A Dual Ascent Framework for Lagrangean Decomposition of Combinatorial Problems

Paul Swoboda[1], Jan Kuske[2], Bogdan Savchynskyy[3]
[1]IST Austria, [2]Universität Heidelberg, [3]TU Dresden
pswoboda@ist.ac.at

Abstract

We propose a general dual ascent framework for Lagrangean decomposition of combinatorial problems. Although methods of this type have shown their efficiency for a number of problems, so far there was no general algorithm applicable to multiple problem types. In this work, we propose such a general algorithm. It depends on several parameters, which can be used to optimize its performance in each particular setting. We demonstrate efficacy of our method on graph matching and multicut problems, where it outperforms state-of-the-art solvers including those based on subgradient optimization and off-the-shelf linear programming solvers.

1. Introduction

Computer vision and machine learning give rise to a number of powerful computational models. It is typical that inference in these models reduces to non-trivial combinatorial optimization problems. For some of the models, such as conditional random fields (CRF), powerful specialized solvers like [48, 49, 12, 52] were developed. In general, however, one has to resort to off-the-shelf integer linear program (ILP) solvers like CPLEX [2] or Gurobi [36]. Although these solvers have made a tremendous progress in the past decade, the size of the problems they can tackle still remains a limiting factor for many potential applications, as the running time scales super-linearly in the problem size. The goal of this work is to partially fill this gap between practical requirements and existing computational methods.

It is an old observation that many important optimization ILPs can be efficiently decomposed into easily solvable combinatorial sub-problems [32]. The convex relaxation, which consists of these sub-problems coupled by linear constraints is known as Lagrangean or dual decomposition [31, 50]. Although this technique can be efficiently used in various scenarios to find approximate solutions of combinatorial problems, it has a major drawback: In the most general setting only slow (sub)gradient-based techniques [51, 57, 50, 42, 61] can be used for optimization of the corresponding convex relaxation.

In the area of conditional random fields, however, it is well-known [41] that message passing or dual (block-coordinate) ascent algorithms (like e.g. TRW-S [48]) significantly outperform (sub)gradient-based methods. Similar observations were made much earlier in [59] for a constrained shortest path problem.

Although dual ascent algorithms were proposed for a number of combinatorial problems (see the related work overview below), there is no general framework, which would (i) give a generalized view on the properties of such algorithms and more importantly (ii) provide tools to easily construct such algorithms for new problems. Our work provides such a framework.

Related Work Dual ascent algorithms optimize a dual problem and guarantee monotonous improvement (non-deterioration) of the dual objective. The most famous examples in computer vision are block-coordinate ascent (known also as *message passing*) algorithms like TRW-S [48] or MPLP [28] for maximum a posteriori inference in conditional random fields [41].

To the best of our knowledge the first dual ascent algorithm addressing integer linear programs belongs to Bilde and Krarup [11] (the corresponding technical report in Danish appeared 1967). In that work an uncapacitated facility location problem was addressed. A similar problem (simple plant location) was addressed with an algorithm of the same class in [30]. In 1980 Fisher and Hochbaum [22] constructed a dual ascent-based algorithm for a problem of database location in computer networks, which was used to optimize the topology of Arpanet [1], predecessor of Internet. The generalized linear assignment problem was addressed by the same type of algorithms in [23]. The Authors considered a Lagrangean decomposition of this problem into multiple knapsack problems, which were solved in each iteration of the method. An improved version of this algorithm was proposed in [34]. Efficient dual ascent based solvers were also proposed for the min-cost flow in [25], for the set covering and the set partitioning problems in [24] and the resource-constrained minimum weighted arborescence problem in [35]. The work [33] describes basic principles for constructing dual ascent algorithms. Although the authors provide several examples, they do not go beyond that and stick to the claim that these methods are structure dependent and problem specific.

The work [18] suggests to use the max-product belief propagation [73] to decomposable optimization problems. However, their algorithm is neither monotone nor even convergent in general.

In computer vision, dual block coordinate ascent algorithms for Lagrangean decomposition of combinatorial problems were proposed for multiple targets tracking [8], graph matching (quadratic assignment) problem [78] and inference in conditional random fields [48, 49, 28, 74, 75, 62, 37, 56, 72, 39]. From the latter, the TRW-S algorithm [48] is among the most efficient ones for pairwise conditional random fields according to [41]. The SRMP algorithm [49] generalizes TRW-S to conditional random fields of arbitrary order. In a certain sense, our framework can be seen as a generalization of SRMP to a broad class of combinatorial problems.

