Global MAP-Optimality by Shrinking the Combinatorial Search Area with Convex Relaxation

Bogdan Savchynskyy¹ Jörg Kappes² Paul Swoboda² Christoph Schnörr¹,²
¹Heidelberg Collaboratory for Image Processing, Heidelberg University, Germany
bogdan.savchynskyy@iwr.uni-heidelberg.de
²Image and Pattern Analysis Group, Heidelberg University, Germany
{kappes,swoboda,schnoerr}@math.uni-heidelberg.de

Abstract

We consider energy minimization for undirected graphical models, also known as the MAP-inference problem for Markov random fields. Although combinatorial methods, which return a provably optimal integral solution of the problem, made a significant progress in the past decade, they are still typically unable to cope with large-scale datasets. On the other hand, large scale datasets are often defined on sparse graphs and convex relaxation methods, such as linear programming relaxations then provide good approximations to integral solutions.

We propose a novel method of combining combinatorial and convex programming techniques to obtain a global solution of the initial combinatorial problem. Based on the information obtained from the solution of the convex relaxation, our method confines application of the combinatorial solver to a small fraction of the initial graphical model, which allows to optimally solve much larger problems.

We demonstrate the efficacy of our approach on a computer vision energy minimization benchmark.

1 Introduction

The focus of this paper is energy minimization for Markov random fields. In the most common pairwise case this problem reads

$$\min_{x \in X_G} E_G(x) := \min_{x \in X_G} \sum_{v \in V_G} \theta_v(x_v) + \sum_{uv \in E_G} \theta_{uv}(x_u, x_v),$$ (1)

where $G = (V_G, E_G)$ denotes an undirected graph with the set of nodes $V_G \ni v$ and the set of edges $E_G \ni uv$; variables $x_v$ belong to the finite label sets $X_v$, $v \in V_G$; potentials $\theta_v : X_v \to \mathbb{R}$, $\theta_{uv} : X_u \times X_v \to \mathbb{R}$, $v \in V_G, uv \in E_G$, are associated with the nodes and the edges of $G$ respectively. We denote by $X_G$ the Cartesian product $\otimes_{v \in V_G} X_v$.

Problem (1) is known to be NP-hard in general, hence existing methods either consider its convex relaxations or/and apply combinatorial techniques such as branch-and-bound, combinatorial search, cutting plane etc. on top of convex relaxations. The main contribution of this paper is a novel method to combine convex and combinatorial approaches to compute a provably optimal solution. The method is very general in the sense that it is not restricted to a specific convex programming or combinatorial algorithm, although some algorithms are more preferable than others. The main restriction of the method is the neighborhood structure of the graph $G$: it has to be sparse. Basic grid graphs of image data provide examples satisfying this requirement. The method is applicable also to higher-order problems, defined on so called factor graphs [1], however we will concentrate mainly on the pairwise case to keep our exposition simple.

Underlying idea. Fig. 1 demonstrates the main idea of our method. Let $A$ and $B$ be two subgraphs covering $G$. Select them so that the only common nodes of these subgraphs lie on their mutual border.
The local polytope relaxation (LP) of (1) was proposed and analyzed in [4] (see also the recent review [2]). An alternative view on the same relaxation was proposed in [5]. This view appeared to be very close to the idea of the Lagrangian or dual decomposition technique (see [6] for applications to [1]). This idea stimulated development of efficient solvers for convex relaxations of (1). Scalable solvers for the LP relaxation became a hot topic in recent years [7–14]. The algorithms however, which guarantee attainment of the optimum of the convex relaxation at least theoretically, are quite slow in practice, see e.g. comparisons in [11, 15]. Remarkably, the fastest scalable algorithms for convex relaxations are based on coordinate descent: the diffusion algorithm [2] known from the seventies and especially its dual decomposition based variant TRW-S [16]. There are other closely related methods [17, 18] based on the same principle. Although these algorithms do not guarantee attainment of the optimum, they converge [19] to points fulfilling a condition known as arc consistency [2] or weak tree agreement [16]. We show in Section 5 that this condition plays a significant role for our approach. It is a common observation that in the case of sparse graphs and/or strong evidence of the unary terms \(\theta_v\), \(v \in V_G\), the approximate solutions delivered by such solvers are quite good from the practical viewpoint. The belief, that these solutions are close to optimal ones is evidenced by numerical bounds, which these solvers provide as a byproduct.

