \( f_{P2}^* \) be the optimal objective of (P2), and \( \tilde{f}_{S2} \) be the objective attained by the rounded solution \( \tilde{X} \), then \( f_{S2}^* - f_{P2}^* \leq \tilde{f}_{S2} - f_{S2}^* \).

Again, we can use Proposition 7 to compute how far the solution computed by FUSES is from the optimal objective attained by the MAP estimator.

**V. Experiments**

This section evaluates the proposed MRF solvers on semantic segmentation problems, comparing their performance against the state of the art.

**A. FUSES and DARS: Implementation Details**

We implemented FUSES and DARS in C++ using Eigen’s sparse matrix manipulation and leveraging the optimization suite developed in [21]. Sparse matrix manipulation is crucial for speed and memory reasons, since the involved matrices are very large. For instance in DARS, the matrix \( \mathbf{L} \lambda \) in (R1) has size \((NK + 1) \times (NK + 1)\) where typically \( N > 10^7 \) and \( K > 20 \).

We initialize the rank \( r \) of the Riemannian Staircase to be \( r = 2 \) for DARS and \( r = K + 1 \) for FUSES (this is the smallest rank for which we expect a rank-deficient solution). The Riemannian optimization problems (R1) and (R2) are solved iteratively using the truncated-Newton trust-region method. We refer the reader to [23] for a description of the implementation of a truncated-Newton trust-region method. As in [23], we use the Lanczos algorithm to check that (R1) and (R2) converged to rank-deficient second-order critical points, which are optimal according to Proposition 1 and Proposition 6 respectively. If the optimality condition is not met, the algorithm proceeds to the next step of the Riemannian staircase, repeating the optimization with the rank \( r \) increased by 1. In all experiments, FUSES finds an optimal solution in the first iteration of the staircase (\( r = K + 1 \)), while we observed that the rank in DARS (initially \( r = 2 \)) sometimes increases to \( 6 - 8 \). In DARS, we limit the number of dual ascent iterations to \( T = 1000 \), and we terminate iterations when the gradient in (5) has norm smaller than 0.1. Using a constant stepsize \( \alpha = 0.05 \) ensured convergence in all tests.

**B. Setup, Compared Techniques, and Performance Metrics**

**Setup.** We evaluate FUSES and DARS using the Cityscapes dataset [24], which contains a large collection of images of urban scenes with pixel-wise semantic annotations. The annotations include 30 semantic classes (e.g., road, sidewalk, person, car, building, vegetation). We first extract superpixels from the images using OpenCV (we obtain around 1000 superpixels per image, unless specified otherwise). Then, the unary terms are obtained using Bonnet [23], which uses a CNN to obtain pixel-wise segmentation (Bonnet only uses 20 classes for classification purposes); the unary potential for each superpixel is set based on the majority of labels.

**Compared techniques.** We compare the proposed techniques against three state-of-the-art methods: \( \alpha \)-expansion [6] (label: \( \alpha \)-exp), Loopy Belief Propagation [8] (label: LBP) and Tree-Reweighted Message Passing [7] (label: TRW-S). We use the implementation of these methods available in the newly-released OpenGM2 library [26].

**Performance metrics.** We evaluate the results in terms of suboptimality, accuracy, and CPU time. We measure the suboptimality using three metrics: the percentage of optimal labels, the percentage relaxation gap, and the percentage rounding gap. The optimal labels are those that agree with the optimal solution of (P0). The relaxation gap is \( (f_{P1} - f_{S1})/f_{P1} \) for DARS, and \( (f_{P2} - f_{S2})/f_{P2} \) for FUSES. The rounding gap is \( (f_{S1} - f_{P1})/f_{P1} \) for DARS, and \( (f_{S2} - f_{P2})/f_{P2} \) for FUSES. We compute the optimal labels (and the corresponding optimal objective) using a commercial tool for integer programming, CPLEX [27]. The runtime of CPLEX increases exponentially in the problem size hence we can only use it offline for benchmarking the proposed solvers. We measure the accuracy using the Intersection over Union (IoU) metric [28], and record the CPU time for each compared technique.