Contribution. We propose a new dual ascent based computational framework for combinatorial optimization. To this end we:

(i) Define the class of problems, called *integer-relaxed pairwise-separable linear programs* (IRPS-LP), our framework can be used for. Our definition captures Lagrangean decompositions of many known discrete optimization problems (Section 2).

(ii) Give a general monotonically convergent message-passing algorithm for solving IRPS-LP, which in particular subsumes several known solvers for conditional random fields (Section 4).

(iii) Give a characterization of the fixed points of our algorithm, which subsumes such well-known fixed point characterizations as *weak tree agreement* [48] and *arc-consistency* [74] (Section 5).

We demonstrate efficiency of our method by outperforming state-of-the-art solvers for two famous special cases of IRPS-LP, which are widely used in computer vision: the multicut and the graph matching problems. (Section 6).

A C++-framework containing the above mentioned solvers and the datasets used in experiments can be obtained at `https://github.com/pawelswoboda/LP_MP`.

We give all proofs in the supplementary material.

Notation. Undirected graphs will be denoted by $G = (V, E)$, where V is a finite *node set* and $E \subseteq \binom{V}{2}$ is *the edge set*. The set of neighboring nodes of $v \in V$ w.r.t. graph G is denoted by $\mathcal{N}_G(v) := \{u : uv \in E\}$. The convex hull of a set $X \subset \mathbb{R}^n$ is denoted by $\operatorname{conv}(X)$. Disjoint union is denoted by $\dot\cup$.

2. Integer-Relaxed Pairwise-Separable Linear Programs (IRPS-LP)

Combinatorial problems of the form $\min_{x \in X} \theta(x)$, where $X \subseteq \{0,1\}^n$ are binary vectors, often have a decomposable representation as $\min_{\substack{x_i \in X_i \\ i=1,\dots,k}} \sum_{i=1}^{k} \langle \theta_i, x_i \rangle$ for $X_i \subseteq \{0,1\}^{d_i}$ being sets of binary vectors, typically corresponding to subsets of the coordinates of X. This decomposed problem is equivalent to the original one under a set of linear constraints $A_{(i,j)} x_i = A_{(j,i)} x_j$, which guarantee the mutual consistency of the considered components. By replacing X_i by its convex hull $\operatorname{conv}(X_i)$ we switching to real-valued vectors from binary ones and obtain a *convex relaxation* of the problem. It reads:

$$\min_{\mu \in \Lambda_{\mathbb{G}}} \sum_{i=1}^{k} \langle \theta_i, \mu_i \rangle \,, \text{ where } \Lambda_{\mathbb{G}} \text{ is defined as} \tag{1}$$

$$\Lambda_{\mathbb{G}} := \left\{ (\mu_1 \dots \mu_k) \;\middle|\; \begin{array}{ll} \mu_i \in \operatorname{conv}(X_i) & i \in \mathbb{F} \\ A_{(i,j)} \mu_i = A_{(j,i)} \mu_j & \forall ij \in \mathbb{E} \end{array} \right\}. \tag{2}$$

Here $\mathbb{F} := \{1, \dots, k\}$ are called *factors* of the decomposition and $\mathbb{E} \subseteq \binom{\mathbb{F}}{2}$ correspond to the *coupling constraints*. The undirected graph $\mathbb{G} = (\mathbb{F}, \mathbb{E})$ is called *factor graph*. We will use variable names μ whenever we want to emphasize $\mu_i \in \operatorname{conv}(X_i)$ and x whenever $x_i \in X_i$, $i \in \mathbb{F}$.

Definition 1 (IRPS-LP)**.** *Assume that for each edge $ij \in \mathbb{E}$ the matrices of the coupling constraints $A_{(i,j)}, A_{(j,i)}$ are such that $A_{(i,j)} \in \{0,1\}^{K \times d_i}$ and $A_{(i,j)} x_i \in \{0,1\}^K$ $\forall x_i \in X_i$ for some $K \in \mathbb{N}$, analogously for $A_{(j,i)}$. The problem $\min_{\mu \in \Lambda_{\mathbb{G}}} \sum_{i \in \mathbb{F}} \langle \theta_i, \mu_i \rangle$ is called an* **I**nteger-**R**elaxed **P**airwise-**S**eparable **L**inear **P**rogram*, abbreviated by* IRPS-LP.

In the following, we give several examples of IRPS-LP. To distinguish between notation for the factor graph of IRPS-LP, where we stick to bold letters (such as $\mathbb{G}$, $\mathbb{F}$, $\mathbb{E}$) we will use the straight font (such as G, V, E) for the graphs occurring in the examples.