The techniques used in combinatorial solvers specialized to problem (1) include most of the classical tools: cutting plane, combinatorial search and branch-and-bound methods were adapted to the problem (1). The ideas of the cutting plane method form the basis for tightening the LP relaxation within the dual decomposition framework (see the recent review [20] and references therein) and for finding an exact solution for Potts models [21], which is a special class of problem (1). Combinatorial search methods with dynamic programming based heuristics were successfully applied.
to problems defined on dense and fully connected but small graphs [22]. The specialized branch-and-bound solvers [23][24] also use convex (mostly LP) relaxations and/or a dynamic programming technique to produce bounds in the course of the combinatorial search [25]. However the reported applicability of most combinatorial solvers nowadays is limited to small graphs. Specialized solvers like [21] scale much better, but are focused on a certain narrow class of problems.

The goal of this work is to employ the fact, that local polytope solvers provide good approximate solutions and to restrict computational efforts of combinatorial solvers to a relatively small, and hence tractable part of the initial problem.

**Contribution.** We propose a novel method for obtaining a globally optimal solution of the energy minimization problem (1) for sparse graphs and demonstrate its performance on a series of large-scale benchmark datasets. We were able to

\[ \text{satisfy} \]

- solve previously unsolved large-scale problems of several different types, and
- attain optimal solutions of hard instances of Potts models *an order of magnitude* faster than specialized state of the art algorithms [21].

For an evaluation of our method we use datasets from the very recent benchmark [15].

**Paper structure.** In Section 2 we provide the definitions for the local polytope relaxation and arc consistency. Section 3 is devoted to the specification of our algorithm. In Sections 4 and 5 we provide results of the experimental evaluation and conclusions.

## 2 Preliminaries

**Notation.** A vector \( x \) with coordinates \( x_v, \ v \in \mathcal{V}_G \), will be called *labels* and its coordinates \( x_v \in \mathcal{L}_v \) – *labels*. The notation \( x|_\mathcal{W} \), \( \mathcal{W} \subset \mathcal{V}_G \) stands for the restriction of \( x \) to the subset \( \mathcal{W} \), i.e. for the subvector \((x_v, \ v \in \mathcal{W})\). To shorten notation we will sometimes write \( x_{uv} \in \mathcal{L}_{uv} \) in place of \((x_u, x_v) \in \mathcal{L}_u \times \mathcal{L}_v \) for \((u, v) \in \mathcal{E}_G\). Let also \( \mathcal{W}(v) \), \( v \in \mathcal{V}_G \), denote the set of neighbors of node \( v \), that is the set \{ \( u \in \mathcal{V}_G : uv \in \mathcal{E}_G \)\}. The notations \( \mathcal{N} \) and \( \mathcal{N}(v) \) will be called the *local polytope* commonly used for discrete graphical models [3]. It is well-known that the local polytope constitutes an outer bound (relaxation) of the convex hull of all indicator vectors of labelings (marginal polytope; cf. [3]).

**LP relaxation.** The *local polytope* relaxation of (1) reads (see e.g. [2])

\[
\begin{align*}
\min_{\mu \geq 0} \quad & \sum_{v \in \mathcal{V}_G} \sum_{x_u \in \mathcal{L}_u} \theta_v(x_v) \mu_v(x_v) + \sum_{uv \in \mathcal{E}_G} \sum_{x_u, x_v \in \mathcal{L}_{uv}} \theta_{uv}(x_u, x_v) \mu_{uv}(x_u, x_v) \\
\text{s.t.} \quad & \sum_{x_u \in \mathcal{V}_G} \mu_u(x_u) = 1, \ v \in \mathcal{V}_G \\
& \sum_{x_v \in \mathcal{V}_G} \mu_{uv}(x_u, x_v) = \mu_u(x_u), \ x_u \in \mathcal{L}_u, \ uv \in \mathcal{E}_G \\
& \sum_{x_v \in \mathcal{V}_G} \mu_{uv}(x_u, x_v) = \mu_v(x_v), \ x_v \in \mathcal{L}_v, \ uv \in \mathcal{E}_G.
\end{align*}
\]

(2)

This formulation is based on the overcomplete representation of indicator vectors \( \mu \) constrained to the *local polytope* commonly used for discrete graphical models [3]. It is well-known that the local polytope constitutes an outer bound (relaxation) of the convex hull of all indicator vectors of labelings (marginal polytope; cf. [3]).