**C. Semantic Segmentation Results**

Fig. 2 shows a typical execution of the algorithms for a single image in the Cityscapes dataset. Fig. 2(a) shows the convergence of FUSES, reporting the relaxed objective attained by iteratively solving (R2) (FUSES-relaxed), the objective of the corresponding rounded estimate at each iteration (FUSES-rounded), and the optimal cost attained by CPLEX (Exact). The approach converges in few milliseconds, and the corre-
Table I provides statistics describing the performance of the compared techniques on the Cityscapes’ Lindau dataset over 59 images (we use approximately 1000 superpixels). We show the percentage of optimal labels (“Optimal Labels” column), the relaxation gap (“Relax Gap” column), and the rounding gap (“Round Gap” column). The tables show that FUSES and DARS have comparable suboptimality (typically larger than the other compared techniques). FUSES and DARS produce optimal assignments for most of the nodes in the MRF, and attain a rounded cost within 0.2% of the optimum. The IoU (“Accuracy” column) shows that all the techniques have comparable accuracy (around 34%). All the compared techniques produce more accurate results than the CNN-based segmentation produced by Bonnet, which has IoU equal to 30.11% on this dataset. Note that the accuracy depends on the parameters of the MRF ($\delta_i$ and $\delta_{ij}$) besides depending on the solver. FUSES is the fastest MRF solver and can compute a solution in milliseconds, while not relying on any initial guess. Table II shows that even with 2000 superpixels, the advantages of FUSES remain. Fig. 3 shows qualitative segmentation results obtained using the proposed techniques. We also attempted to use a general-purpose SDP solver, CVX [10], for our evaluation: with only 200 superpixels, CVX requires more than 50 minutes to solve the SDP ([11]), while for 1000 superpixels it crashes due to excessive memory usage.

Fig. 3(a) shows the relaxation gap for FUSES and DARS for increasing number of nodes; we control the number of nodes by controlling how many superpixels each image is divided in. The relaxation gap decreases for increasing number of nodes, which is a desirable feature since one typically solves large problems (>1000 nodes). The relaxation gap in FUSES is slightly larger: in hindsight, we traded-off suboptimality for fast computation. Fig. 3(b) shows the relaxation gap for FUSES and DARS for increasing number of labels; we artificially reduce the number of labels in Cityscapes for this test. The quality of both relaxations does not degrade significantly for increasing number of labels.

This section reviews inference techniques (Section VI-A and VI-B) and applications (Section VI-C) for pairwise MRFs including work on semantic segmentation. Our presentation is based on [1], [2], [3] but also covers more recent work on MRFs and semantic segmentation.

A. Exact Inference in MRFs

Efficient Algorithms. Inference in MRFs is intractable in general. However, particular instances of the problem are solvable in polynomial time. In particular, the Ising model can be solved exactly in polynomial time via graph cut [29], [30]. Note that graph cut algorithms are exact when binary potentials are “attractive”, i.e., $\delta_{ij} \geq 0$ in (1) (priors encourage nearby nodes to have the same label). MRFs with repulsive potentials ($\delta_{ij} < 0$) are intractable in general [31]. A more general (necessary and sufficient) condition that ensures optimality of graph cut for binary pairwise MRFs with classes $\mathcal{L} = \{0; 1\}$ is the regularity condition:

$$E_{ij}(0, 0) + E_{ij}(1, 1) \leq E_{ij}(0, 1) + E_{ij}(1, 0) \quad (14)$$

for any $(i, j) \in E$, see Lemma 3.2 and Theorem 3.1 in [31]. The regularity condition in eq. (14) is a special case of submodularity, and indeed the corresponding potentials are also called submodular [31], [32], [33].

For multi-label pair-wise MRFs, exact solutions exist for the case when the binary potentials are convex functions of the labels [34], [35], [36] and for the case where the binary

| Method | Optimal Labels (%) | Relax Gap (%) | Round Gap (%) | Accuracy (% IoU) | Runtime (ms) |
|--------|---------------------|---------------|---------------|-----------------|--------------|
| FUSES  | 98.743 | 2.284 | 0.120 | 34.07 | 17.75 |
| DARS   | 99.134 | 0.644 | 0.075 | 34.21 | 2013.46 |
| $\alpha$-exp | 99.825 | - | 6.45e-3 | 34.50 | 49.54 |
| LBP    | 99.865 | - | 3.53e-3 | 34.51 | 102.92 |
| TRW-S  | 99.992 | - | 2.794e-5 | 34.58 | 49.55 |

**TABLE I**

**Performance on the Cityscapes’ Lindau Dataset (1000 Superpixels).**
potentials are linear and the unary potentials are convex \cite{37}. We remark that these approaches assume a linear ordering of the labels, where the potentials penalize node labels depending on their label distance $|x_i - x_j|$; this means that choosing $x_i = 1$ and $x_j = 3$ incurs a larger penalty than choosing $x_i = 1$ and $x_j = 2$; on the other hand, the Potts model in eq. (1) penalizes in the same way any class mismatch $x_i \neq x_j$. Assuming a linear ordering is often unrealistic in practice; for instance, in semantic segmentation the classes (e.g., cat, table, car) do not admit a linear order in general. Moreover, convexity is a strong assumption for several MRF applications, such as depth reconstruction, where nonconvex costs have the desirable property of being discontinuity-preserving \cite{31} contrarily to convex ones, which tend to smooth out depth discontinuities. Inference in multi-class MRF based on the Potts model is NP-hard, see \cite{38}.