Example 1 (MAP-inference for CRF)**.** A conditional random field is given by a graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, a discrete label space $X = \prod_{u \in \mathsf{V}} X_u$, unary $\theta_u : X_u \to \mathbb{R}$ and pairwise costs $\theta_{uv} : X_u \times X_v \to \mathbb{R}$ for $u \in \mathsf{V}$, $uv \in \mathsf{E}$. We also denote $X_{uv} := X_u \times X_v$. The associated *maximum a posteriori (MAP)-inference problem* reads

$$\min_{x \in X} \sum\nolimits_{u \in \mathsf{V}} \theta_u(x_u) + \sum\nolimits_{uv \in \mathsf{E}} \theta_{uv}(x_{uv}) \,, \tag{3}$$

where x_u and x_{uv} denote the components corresponding to node $u \in \mathsf{V}$ and edge $uv \in \mathsf{E}$ respectively. The well-known local polytope relaxation [74] can be seen as an IRPS-LP by setting $\mathbb{F} = \mathsf{V} \cup \mathsf{E}$, that is associating to each node $v \in \mathsf{V}$ *and* each edge $uv \in \mathsf{E}$ a factor, and introducing two coupling constraints for each edge of the graphical model, i.e. $\mathbb{E} = \{\{u, uv\}, \{v, uv\} : uv \in \mathsf{E}\}$. For the sake of notation we will assume that each label $s \in X_u$ is associated a unit vector $(0, \dots, 0, \underbrace{1}_{s}, 0 \dots, 0)$ with dimensionality equal to the total number of labels $|X_u|$ and 1 on the s-th position. Therefore, the notation $\operatorname{conv}(X_u)$ makes sense as a convex hull of all such vectors. After denoting an N-dimensional *simplex* as $\Delta_N := \{\mu \in \mathbb{R}^N_+ \colon \sum_{i=1}^N \mu_i = 1\}$ the resulting relaxation reads

$$\min_{\mu \in \mathsf{L}_\mathsf{G}} \quad \langle \theta, \mu \rangle := \sum_{u \in \mathsf{V}} \langle \theta_u, \mu_u \rangle + \sum_{uv \in \mathsf{E}} \langle \theta_{uv}, \mu_{uv} \rangle \tag{4}$$

in the overcomplete representation [71] and L_G is defined as

$$\begin{array}{ll} \underline{\mu_u \in \operatorname{conv}(X_u) :} & \mu_u \in \Delta_{|X_u|}, u \in \mathsf{V} \\ \underline{\mu_{uv} \in \operatorname{conv}(X_{uv}) :} & \mu_{uv} \in \Delta_{|X_{uv}|}, uv \in \mathsf{E} \\ \underline{A_{(uv,u)}\mu_{uv} = A_{(u,uv)}\mu_u :} & \sum\limits_{x_v \in X_v} \mu_{uv}(x_u, x_v) = \mu_u(x_u), \\ & uv \in \mathsf{E}, (x_u, x_v) \in X_{uv}, \\ & u \in uv, x_u \in X_u\,. \end{array} \tag{5}$$

Here $\mu_u(x_u)$ and $\mu_{uv}(x_u, x_v)$ denote those coordinates of vectors μ_u and μ_{uv}, which correspond to the label x_u and the pair of labels (x_u, x_v) respectively.

Example 2 (Graph Matching)**.** The graph matching problem, also known as *quadratic assignment* [13] or *feature matching*, can be seen as a MAP-inference problem for CRFs (as in Example 1) equipped with additional constraints: The label set of G belongs to a *universe* $\mathcal{L}$, i.e. $X_u \subseteq \mathcal{L}\ \forall u \in \mathsf{V}$ and each label $s \in \mathcal{L}$ can be assigned *at most once*. The overall problem reads

$$\min_x \sum_{u \in \mathsf{V}} \theta_u(x_u) + \sum_{uv \in \mathsf{E}} \theta_{uv}(x_u, x_v) \text{ s.t. } x_u \neq x_v \forall u \neq v\,. \tag{6}$$

Graph matching is a key step in many computer vision applications, among them tracking and image registration, whose aim is to find a one-to-one correspondence between image points. For this reason, a large number of solvers have been proposed in the computer vision community [18, 76, 78, 70, 53, 69, 63, 29, 79, 38, 54, 16]. Among them two recent methods [70, 78] based on Lagrangean decomposition show superior performance and provide lower bounds for their solutions. The decomposition we describe below, however, differs from those proposed in [70, 78].