The Lagrange dual of (2) reads

\[
\begin{align*}
\max_{\phi, \gamma} \quad & \sum_{v \in \mathcal{V}_G} \gamma_v + \sum_{uv \in \mathcal{E}_G} \gamma_{uv} \\
\text{s.t.} \quad & \gamma_v \leq \theta_v(x_v) := \theta_v(x_v) - \sum_{u \in \mathcal{W}(v)} \phi_{v,u}(x_v), \ v \in \mathcal{V}_G, \ x_v \in \mathcal{L}_v, \\
& \gamma_{uv} \leq \theta_{uv}(x_u, x_v) := \theta_{uv}(x_u, x_v) + \phi_{v,u}(x_v) + \phi_{u,v}(x_u), \ uv \in \mathcal{E}_G, \ (x_u, x_v) \in \mathcal{L}_{uv}.
\end{align*}
\]

(3)

In the constraints of (3) we introduced the reparametrized potentials \( \theta_v, \Phi \). One can see, that for any values of the dual variables \( \phi \) the reparametrized energy \( E_{\Phi, G}(x) \) is equal to the non-reparametrized one \( E_{G,G}(x) \) for any labeling \( x \in \mathcal{L}_G \). The objective function of the dual problem is equal to \( D(\phi) := \sum_{v \in \mathcal{V}_G} \theta_v(x_v) + \sum_{uv \in \mathcal{E}_G} \theta_{uv}(x_u, x_v) \), where \( x_v \in \arg \min_{x_v \in \mathcal{L}_v} x_v\). A *reparametrization*, that is reparametrized potentials \( \theta_v \), will be called *optimal*, if the corresponding \( \phi \) is the solution of the dual problem (3). In general neither the optimal \( \phi \) is unique nor the optimal reparametrization.
\textbf{Definition 1 (Strict arc consistency).} We will call the node \( v \in \mathcal{V}_G \) strictly arc consistent w.r.t. potentials \( \theta \) if there exist labels \( x'_v \in \mathcal{X}_v \) and \( x'_u \in \mathcal{X}_u \) for all \( u \in \mathbf{n}(v) \), such that \( \theta_v(x'_v) < \theta_v(x_v) \) for all \( x_v \in \mathcal{X}_v \setminus \{x'_v\} \) and \( \theta_{vu}(x'_v, x_u) < \theta_{vu}(x_v, x_u) \) for all \( (v, u) \in \mathcal{E}_G \setminus \{(x'_v, x'_u)\} \). The label \( x'_v \) will be called locally optimal.

If all nodes \( v \in \mathcal{V}_G \) are strictly arc consistent w.r.t. the potentials \( \tilde{\theta} \), the dual objective value \( D(\phi) \) becomes equal to the energy

\[
D(\phi) = E_{G, \tilde{\theta}}(x') = E_{G, \theta}(x')
\]

of the labeling \( x' \) constructed by the corresponding locally optimal labels. From duality it follows, that \( D(\phi) \) is a lower bound for energies of all labelings \( E_{G, \theta}(x) \), \( x \in \mathcal{X}_G \). Hence attainment of equality (4) shows that (i) \( \phi \) is the solution of the dual problem (3) and (ii) \( x' \) is the solution of the non-relaxed problem (1). In this work we provide an algorithm, which is able to prove this optimality or discard it. The algorithm applies combinatorial optimization techniques only to the arc inconsistent part of the model, which is often much smaller than the whole model in applications.