In the special case where the topology of the MRF is a chain (e.g., when the MRF describes a 1D signal or sequence), or more generally a tree, Dynamic Programming provides an optimal MAP estimate in polynomial time, see \cite{33, 39}. Related work \cite{40, 41} also extends dynamic programming to certain families of graphs with cycles and small cliques.

**Global Integer Solvers.** The energy minimization problem \cite{40} is a quadratic integer program and can be easily reformulated as a binary optimization problem \cite{42, 43, 44}. Integer programming is NP-hard in general, but one may still resort to state-of-the-art integer solvers (e.g., CPLEX \cite{27}) for moderate-size instances. For quadratic and linear programs, integer solvers based on cutting plane methods or branch & bound are able to produce solutions for problems with few hundred variables relatively quickly (i.e., in few seconds), but become unacceptably slow for larger problems. A Branch-and-Cut approach is proposed in \cite{45}. An evaluation and a broader review of integer programming for MRFs is given in \cite{43}.

**B. Approximate and Local Inference in MRFs**

**Iterative Local Solvers and Meta-heuristics.** Local solvers start at a given initial guess and iteratively try to converge to a local optimum of the cost function. Early work includes the Iterative Local Modes (ICM) of Besag \cite{46}, which at each iteration greedily changes the label of a node in order to get the largest decrease in the cost. ICM is known to be very sensitive to the quality of the initial guess \cite{1}. In order to improve convergence, Geman and Geman \cite{47} use Simulated Annealing to perform inference in MRFs. Simulated Annealing requires exponential time to converge in theory and is notoriously slow in practice \cite{48}.

**Graph Cuts and Move-Making Algorithms.** While graph cut methods are able to compute globally optimal solutions in binary pairwise MRFs with submodular potentials (Section VI-A), they are only able to converge to local minima in non-submodular binary MRFs or in multi-class MRFs. For the binary case, related works \cite{49, 32} develop schemes to approximately solve MRFs with non-submodular potentials. Regarding the multi-class case, popular graph cut methods include the swap-move ($\alpha$-$\beta$-swap) and the expansion-move ($\alpha$-expansion) algorithms, both proposed in \cite{6}. At each inner iteration, these algorithms solve a binary segmentation problem using graph cut, while the outer loop attempts to reconcile the binary results into a coherent multi-class segmentation. Boykov et al. \cite{6} show that the swap-move algorithm is applicable whenever the smoothness potentials are semi-metric (i.e., $E_{ij}(x_i, x_j) = E_{ij}(x_i, x_j) \geq 0$ and $E_{ij}(x_i, x_j) = 0 \iff x_i = x_j$), and the expansion-move algorithm is applicable whenever the smoothness potentials are metric (i.e., they are semi-metric and also satisfy the triangle inequality $E_{ij}(x_i, x_j) \leq E_{ik}(x_i, x_k) + E_{kj}(x_k, x_j)$); these conditions are further generalized in \cite{31}. Under these conditions, Boykov et al. \cite{6} show that these graph cut methods produce “strong” local minima, i.e., local minima where no allowed move is able to further reduce the cost. Moreover, these techniques produce a local solution with is proven to be within a known factor from the global minimum \cite{6}. When these conditions are not satisfied, approximations of the cost function can be used \cite{50, 38}. Komodakis and Tziritas \cite{51} draw connections between move-making algorithms and the dual of linear programming relaxations. Kumar and Koller \cite{52, 53} propose a move-making approach that applies to the semi-metric case and attains the same guarantees of the linear relaxation (see paragraph below) in the metric case. Faster algorithmic variants are proposed by Alahari et al. \cite{54}. Lempitsky et al. \cite{55} provide a low-complexity algorithm (LogCut) that requires an offline learning step. A summary of the MRF formulations that can be solved exactly or within a constant factor from the global minimum via graph cut is given in \cite{31}. When the potentials do not satisfy the conditions for applicability of graph cut methods, approximate versions of these techniques can be still applied \cite{50} but the corresponding performance bounds no longer hold.