Our IRPS-LP representation for graph matching consists of two blocks: (i) the CRF itself (which further decomposes into node- and edge-subproblems with variables $(\mu_u)_{u \in \mathsf{V}}$ and (ii) additional *label-factors* keeping track of nodes assigned the label s. We introduce these label-factors for each label $s \in \mathcal{L}$. The set of possible configurations of this factor $X_s := \{u \in \mathsf{V} : s \in X_u\} \cup \{\#\}$ consists of those nodes $u \in \mathsf{V}$ which can be assigned the label s and an additional dummy node $\#$. The dummy node $\#$ denotes non-assignment of the label s and is necessary, as not every label needs to be taken. As in Example 1, we associate a unit binary vector with each element of the set X_s, and $\operatorname{conv}(X_s)$ denotes the convex hull of such vectors. The set of factors becomes $\mathbb{F} = \mathsf{V} \dot\cup \mathsf{E} \dot\cup \mathcal{L}$, with the set $\mathbb{E} = \{\{u, uv\}, \{v, uv\} : uv \in \mathsf{E}\} \cup \{\{u, l\} : u \in \mathsf{V}, l \in X_u\}$ of the factor-graph edges. The resulting IRPS-LP formulation reads

$$\begin{array}{ll} \min\limits_{\mu, \tilde\mu} \sum\limits_{u \in \mathsf{V}} \langle \theta_u, \mu_u \rangle + \sum\limits_{uv \in \mathsf{E}} \langle \theta_{uv}, \mu_{uv} \rangle + \sum\limits_{s \in \mathcal{L}} \langle \tilde\theta_s, \tilde\mu_s \rangle & \\ \mu \in \mathsf{L}_\mathsf{G} & \\ \tilde\mu_s \in \operatorname{conv}(X_s), & s \in \mathcal{L} \\ \mu_u(s) = \tilde\mu_s(u), & s \in X_u\,. \end{array} \tag{7}$$

Here we introduced (i) auxiliary variables $\tilde\mu_s(u)$ for all variables $\mu_u(s)$ and (ii) auxiliary node costs $\tilde\theta_s \equiv 0\ \forall s \in \mathcal{L}$, which may take other values in course of optimization. Factors associated with the vectors μ_u and μ_{uv} correspond to the nodes and edges of the graph G (node- and edge-factors), as in Example 1 and are coupled in the same way. Additionally, factors associated with the vectors $\tilde\mu_s$ ensure that the label s can be taken at most once. These label-factors are coupled with node-factors (last line in (7)).

Example 3 (Multicut)**.** The multicut problem (also known as correlation clustering) for an undirected weighted graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is to find a partition $(\Pi_1, \dots, \Pi_k)$, $\Pi_i \subseteq \mathsf{V}$, $\mathsf{V} = \dot\cup_{i=1}^k \Pi_i$ of the graph vertexes, such that the total cost of edges connecting different components is minimized. The number k of components is not fixed but is determined by the algorithm. See Fig. 1 for an illustration. Although the problem has numerous applications in computer vision [4, 5, 6, 77] and beyond [7, 60, 14, 15], there is no scalable solver, which could provide optimality bounds. Existing methods are either efficient primal heuristics [66, 58, 27, 19, 20, 9, 10] or combinatorial branch-and-bound/branch-and-cut/column generation algorithms, based on off-the-shelf LP solvers [43, 44, 47, 77]. Move-making algorithms do not provide lower bounds, hence, one cannot judge their solution quality or employ them in branch-and-bound procedures. Off-the-shelf LP solvers on the other hand scale super-linearly, limiting their application in large-scale problems.

Instead of directly optimizing over partitions (which has many symmetries making optimization difficult in a linear programming setting), we follow [17] and formulate the problem in the edge domain. Let θ_e, $e \in \mathsf{E}$ denote the cost of graph edges and let C be the set of all cycles of the graph G. Each edge that belongs to different components is called a *cut edge*. The multicut problem reads

$$\min_{x \in \{0,1\}^{|\mathsf{E}|}} \sum_{e \in \mathsf{E}} \theta_e x_e\,, \quad \text{s.t. } \forall \mathsf{C}\ \forall e' \in \mathsf{C} \colon \sum_{e \in \mathsf{C} \setminus \{e'\}} x_e \geq x_{e'}\,. \tag{8}$$