\textbf{Remark 1.} Efficient dual decomposition based algorithms optimize dual functions, which differ from \( E_{G, \theta} \) (see e.g. [15, Sec. 2.2]), but are equivalent to it in the sense of equal optimal values. Getting reparametrizations \( \tilde{\theta} \) is less straightforward in these cases, but can be efficiently computed (see e.g. [16] Sec. 2.2).

\section{Algorithm description}

The graph \( \mathcal{A} = (\mathcal{V}_A, \mathcal{E}_A) \) will be called an (induced) subgraph of the graph \( \mathcal{G} = (\mathcal{V}_G, \mathcal{E}_G) \), if \( \mathcal{V}_A \subset \mathcal{V}_G \) and \( \mathcal{E}_A = \{uv \in \mathcal{E}_G : u, v \in \mathcal{V}_A\} \). The graph \( \mathcal{G} \) will be called supergraph of \( \mathcal{A} \). The subgraph \( \partial \mathcal{A} \) induced by a set of nodes \( \partial \mathcal{V}_A \) of the graph \( \mathcal{A} \), which are connected to \( \mathcal{V}_G \setminus \mathcal{V}_A \), is called its boundary w.r.t. \( \mathcal{G} \), i.e. \( \partial \mathcal{V}_A = \{v \in \mathcal{V}_A \exists uv \in \mathcal{E}_G : u \in \mathcal{V}_G \setminus \mathcal{V}_A\} \). The complement \( \mathcal{B} \) to \( \mathcal{A} \setminus \partial \mathcal{A} \), given by \( \mathcal{V}_B = \{v \in \mathcal{V}_G : v \in \partial \mathcal{A} \cup (\mathcal{V}_G \setminus \mathcal{V}_A)\}, \mathcal{E}_B = \{uv \in \mathcal{E}_G : u, v \in \mathcal{V}_B\}, \) is called boundary complement to \( \mathcal{A} \) w.r.t. the graph \( \mathcal{G} \). Let \( \mathcal{A} \) be a subgraph of \( \mathcal{G} \) and potentials \( \theta_v, \; v \in \mathcal{V}_G \), and \( \theta_{uv} \in \mathcal{E}_G \) be associated with nodes and edges of \( \mathcal{G} \) respectively. We assume, that \( \theta_v, \; v \in \mathcal{V}_A \), and \( \theta_{uv} \in \mathcal{E}_A \) are associated with the subgraph \( \mathcal{A} \). Hence we consider the energy function \( E_{A, \theta} \) to be defined on \( \mathcal{A} \) together with an optimal labeling on \( \mathcal{A} \), which is the one that minimizes \( E_{A, \theta} \).

The following theorem formulates conditions necessary to produce an optimal labeling \( x^* \) on the subgraph \( \mathcal{G} \) from the optimal labelings on its mutually boundary complement subgraphs \( \mathcal{A} \) and \( \mathcal{B} \).

\textbf{Theorem 1.} Let \( \mathcal{A} \) be a subgraph of \( \mathcal{G} \) and \( \mathcal{B} \) be its boundary complement w.r.t. \( \mathcal{A} \). Let \( x_A^* \) and \( x_B^* \) be labelings minimizing \( E_{A, \theta} \) and \( E_{B, \theta} \) respectively and let all nodes \( v \in \mathcal{V}_A \) be strictly arc consistent w.r.t. potentials \( \theta \). Then from

\[
x_{A, v}^* = x_{B, v}^*, \; \text{for all} \; v \in \partial \mathcal{A}
\]

follows that the labeling \( x^* \) with coordinates \( x_v^* = \begin{cases} x_{A, v}^* & v \in \mathcal{A} \\ x_{B, v}^* & v \in \mathcal{B} \setminus \mathcal{A} \end{cases} \), \( v \in \mathcal{V}_G \), is optimal on \( \mathcal{G} \).