**Message-Passing Techniques.** Message passing techniques adjust the MAP estimate at each node in the MRF via local information exchange between neighboring nodes. A popular message passing technique is belief propagation \cite{50}, which results in exact inference in graphs without loops, but is also applicable to generic graphs \cite{57, 58} (loopy belief propagation, or LBP in short). LBP is not guaranteed to converge in presence of cycles, but if convergence is attained LBP returns “strong” local minima \cite{58, 59}. Tree-Reweighted Message Passing \cite{1} (TRW-S) is another popular message-passing algorithm which is also able to estimate a lower-bound on the cost that can be used to assess the quality of the solution. Also in this case the estimate is not guaranteed to converge and may oscillate. Message-passing techniques do not necessarily return integer solutions, hence the resulting estimates need to be rounded, see \cite[Section 4.5]{3}.

**Linear Programming (LP) Relaxations.** These techniques relax the optimization to work on continuous labels rather than discrete ones. Early relaxation techniques include the LP relaxation of the local polytope \cite{17}, which is typically

\footnote{Note that both the Potts model and the truncated $\ell_2$ distance are metrics.}

\footnote{Conditional Random Fields (CRFs) are a special case of MRFs, where the binary terms, rather than being smoothness priors, are data driven.}
applicable only to small problem instances [3]. Kleinberg and Tardos [61] provide suboptimality guarantees for LP relaxations with metric potentials. Gupta and Tardos [62] extend these results considering a truncated linear metric. Chekuri et al. [63] and Werner [64] further refine the suboptimality bounds. Komodakis and Tziritas [65] consider the case of semi-metric and non-metric potentials and derive primal-dual methods to efficiently solve the resulting LP relaxations. Sontag and Jaakkola [66] propose a cutting-plane algorithm for optimizing over the marginal polytope. Other specialized solvers to attack larger instances have also been proposed, including block-coordinate ascent [67], subgradient methods based on dual decomposition [68], [69], [70]. Alternating Directions Dual Decomposition [71], and others [72], [73], [74]. The performance of these techniques is typically sensitive to the choice of the parameters (e.g., stepsize) and can only ensure local convergence [3]. For binary pairwise MRFs, LP relaxation over the local polytope can be solved efficiently by reformulating it as a maximum flow problem, see the roof duality (or QPBO) approach of Rother et al. [75]. LP relaxations typically do not produce an integer solution, therefore the corresponding solutions need to be rounded. Moreover, they are tightly coupled with message-passing algorithms, see [3] Section 4.3. Kumar et al. [4] provide a comparison between linear, quadratic, and second-order cone programming relaxations, showing that the linear relaxation dominates the others.

Spectral and Semidefinite Relaxations. These techniques typically rephrase inference over an MRF in terms of a binary quadratic optimization problem [16], which can be then relaxed to a convex program (more details in Section II-B). Shi and Malik [76] propose a spectral relaxation for image segmentation; more recently, spectral segmentation is used by Aksoy et al. [77]. Keuchel et al. [8] introduce SDP relaxations to several computer vision applications and use interior-point methods and randomized hyperplane techniques to obtain integer solutions, leveraging the celebrated result of Goemans and Williamson [78], which bounds the suboptimality of the resulting solutions. SDP relaxations are known to provide better solutions than spectral methods [5], [16]. While early approaches also recognized the accuracy of SDP relaxations with respect to commonly used alternatives (e.g., [4]), the computational cost of general-purpose SDP solvers prevented widespread use of this technique beyond problems with few hundred variables [5]. Keuchel et al. [15] propose an approach to reduce the dimension of the problem via image preprocessing and superpixel segmentation. Concurrently, Torr [79] proposes the use of SDP relaxations for pixel matching problems. Schellewald and C. Schnörr [80] suggest a similar SPD relaxation for subgraph matching in the context of object recognition. Heiler et al. [81] propose to add constraints in the SDP relaxation to enforce priors (e.g., constrain the number of pixels in a class, or force set of pixels to belong to the same class). Olsson et al. [16] develop a spectral subgradient method which is shown to reduce the relaxation gap of spectral relaxations. Huang et al. [82] use an Alternating Direction Method of Multipliers to speed up computation, while Wang et al. [83], [84] develop a specialized dual solver. Frostig et al. [85] resort to non-convex optimization to approximate the SDP solution, while Wang et al. [86] consider fully-connected CRFs and propose fast solvers for the case where the pairwise potentials admit a low-rank decomposition. We remark that the approach to derive the SDP relaxation is common to all papers above and follows the line of Section II-B Wainwright and Jordan [87] use semidefinite programming to approximately compute the marginal distributions in a graphical model. More generally, semidefinite programming has been a popular way to relax combinatorial integer programming problems [88], [89] and assignment problems [90], [91].