Here $x_e = 1$ signifies a cut edge and the inequalities force each cycle to have none or at least two cut edges. The formulation (8) has exponentially many constraints. However, it is well-known that it is sufficient to consider only chordless cycles [17] in place of the set C in (8). Moreover, the graph can be triangulated by adding additional edges with zero weights and therefore the set of chordless cycles reduces to edge triples. Such triangulation is refered to as *chordal completion* in the literature [26]. The number of triples is cubic, which is still too large for practical efficiency and therefore violated constraints are typically added to the problem iteratively in a cutting plane manner [43, 44]. To simplify the description, we will ignore this fact below and consider all these cycles at once. Assuming a triangulated graph and redefining C as the set of all chordless cycles (triples) we consider the following IRPS-LP relaxation of the multicut problem [1]:

$$\min_{\mu,\tilde{\mu}} \sum_{e\in\mathsf{E}} \theta_e \mu_e + \sum_{c\in\mathsf{C}} \sum_{e\in c} \tilde{\theta}_{e,c}\tilde{\mu}_{e,c}\,, \quad \text{s.t.} \tag{9}$$

$$\begin{cases} \mu_e \in \operatorname{conv}(\{0,1\}) = [0,1],\ e \in \mathsf{E} \\ \forall c \in \mathsf{C},\ e \in c\colon \\ \tilde{\mu}_c := (\tilde{\mu}_{e,c})_{e\in c} \in \operatorname{conv}(\{0,1\}^3 |\, \forall e' \in\ c\colon \sum\limits_{e\in c\setminus\{e'\}} \tilde{\mu}_{e,c} \geq \tilde{\mu}_{e',c}) \\ \quad \equiv \operatorname{conv}(\{0,0,0\},\{0,1,1\},\{1,0,1\},\{1,1,0\},\{1,1,1\}) \\ \mu_e = \tilde{\mu}_{e,c} \end{cases} \tag{10}$$

For the sake of notation we shortened a feasible set definition $\tilde{\mu} \in \operatorname{conv}(\mu' \in \{0,1\}^n\colon \text{constraints on } \mu')$ to $\tilde{\mu} \in \operatorname{conv}(\{0,1\}^n\colon \text{constraints on } \tilde{\mu})$. Here μ_e is the relaxed (potentially non-integer) variable corresponding to x_e. Variable $\tilde{\mu}_{e,c}$ is a copy of μ_e, which corresponds to the cycle c. Therefore, each μ_e gets as many copies $\tilde{\mu}_{e,c}$, as many chordless cycles c contain the edge e. For each cycle the set of binary vectors satisfying the cycle inequality is considered. For a cycle with 3 edges this set can be written explicitly as in (10). Along with copies of μ_e, $e \in \mathsf{E}$ we copy the corresponding cost θ_e and create auxiliary costs $\tilde{\theta}_{e,c} \equiv 0$ for each cycle c containing the edge e. During optimization, the cost θ_e will be redistributed between θ_e itself and its copies $\tilde{\theta}_{e,c}$, $c \in \mathsf{C}$. The factors of the IRPS-LP are associated with each edge (variable μ_e) and each chordless cycle (variable $\tilde{\mu}_c$). Coupling constraints connect edge-factors with those cycle-factors, which contain the corresponding edge (see the last constraint in (10)). An in-depth discussion of message passing for the multicut problem with tighter relaxations can be found in [67].

3. Dual Problem and Admissible Messages

Since our technique can be seen as a dual ascent, we will not optimize the primal problem (1) directly, but instead maximize its dual lower bound.

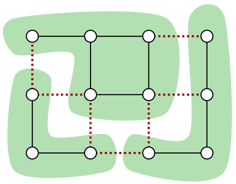

Figure 1. Illustration of Example 3. A multicut of a graph induced by three connected components Π_1, Π_2, Π_3 (green). Red dotted edges indicate cut edges $x_e = 1$.

Dual IRPS-LP The Lagrangean dual to (1) w.r.t. the coupling constraints reads

$$\begin{array}{lll} \max_\phi & D(\phi) := \sum_{i\in\mathbb{F}} \min_{x_i\in X_i} \langle \theta_i^\phi, x_i\rangle & \\ \text{s.t.} & \theta_i^\phi := \theta_i + \sum_{j:ij\in\mathbb{E}} A_{(i,j)}^\top \phi_{(i,j)} & \forall i \in \mathbb{F} \\ & \phi_{(i,j)} = -\phi_{(j,i)} \quad \forall ij \in \mathbb{E} & \end{array} \tag{11}$$