\textbf{Proof.} Let \( \theta \) denote potentials of the problem. Let us define other potentials \( \theta' \) as

\[
\theta'_w(x_w) := \begin{cases} 0, & w \in \partial \mathcal{V}_A \cup \partial \mathcal{E}_A \\ \theta_w(x_w), & w \notin \partial \mathcal{V}_A \cup \partial \mathcal{E}_A \end{cases} \quad \text{then} \quad E_{G, \theta}(x) = E_{A, \theta'}(x|\mathcal{A}) + E_{B, \theta}(x|\mathcal{B}).
\]

From strict
Let all nodes of a graph. Following theorem states this property formally. A solver, keeping the results as if the subproblem over reparametrization allows also to perform only a single run of the LP Single run of LP solver. Obtained for the subgraph B ̃ we consider the reparametrized potentials optimal of labeling It is quite unlikely, that the optimal boundary 3.1 Remarks on Algorithm 1 holds we have found the optimal labeling according to Theorem 1. Otherwise we remove the nodes x plugged in place of x ̇ optimal labeling. We define ̃θ potentials θ, obtained at the LP presolve step of the algorithm. Step (2). We assign to the set V A the nodes of the graph G, which satisfy the strict arc consistency condition. The optimal labeling on A can be trivially computed from the reparametrized unary potentials ̂θ v by x ̂θ v := arg min x v , ̂θ v (x v ), v ∈ A. Step (3). We define B as the boundary complement to A w.r.t. the master graph G and find an optimal labeling x ̃v on the subgraph B with a combinatorial solver. If the boundary condition 5 holds we have found the optimal labeling according to Theorem 1. Otherwise we remove the nodes where this condition fails from A and repeat the whole step until either 5 holds or B = G.

3.1 Remarks on Algorithm 1

Encouraging boundary consistency condition. It is quite unlikely, that the optimal boundary labeling x ̃v obtained based only on the subgraph A coincides with the boundary labeling x ̃B obtained for the subgraph B. To satisfy this condition the unary potentials should be quite strong on the border. In other words, they should be at least strictly arc consistent. Indeed they are so, since we consider the reparametrized potentials ̂θ v, obtained at the LP presolve step of the algorithm.

Single run of LP solver. Reparametrization allows also to perform only a single run of the LP solver, keeping the results as if the subproblem over A has been solved at each iteration. The following theorem states this property formally.

Theorem 2. Let all nodes of a graph A be strictly arc consistent w.r.t. potentials ̂θ v, x be the optimum of E A,̂θ v and A' be a subgraph of A. Then x|A' optimizes E A',̂θ v .

Proof. The proof follows directly from Definition 1. Equation 4 holds for the labeling x|A' plugged in place of x' and graph A' in place of G. Hence x|A' provides a minimum of E A',̂θ v .

Presolving B for combinatorial solver. Many combinatorial solvers use linear programming relaxations as a presolving step. Reparametrization of the subproblem over the subgraph B plays the role of such a presolver, since the optimal reparametrization corresponds to the solution of the dual problem and makes solving the primal one easier.

Connected components analysis. It is often the case that the subgraph B consists of several connected components. We apply the combinatorial solver to each of them independently.
| Dataset | $|V_G|$ | $|X_v|$ | Step (1) LP (TRWS) | Step (3) ILP (CPLEX) | $|B|$ |
|---------|--------|--------|-----------------|-----------------|------|
| tsukuba | 110592 | 16     | 250 186 369537 | 24 36 369218 | 130 656 |
| venus   | 166222 | 20     | 2000 3083 3048296 | 10 69 3048043 | 66 233 |
| teddy   | 168750 | 60     | 10000 14763 1345214 | 1 - - | 2062 - |
| family  | 425632 | 5      | 10000 20156 184825 | 18 2 184813 | 11 109 |
| pano    | 514080 | 7      | 10000 34092 169224 | 1 - - | 24474 - |

Table 1: Results on Middlebury datasets. The column Dataset contains the dataset name, numbers $|V_G|$ of nodes and $|X_v|$ of labels. Columns Step (1) and Step (3) contain number of iterations, time and attained energy at steps (1) and (3) of Algorithm 1 corresponding to solving the LP relaxation and use of a combinatorial solver respectively. The column $|B|$ presents starting and final sizes of the ”combinatorial” subgraph $B$. Dash ”-” stands for failure of CPLEX, due to the size of the combinatorial subproblem.

Subgraph $B$ growing strategy. One can consider different strategies for increasing the subgraph $B$, if the boundary condition (5) does not hold. Our greedy strategy is just one possible option.