C. Applications

Overview. MRFs have been successfully used in several application domains including computer vision, computer graphics, machine learning, and robotics. Popular applications include image denoising, inpainting, and super-resolution [36], [38], [50], [57], image segmentation (reviewed below), stereo reconstruction [92], [96], [93], [95], [94], [97], panorama stitching and digital photomontages [98], image/video-texture synthesis [99], multi-camera scene reconstruction [100], voxel occupancy estimation [101], non-rigid point matching and registration [102], [16], medical imaging [103], [104]. In stereo reconstruction, the labels are the disparities at each pixel and the binary potentials are function of the absolute color differences at nearby pixels. Birchfield and Tomasi [48] provide a comparison of graph-cut methods for stereo reconstruction, while Tappen and Freeman [105] compare graph cut and LBP; Kolomogorov and Rother [106] evaluate TRW-S, LBP, and graph cut. Szeliski et al. [1] compare several techniques on stereo reconstruction, photomontage, image segmentation, and image denoising benchmarks. The study concludes the the expansion move algorithm typically outperforms the swap move algorithm, while ICM performs poorly in practice. In general, the best approach may depend on the application: for instance, the expansion move algorithm is the best performer for the photomontage benchmark, while expansion move and TRW-S perform the best on the depth reconstruction benchmark. A broader evaluation is presented in [3], which also provides a C++ library, OpenGM2 [26], that implements several inference algorithms.

Semantic Segmentation. Semantic segmentation methods assign a semantic label to each “region” in an RBG image (2D segmentation), RBG-D image, or 3D model (3D segmentation). Depending on the approach, labels can be assigned to single pixels/voxels, superpixels, or keypoints [3]. Since semantic segmentation is typically modeled as an MRF, the literature review in Sections VI-A-VI-B already covers several work in segmentation, and indeed segmentation (together with depth reconstruction) is a typical benchmark for inference in MRFs, see [1], [2], [3], [33], [12] and the references therein. Therefore, the goal of this section is to (i) provide a brief taxonomy of semantic segmentation problems, and (ii) review semantic segmentation techniques that do not directly use MRFs. The corresponding literature is vast, and we refer the reader to the excellent survey of Zhu et al. [12] for a broader review of related work.

Taxonomy. Semantic segmentation is different from cluster-
ing, which groups pixels based on similarities without necessarily associating a given semantic label to each group (this is sometimes called non-semantic, unsupervised, or bottom-up segmentation \[107, 112\]). While semantic segmentation classifies image regions into semantic classes, instance segmentation also attempts to discern multiple objects belonging to the same class. In full analogy with MRFs, segmentation problems can be divided in binary segmentation problems (where only two classes, foreground and background, are segmented) and multi-class segmentation problems, where more than two labels are allowed. We can further divide the literature depending on the type of input data the segmentation operates on, including isolated RGB images (most common setup in computer vision), stereo images \[38\], RGB-D images \[108, 110\], volumetric 3D data (e.g., volumetric X-ray CT images \[110\]), or 3D voxel-based models \[111\], or multiple RGB images; the latter setup is typically referred to as co-segmentation \[107, 113, 115\] (for generic unordered images), or temporal (or video) segmentation \[114\] (if images are collected over time). Thoma \[107\] also categorizes the segmentation problems into active (where one can influence the data collection mechanism, as it happens in robotics), passive (where the input data is given), and interactive (where a human user provides coarse information to the segmentation algorithms).

Other Approaches. Traditional approaches for semantic segmentation work by extracting and classifying features in the input data, and then enforcing consistency of the classification across pixels (e.g., using MRFs or other models). Common features include pixel color, histogram of oriented gradients, SIFT, or textons, to mention a few \[107, 115\]. Shotton et al. \[116, 117\] use textons and Random Decision Forests for semantic segmentation. Yang et al. \[118\] use Support Vector Machine (SVM) demonstrating competitive performance in the PASCAL segmentation challenge \[119\]. A latent SVM model is used by Felszenwalb et al. \[120\] to detect objects using deformable part models. Winn and Shotton \[121\] use a CRF-based algorithm, named the Layout Conditional Random Field (LayoutCRF), to detect and segment objects from their parts; the approach is further generalized by Hoiem et al. \[122\]; Shotton et al. \[123\] use textons within a CRF model for object segmentation. Kumar et al. \[124\] use MRFs to detect and segment objects in an image. Bray et al. \[125\] concurrently segment and estimate the 3D pose of a human body from multiple views. Higher-order MRF formulations are also used for semantic segmentation, see the work by Kohli and co-authors \[126, 127, 128\] and the review \[3\]. Approaches for interactive segmentation include intelligent scissors \[129\], active contour models \[130, 131\] (based on dynamic programming), and graph cut methods (GrabCut \[132\]). While most of the work mentioned so far operates on a discrete set of nodes of a graphical model, related work in multi-class segmentation also includes contributions modeling the problem over a continuous domain; examples of such efforts include the variational method of Lellmann et al. \[133\], and the anisotropic diffusion method of Kim et al. \[113\]; see the chapter by Cremers et al. \[134\] for a recent survey. More recently, deep convolutional neural networks have become a popular solution for semantic segmentation, see the recent review of Garcia-Garcia et al. \[28\].