Here $\phi_{(i,j)} \in \mathbb{R}^K$ for $A_{(i,j)} \in \{0,1\}^{K\times d_i}$ for some $K \in \mathbb{N}$. The function $D(\phi)$ is called *lower bound* and is concave in ϕ. The modified primal costs θ^ϕ are called *reparametrizations* of the potentials θ. We have duplicated the dual variables by introducing $\phi_{(i,j)} := -\phi_{(j,i)}$ to symmetrize notation. In practice, only one copy is stored and the other is computed on the fly. Note that in this doubled notation the reparametrized node and edge potentials of the CRF from Example 1 read

$$\begin{array}{l} \theta_u^\phi(x_u) = \theta_u(x_u) + \sum_{v\colon\, uv\in\mathsf{E}} \phi_{u,uv}(x_u) \\ \theta_{uv}^\phi(x_u,x_v) = \theta_{uv}(x_u,x_v) + \phi_{uv,v}(x_v) + \phi_{uv,u}(x_u) \\ \phi_{u,uv} = -\phi_{uv,u} \end{array}$$

It is well-known for CRFs that cost of feasible solutions are invariant under reparametrization. We generalize this to the IRPS-LP-case.

Proposition 1. *$\sum_{i\in\mathbb{F}} \langle\theta_i, \mu_i\rangle = \sum_{i\in\mathbb{F}} \langle\theta_i^\phi, \mu_i\rangle$, whenever $\mu_1, \ldots, \mu_k$ obey the coupling constraints.*

Admissible Messages While Proposition 1 guarantees that the primal problem is invariant under reparametrizations, the dual lower bound $D(\phi)$ is not. Our goal is to find ϕ such that $D(\phi)$ is maximal. By linear programming duality, $D(\phi)$ will then be equal to the optimal value of the primal (1).

First we will consider an elementary step of our future algorithm and show that it is non-decreasing in the dual objective. This property will ensure the monotonicity of the whole algorithm. Let θ^ϕ be any reparametrization of the problem and $D(\phi)$ be the corresponding dual value. Let us consider changing the reparametrization of a factor i by a vector Δ with the only non-zero components $\Delta_{(i,j)}$ and $\Delta_{(j,i)}$. This will change reparametrization of the coupled factors j (such that $ij \in \mathbb{E}$) due to $\Delta_{(i,j)} = -\Delta_{(j,i)}$. The lemma below states properties of $\Delta_{(i,j)}$ which are sufficient to guarantee improvement of the corresponding dual value $D(\phi + \Delta)$:

[1] One can show that this relaxation coincides with the standard LP relaxation for the multicut problem [17]

Lemma 1 (Monotonicity Condition)**.** *Let $ij \in \mathbb{E}$ be a pair of factors related by the coupling constraints and $\phi_{(i,j)}$ be a corresponding dual vector. Let $x_i^* \in \underset{x_i \in X_i}{\operatorname{argmin}} \langle \theta_i^\phi, x_i \rangle$ and $\Delta_{(i,j)}$ satisfy*

$$\Delta_{(i,j)}(s) \begin{cases} \geq 0, & \nu(s) = 1 \\ \leq 0, & \nu(s) = 0 \end{cases}, \ \textit{where } \nu := A_{(i,j)} x_i^* \,. \qquad (12)$$

Then $x_i^* \in \underset{x_i \in X_i}{\operatorname{argmin}} \langle \theta_i^{\phi+\Delta}, x_i \rangle$ implies $D(\phi) \leq D(\phi + \Delta)$.

Example 4. Let us apply Lemma 1 to Example 1. Let ij correspond to $\{u, uv\}$, where $u \in \mathsf{V}$ is some node and $uv \in \mathsf{E}$ is any of its incident edges. Then x_i^* corresponds to a locally optimal label $x_u^* \in \arg\min_{s \in X_u} \theta_u(s)$ and $\nu(s) = [\![s = x_u^*]\!]$. Therefore we may assign $\Delta_{u,uv}(s)$ to any value from $[0, \theta_u(x_u^*) - \theta_u(s)]$. This assures that (20) is fulfilled and x_u^* remains a locally optimal label after reparametrization even if there are multiple optima in X_u.

Lemma 1 can be straightforwardly generalized to the case, when more than two factors must be reparametrized simultaneously. In terms of Example 1 this may correspond to the situation when a graph node sends messages to several incident edges at once:

Definition 2. *Let $i \in \mathbb{F}$ be a factor and $J = \{j_1, \ldots, j_l\} \subseteq \mathcal{N}_{\mathbb{G}}(i)$ be a subset of its neighbors. Let $\theta_i^\Delta := \theta_i + \sum_{j \in J} A_{(i,j)}^\top \Delta_{(i,j)}$, $\Delta_{(i,j)} (= -\Delta_{(j,i)})$ satisfies (20) for all $j \in J$ and all other coordinates of Δ are zero. If there exists $x_i^* \in \operatorname{argmin}_{x_i \in X_i} \langle \theta_i, x_i \rangle$ such that $x_i^* \in \operatorname{argmin}_{x_i \in X_i} \langle \theta_i^\Delta, x_i \rangle$, the dual vector Δ is called* admissible. *The set of admissible vectors is denoted by $AD(\theta_i, x_i^*, J)$.*