Optimality of reparametrization. As one can see, the reparametrization plays a significant role for our algorithm: it (i) is required for Theorem 1 to hold; (ii) serves as a criterion for the initial splitting of $G$ into $A$ and $B$; (iii) makes the local potentials on the border $\partial A$ stronger; (iv) allows to avoid multiple runs of the LP solver, when the subgraph $A$ shrinks; (v) can speed-up some combinatorial solvers by serving as a presolve result. However, there is no real reason to search for an optimal reparametrization: all its mentioned functionality remains valid also if it is non-optimal. Of course, one pays a certain price for the non-optimality: (i) the initial subgraph $B$ becomes larger; (ii) the local potentials – weaker; (iii) the presolve results for the combinatorial solver become less precise. Note that even for non-optimal reparametrizations Theorem 2 holds and we need to run the LP solver only once.

4 Experimental evaluation

We tested our approach on problems from the Middlebury energy minimization benchmark [26] and the recently published discrete energy minimization benchmark [15], which includes the datasets from the first one. We have selected computer vision benchmarks intentionally, because many problems in this area fulfill our requirements: the underlying graph is sparse (typically it has a grid structure) and the LP relaxation delivers good practical results.

Since our experiments serve mainly as proof of concept we used general, though not always the most efficient solvers: TRW-S [16] as the LP-solver and CPLEX [27] as the combinatorial one within the OpenGM framework [28]. Unfortunately the original version of TRW-S does not provide information about strict arc consistency and does not output a reparametrization. Therefore we used our own implementation in the experiments. Depending on the type of the pairwise factors (Potts, truncated $\ell_2$ or $\ell_1$-norm) we found our implementation up to an order of magnitude slower than the freely available code of V. Kolmogorov. This fact suggests that the provided processing time can be significantly improved in more efficient future implementations.

In the first round of our experiments we considered problems (i.e. graphical models with the specified unary and pairwise factors) of the Middlebury MRF benchmark, most of which remained unsolved, to the best of our knowledge.

MRF stereo dataset consists of 3 models: tsukuba, venus and teddy. Since the optimal integral solution of tsukuba was recently obtained by LP-solvers [11,13], we used this dataset to show how our approach performs for clearly non-optimal reparametrizations. For this we run TRW-S for 250 iterations only. The size of the subgraph $B$ grew from 130 to 656 nodes out of more than 100000 nodes of the original problem (see Table 1). On venus we obtained an optimal labeling after 10 iterations of our algorithm. During these iterations the size of the set $B$ grew from 66 to 233 nodes, which is only 0.14% of the original problem size. The dataset teddy remains unsolved: though
Table 2: Exemplary Potts model comparison. Datasets taken from the Color segmentation (N8) set. Column $E_{\theta}(x^*)$ shows the optimal energy value, columns Step (1) LP and Step (3) ILP contain number of iterations and time spent at the steps (1) and (3) of Algorithm 1 corresponding to solving the LP relaxation and use of a combinatorial solver respectively. The column MCA stands for the time of the multiway-cut solver reported in [21]. The MPLP [17] column provides number of iterations and time of the LP presolve and the time of the tightening cutting plane phase (ILP).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$E_{\theta}(x^*)$</th>
<th>Step (1) LP</th>
<th>Step (3) ILP</th>
<th>MCA</th>
<th>MPLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>pfau</td>
<td>24010.44</td>
<td>1000</td>
<td>276</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>palm</td>
<td>12253.75</td>
<td>200</td>
<td>65</td>
<td>17</td>
<td>93</td>
</tr>
<tr>
<td>clownfish</td>
<td>14794.18</td>
<td>100</td>
<td>32</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>crops</td>
<td>11853.12</td>
<td>100</td>
<td>32</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>strawberry</td>
<td>11766.34</td>
<td>100</td>
<td>29</td>
<td>8</td>
<td>31</td>
</tr>
</tbody>
</table>

Figure 2: Results for the pfau-instance from [15]. Gray pixels in (b) and (c) mark nodes that need to be labeled by the combinatorial solver. Our approach (c) leads to much smaller combinatorial problem instances than Kovtun’s method (b) used in [29]. While Kovtun’s method gets partial optimality for 5% of the nodes only, our approach requires to solve only tiny problems by a combinatorial solver.