VII. CONCLUSION

We propose fast optimization techniques to solve two semidefinite relaxations of maximum a posteriori inference in Markov Random Fields (MRFs). The first technique, named DARS (Dual Ascent Riemannian Staircase), provides a scalable solution for the standard SDP relaxation proposed in the literature. The second technique, named FUSES (Fast Unconstrained SEMidefinite Solver), is based on a novel relaxation. We test the proposed approaches in semantic segmentation problems and compare them against state-of-the-art MRF solvers, including move-making and message-passing methods. Our experiments show that (i) FUSES and DARS produce near-optimal solutions, attaining an objective within 0.2% of the optimum, (ii) our approaches are remarkably faster than general-purpose SDP solvers, while FUSES is more than two orders of magnitude faster than DARS, (iii) FUSES is faster than local search methods while being a global solver.
formulation,” in *Proc. of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001, pp. 109–118.

[64] T. Werner, “A linear programming approach to max-sum problem: A review,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 7, pp. 1165–1179, 2007.

[65] N. Komodakis and G. Tziritas, “Approximate Labeling via Graph Cuts Based on Linear Programming,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 8, pp. 1436–1453, 2007.

[66] D. Sontag and T. Jaakkola, “New outer bounds on the marginal polytope,” in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 1393–1400.

[67] V. Kolmogorov, “Convergent Tree-Weighted Message Passing for Energy Minimization,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 10, pp. 1568–1583, 2006.

[68] N. Komodakis, N. Paragios, and G. Tziritas, “MRF Energy Minimization and Beyond via Dual Decomposition,” *pani*, vol. 33, no. 3, pp. 531–552, 2011.

[69] J. Kappes, M. Speth, G. Reinelt, and C. Schür, “Towards efficient and exact map-inference for large scale discrete computer vision problems via combinatorial optimization,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[70] M. Guignard and S. Kim, “Lagrangian decomposition: A model yielding stronger lagrangean bounds,” *Mathematical Programming*, vol. 39, no. 2, pp. 215–228, 1987.

[71] A. Martins, M. Figueiredo, P. ABgual, N. Smith, and E. Xing, “An augmented Lagrangian approach to constrained MAP inference,” in *Int. Conf. on Machine Learning (ICML)*, 2011, pp. 169–176.

[72] B. Savchynskyy, S. Schmidt, J. Kappes, and C. Schur, “Efficient mrf energy minimization via adaptive diminishing smoothing,” in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012, pp. 746–755.

[73] A. Globerson and T. Jaakkola, “Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations,” in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 553–560.

[74] D. Sontag, D. Choe, and Y. Li, “Efficiently searching for frustrated cycles in map inference,” in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012.

[75] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, “Optimizing binary mrf’s via extended roof duality,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[76] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 5, 2000.

[77] Y. Aksos, T. Oh, S. Paris, M. Pollefeys, and W. Matusik, “Semantic soft segmentation,” *SIGGRAPH*, vol. 37, no. 4, pp. 72:1–72:13, 2018.

[78] M. Goemans and D. Williamson, “Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming,” *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[79] P. H. S. Torr, “Solving markov random fields using semi definite programming,” in *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2003.

[80] C. Schellewald and C. Schönhörf, “Probabilistic subgraph matching based on convex relaxation,” in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer Berlin Heidelberg, 2005, pp. 171–186.

[81] M. Heiler, J. Schreuel, and C. Schönhörf, “Semantic Clustering for Image Segmentation with A-priori Knowledge,” in *Pattern Recognition*. Springer, Berlin, Heidelberg, 2005, pp. 309–317.

[82] P. Huang, Y. Chen, and L. Guibas, “Scalable semidefinite relaxation for maximum a posterior estimation,” in *Proc. of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI ’09, London, UK: UK: Springer-Verlag, 2000, pp. 62–71. [Online]. Available: https://doi.org/10.1007/978-3-540-89956-2_6

[83] M. Tappen and W. Freeman, “Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters,” in *Int. Conf. on Computer Vision (ICCV)*, 2003, pp. 900–907.