Lemma 2. *Let $\Delta \in AD(\theta_i^\phi, x_i^*, J)$ then $D(\phi) \leq D(\phi + \Delta)$.*

Procedure 1: Message-Passing Update Step.

```
Input: Factor $i \in \mathbb{F}$, neighboring factors
  $J = \{j_1, \ldots, j_l\} \subseteq \mathcal{N}_{\mathbb{G}}(i)$, dual variables $\phi$
  Compute $x_i^* \in \arg\min_{x_i \in X_i} \langle \theta^\phi, x_i \rangle$     (13)
  Choose $\delta \in \mathbb{R}^{d_i}$ s.t. $\delta(s) \begin{cases} > 0, & x_i^*(s) = 1 \\ < 0, & x_i^*(s) = 0 \end{cases}$     (14)
Maximize admissible messages to $J$:
  $\Delta^*_{(i,J)} \in \underset{\Delta \in AD(\theta_i^\phi, x_i^*, J)}{\operatorname{argmax}} \langle \delta, \theta_i^{\phi+\Delta} \rangle$     (15)
Output: $\Delta^*_{(i,J)}$.
```

Message-Passing Update Step To maximize $D(\phi)$, we will iteratively visit all factors and adjust messages ϕ connected to it, monotonically increasing the lower bound (11). Such an elementary step is defined by Procedure 1.

Procedure 1 is defined up to the vector δ, which satisfies (14) (see Proc. 1). Usually, $\delta(s) = \begin{cases} 1, & x_i^*(s) = 1 \\ -1, & x_i^*(s) = 0 \end{cases}$ is a good choice. Although different δ may result in different efficiency of our framework, fulfillment of (14) is sufficient to prove its convergence properties.

The reparametrization adjustment problem (15) serves the intuitive goal to move as much slack as possible from the factor i to its neighbors J. For example, for the setting of Example 4 its solution reads $\Delta_{u,uv}(s) = \theta_u^\phi(x_u^*) - \theta_u^\phi(s)$. Depending on the selected δ it might correspond to maximization of the dual objective in the direction defined by admissible reparametrizations. Although maximization (15) is not necessary to prove convergence of our method (as we show below, only a feasible solution of (15) is required for the proof), (i) it leads to faster convergence; (ii) for the case of CRFs (as in Example 1) it makes our method equivalent to well established techniques like TRW-S [48] and SRMP [49], as shown in Section 4.1.

The following proposition states that the elementary update step defined by Procedure 1 can be performed efficiently. That is, the size of the reparametrization adjustment problem (15) grows linearly with the size of the factor i and its attached messages:

Proposition 2. *Let $\operatorname{conv}(X_i) = \{\mu_i : A_i \mu_i \leq b_i\}$ with $A_i \in \mathbb{R}^{n \times m}$. Let the messages in problem (15) have size $n_1, \ldots, n_{|J|}$. Then (15) is a linear program with $O(n + n_1 + \ldots + n_{|J|})$ variables and $O(m + n_1 + \ldots + n_{|J|})$ constraints.*

4. Message Passing Algorithm

Now we combine message passing updates into Algorithm 2. It visits every node of the factor graph and performs the following two operations: (i) **Receive Messages**, when messages are received from a subset of neighboring factors, and (ii) **Send Messages**, when messages to some neighboring factors are computed and reweighted by ω. Distribution of weights ω may influence the efficiency of Algorithm 2 just like it influences the efficiency of message passing for CRFs (see [49]). We provide typical settings in Section 4.1. Usually, factors are traversed in some given a-priori order alternately in forward and backward direction, as done in TRW-S [48] and SRMP [49]. We refer to [49] for a motivation for such a schedule of computations.

We will discuss parameters of Algorithm 2 (factor partitioning $\{J_i\}$, weights ω_{J_i}) right after the theorem stating monotonicity for any choice of parameters.