Potts models. Our approach appeared to be especially efficient for Potts models. We tested it on the following datasets from the benchmark [15]: Color segmentation (N4), Color segmentation (N8), Color segmentation, Brain and managed to solve all 26 problem instances to optimality. Solving Potts models to optimality is not a big issue anymore due to the recent work [21], which related this problems to the multiway-cut problem [31] and adopted a quite efficient solver based on the cutting plane technique. However, we were able to outperform even this specialized solver on hard instances, which we collected in Table 2. There is indeed a simple explanation for this phenomenon: the difficult instances are those, for which the optimal labeling contains many small areas corresponding to different labels, see e.g. Fig 2. This is not very typical for Potts models, where an optimal labeling typically consists of a small number of large segments. Since the number of cutting planes, which have to be processed by the multiway-cut solver, grows with the total length of the segment borders, the overall performance significantly drops on such instances. Our approach is able to correctly label most of the borders when solving the LP relaxation. Since the resulting subgraph $B$, passed to the combinatorial solver, is quite small, the corresponding subproblems appear
easy to solve even for a general-purpose solver like CPLEX. Indeed, we expect an increase in the overall performance of our method if the multway-cut solver would be used in place of CPLEX.

For Potts models there exist methods \[29,32\] providing part of an optimal solution, known as partial optimality. Often they allow to drastically simplify the problem so that it can be solved to global optimality on the remaining variables very fast, see \[30\]. However for hard instances like pfau these methods can label only a small fraction of graph nodes persistently, hence combinatorial solvers cannot solve the rest, or require a lot of time. Our method does not provide partially optimal variables: if it cannot solve the whole problem no node can be labelled as optimal at all. On the upside the subgraph \(B\) which is given to a combinatorial solver is typically much smaller, see Fig. 2.

For comparison we tested the MPLP solver \[17\], which is based on coordinate descent LP iterations and tightens the LP relaxation with the cutting plane approach described in \[33\]. We used its publicly available code \[34\]. However this solver did not managed to solve any of the considered difficult problems (marked as unsolved in the OpenGM Benchmark \[15\]), such as color-seg-n8/pfau, mrf_stereo/{venus, teddy}, mrf_photomontage/{family, pano}. For easier instances of the Potts model, we found our solver an order of magnitude faster than MPLP (see Table 2 for the exemplary comparison), though we tried different numbers of LP presolve iterations to speed up the MPLP.

**Summary.** Our experiments show that our method used even with quite general and not always the most efficient solvers like TRW-S and CPLEX allows to (i) find globally optimal solutions of large scale problem instances, which were previously unsolvable; (ii) solve hard instances of Potts models an order of magnitude faster than with a modern specialized combinatorial multiway-cut method; (iii) overcome the cutting-plane based MPLP method on the tested datasets.

## 5 Conclusions and future work

The method proposed in this paper provides a novel way of combining convex and combinatorial algorithms to solve large scale optimization problems to a global optimum. It does an efficient extraction of the subgraph, where the LP relaxation is not tight and combinatorial algorithms have to be applied. Since this subgraph often corresponds to only a tiny fraction of the initial problem, the combinatorial search becomes feasible. The method is very generic: any linear programming and combinatorial solvers can be used to carry out the respective steps of Algorithm 1. It is particularly efficient for sparse graphs and when the LP relaxation is almost tight.

In the future we plan to generalize the method to higher order models, tighter convex relaxations for the convex part of our solver and apply alternative and specialized solvers both for the convex and the combinatorial parts of our approach.

**Acknowledgement.** This work has been supported by the German Research Foundation (DFG) within the program Spatio-/Temporal Graphical Models and Applications in Image Analysis, grant GRK 1653. Authors thank A. Shekhovtsov, B. Flach, T. Werner, K. Antoniuk and V. Franc from the Center for Machine Perception of the Czech Technical University in Prague for fruitful discussions.

**References**