[84] V. Kolmogorov and C. Rother, “Comparison of Energy Minimization Algorithms for Highly Connected Graphs,” 2006.

[85] M. Thoma, “A survey of semantic segmentation,” *ArXiv Preprint: 1602.06541*, 2017.

[86] D. Zhuo, T. Sinisa, and L. Longin, “Semantic segmentation of RGB-D images with mutex constraints,” *Int. Conf. on Computer Vision (ICCV)*, pp. 1733–1741, 2015.

[87] S. Gupta, P. Arbelaez, R. Girshick, and J. Malik, “Indoor scene understanding with RGB-D images: Bottom-up segmentation, object detection and semantic segmentation,” *Int. J. of Computer Vision*, vol. 112, no. 2, pp. 133–149, 2015.

[88] S. Hu, E. Hoffman, and J. Reinhardt, “Automatic lung segmentation for accurate quantification of volumetric X-ray CT images,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 6, pp. 490–498, 2001.

[89] S. Kolundzija, E. V. Fella, E. Li, and J. Relge, “Joint semantic segmentation and 3D reconstruction from monocular video,” in *European Conf. on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 8694, 2014, pp. 703–718.

[90] J. Rother, T. Minka, and V. Kolmogorov, “Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2006, pp. 993–1000.
We first rewrite each node variable $x_i \in \mathcal{L} \equiv \{1, \ldots, K\}$ in (P0) as a vector $w_i \in \{-1,+1\}^K$, such that $w_i$ has a single entry equal to +1 (all the others are −1), and if the j-th entry of $w_i$ is +1, then the corresponding node has label j. Each vector $w_i \in \{-1,+1\}^N$ is a valid label assignment as long as there is a unique entry equal to +1, or, equivalently, $1^T w_i = 2 - K, \ i = 1, \ldots, N$. Using this vector parametrization we rewrite the unary and binary potentials (1) as:

$$E_i(w_i) = \frac{\delta}{2} (1 - e^T x_i, w_i), \quad (15)$$

$$E_{ij}(w_i, w_j) = \frac{\delta_{ij}}{2} (w_i^T w_j - K + 2)$$

where $e_{x_i}$ is a vector of all zeros, except the entry in position $x_i$ (measured class label for node i), which is equal to +1. The reparametrization of the unary potentials in (15) can be seen to be the same as (1) by observing that $e_{x_i}^T w_i = +1$ if $x_i = x_i$ or −1 otherwise; similarly, the reparametrization of the binary potentials follows from the fact that $w_i^T w_j = K$ if $x_i = x_j$ or $K - 2$ otherwise.

Using (15), we rewrite Problem (P0) as:

$$\min_{w_i, i = 1, \ldots, N} \sum_{i \in U} \frac{\delta}{2} e^T T x_i, w_i + \sum_{(i,j) \in B} \frac{\delta_{ij}}{2} w_i^T w_j$$

subject to $w_i \in \{-1,+1\}^K$

$$1^T w_i = 2 - K, \ i = 1, \ldots, N$$

where we dropped the constant terms from (15) (which are irrelevant for the optimization), and where the constraint $1^T w_i = 2 - K$ enforces each vector to have at most one entry equal to +1 (i.e., we assign a single label to each node).

In order to obtain Problem (2), we adopt a more compact notation by stacking all vectors $w_i$, with $i = 1, \ldots, N$, in a single $NK$-vector $w = [w_1^T w_2^T \ldots w_N^T]^T$, and note that the cost function (16) is quadratic in the entries of $w$. Therefore, we rewrite problem (16) as:

$$\min_w 2b^T w + w^T A w$$

subject to $w \in \{-1,+1\}^{NK}$

$$1^T w_i = 2 - K, \ i = 1, \ldots, N$$

where $A$ is an $NK \times NK$ symmetric block matrix, and $b$ is an $NK$-vector, and $u_i = e_i^T \otimes 1_K^T$, where $e_i$ is an $N$-vector which is all zero, except the i-th entry which is one, $1_K$ is a $K$-vector of ones, and $\otimes$ is the Kronecker product. The constraint $1^T w_i = 2 - K$ simply rewrites the constraint $1^T w_i = 2 - K$ in (16). The reader can also verify by inspection that the following choice of $A$ and $b$ ensures that the objective in (17) is the same as (16):

$$b \doteq \begin{cases} [b_i] = -\frac{\delta}{2} e_{x_i}, & \text{if } i \in U \\ [b_i] = 0_K, & \text{otherwise} \end{cases} \quad (18)$$

$$A \doteq \begin{cases} [A]_{ij} = -\frac{\delta_{ij}}{2}, & \text{if } (i,j) \in B \\ [A]_{ij} = 0_{K \times K}, & \text{otherwise} \end{cases}$$

where $b$ stacks $N$ subvectors of size $K$, $[b_i]$ is the i-th subvector of $b$, $[A]_{ij}$ is the $K \times K$ block of $A$ in block row $i$ and block column $j$, and $1_K$ is the identity matrix of size $K$.

Now we observe that for a scalar $\omega$, we can equivalently write $\omega \in \{-1,+1\}$ as $\omega \in \{\mathbb{R} : \omega^2 = 1\}$. Moreover, we
note that the diagonal of the matrix $ww^T$ contains the squares of every entry of $w$. Combining these two observations, we rewrite problem (16) equivalently as:

$$\min_w \quad w^T A w + 2b^T w \\
\text{subject to} \quad \text{diag}(ww^T) = 1_{NK},$$

(19)

which is the same as (2), concluding the proof.

**Appendix B: Proof of Proposition 2**

Here we assume that the dual ascent iterations of DARS converged to a value $\lambda^*$ (i.e., the dual iterations reach a solution where the gradient in (5) is zero) and since the gradient with respect to the $i$-th dual variable is $\nabla L_i(Y; \lambda) = \text{tr} (U_i Y^*) - K + 2$, it follows $\text{tr} (U_i Y^*) = K - 2$, for $i = 1, \ldots, N$. This implies that $d(\lambda^*) = \text{tr} (LY^*)^{(14)}$ and (22) becomes:

$$\text{tr} (LY^*) \leq f_{S1}^*$$

(23)

Now we note that since $Y^*$ satisfies $\text{tr} (U_i Y^*) = K - 2$, for $i = 1, \ldots, N$, it is a feasible solution for (S1). Therefore by optimality of $f_{S1}^*$ it holds:

$$f_{S1}^* \leq \text{tr} (LY^*)$$

(24)

Combining (23) and (24) it follows $\text{tr} (LY^*) = f_{S1}^*$, i.e., $Y^*$ attains the optimal objective in (S1), proving the first claim.

To prove the second claim, we observe that (S1) is a relaxation of (P1), hence it holds:

$$f_{S1}^* \leq f_{P1}^* \iff -f_{P1}^* \leq -f_{S1}^*$$

(25)

Adding $\hat{f}_{S1}$ to both sides of (25) yields the desired inequality $\hat{f}_{S1} - f_{P1}^* \leq \hat{f}_{S1} - f_{S1}^*$, concluding the proof of Proposition 2.

**Appendix C: Proof of Proposition 3**

Here we prove that solving (11) is equivalent to solving (P0), in the sense that the solution set of a problem is in 1-to-1 correspondence with the solution set of the other. Towards this goal, we show that (11) can be simply obtained as a reparametrization of (P0). Let us first consider the constraints in (11), and note that the equality $\text{diag}(XX^T) = 1_N$ can be written explicitly as $\|X_i\|^2 = 1$ for $i = 1, \ldots, N$, where $X_i$ is the $i$-th row of $X$; since the $i$-th row $X_i$ describes the label assignment of node $i$ and is a binary vector, the constraint $\|X_i\|^2 = 1$ enforces $X_i$ to have a unique nonzero element, hence matching the requirement of assigning a unique label to each node in (P0).

We are only left to prove that the two objective functions are equivalent. For this purpose, we observe that the objective in (P0), with the choice of potentials in (1), can be written as a function of the rows of $X$ as:

$$E(X) = \sum_{i \in U} \delta_i (1 - e_i^T X_i^T) + \sum_{(i,j) \in B} \delta_{ij} (1 - X_i X_j^T)$$

(26)

where $\equiv$ denotes equality up to constant (irrelevant for the optimization). Using definitions (10), by inspection we can verify that $\text{tr} (G X^T) = \sum_{i \in U} -\delta_i e_i^T X_i^T$ and that

$$\text{tr} (X^T H X) = \text{tr} (H X X^T) = \sum_{i=1,\ldots,N} H_{ij} [X X^T]_{ij}$$

$$= \sum_{j=1,\ldots,N} H_{ij} X_i X_j^T \equiv \sum_{(i,j) \in B} -\delta_{ij} X_i X_j^T$$

which demonstrates the equality (up to constant) of the objective functions in (P0) and (11), concluding the proof.