Theorem 1. *Algorithm 2 monotonically increases the dual lower bound (11).*

Algorithm 2: One Iteration of Message-Passing

```
1  for $i \in \mathbb{F}$ in some order do
2      Receive Messages:
3      Choose a subset of connected factors
         $J_{receive} \subseteq \mathcal{N}_{\mathbb{G}}(i)$
4      for $j \in J_{receive}$ do
5          Compute $\Delta^*_{(j,\{i\})}$ with Procedure 1.
6          Set  $\phi = \phi + \Delta^*_{(j,\{i\})}$.
7      end
8
9      Send Messages:
10     Choose partition $J_1 \dot\cup \ldots \dot\cup J_l \subseteq \mathcal{N}_{\mathbb{G}}(i)$.
11     for $J \in \{J_1, \ldots, J_l\}$ do
12         Compute $\Delta^*_{(i,J)}$ with Procedure 1.
13     end
14     Choose weights $\omega_{J_1}, \ldots, \omega_{J_l} \geq 0$ such that
         $\omega_{J_1} + \ldots + \omega_{J_l} \leq 1$.
15     for $J \in \{J_1, \ldots, J_l\}$ do
16         Set $\phi = \phi + \omega_J \Delta^*_{(i,J)}$.
17     end
18 end
```

4.1. Parameter Selection for Algorithm 2

There are the following free parameters in Algorithm 2: (i) The order of traversing factors of $\mathbb{F}$; (ii) for each factor the neighboring factors from which to receive messages $J_{receive} \subseteq \mathcal{N}_{\mathbb{G}}(i)$; (iii) the partition $J_1 \dot\cup \ldots \dot\cup J_l \subseteq \mathcal{N}_{\mathbb{G}}(i)$ of factors to send messages to and (iv) the associated weights $\omega_{J_1}, \ldots, \omega_{J_l}$ for messages.

Although for any choice of these parameters Algorithm 2 monotonically increases the dual lower bound (as stated by Theorem 1), its efficiency may significantly depend on their values. Below, we will describe the parameters for Examples 1-3, which we found the most efficient empirically.

Sending a message by some factor automatically implies receiving this message by another, coupled factor. Therefore, usually there is no need to go over all factors in Algorithm 2. It is usually sufficient to guarantee that all coupling constraints are updated by Procedure 1. Formally, we can always exclude processing some factors by setting $J_{receive}$ and J_i, $i = 1, \ldots, l$ to the empty set. Instead, we will explicitly specify, which factors are processed in the loop of Algorithm 2 in the examples below.

Parameters for Example 1, MAP-inference in CRFs. Pairwise CRFs have the specific feature that node factors are coupled with edge factors only. This implies that processing only node factors in Algorithm 2 is sufficient. Below, we describe parameters, which turn Algorithm 2 into SRMP [49] (which is up to details of implementation equivalent to TRW-S [48] for pairwise CRFs). Other settings, given in the supplement, may turn it to other popular message passing techniques like MPLP [28] or min-sum diffusion [64].

We order node factors and process them according to this ordering. The ordering naturally defines the sets of incoming E^+_u and outgoing E^-_u edges for each node $u \in \mathsf{V}$. Here $uv \in \mathsf{E}$ is *incoming* for u if $v < u$ and *outgoing* if $v > u$. Each node $u \in \mathsf{V}$ receives messages from all incoming edges, which is $J_{receive} = \mathcal{N}_{\mathbb{G}}(u) = \mathsf{E}^+_u$. The messages are sent to all outgoing edges. Each edge $uv \in \mathsf{E}$ in the partition in line 10 of Algorithm 2 is represented by a separate set. That is, the partition reads $\dot\cup_{e \in \mathsf{E}^-_u} \{e\}$. Weights are distributed uniformly and equal to $w_e = \{\frac{1}{\max\{|\mathsf{E}^-_u|, |\mathsf{E}^+_u|\}}\}$, $e \in \mathsf{E}^-_u$. After each outer iteration, when all nodes were processed, the ordering is reversed and the process repeats. We refer to [49] for substantiation of these parameters.

Parameters for Example 2, Graph Matching. Additionally to the node and edge factors, the corresponding IRPS-LP has also label factors (7). To this end all node factors are ordered, as in Example 1. Each node factor $u \in \mathsf{V}$ receives messages from all incoming edge factors and label factors $J_{receive}(u) = \mathsf{E}^+_u \cup X_u$ and sends them to all outgoing edges *and* label factors. The corresponding partition reads $\dot\cup_{f \in \mathcal{N}_{\mathbb{G}}(u) \backslash \mathsf{E}^+_u} \{f\} \dot\cup X_u$. The weights are distributed uniformly with $w_f = \{\frac{1}{1+\max\{|\mathsf{E}^-_u|, |\mathsf{E}^+_u|\}}\}$. The label factors are processed after all node factors were visited. Each label factor receives messages from all connected node factors and send messages back as well: $J_{receive}(s) = \{u \in \mathsf{V} : s \in X_u\}$. We use the same single set for sending messages, i.e. $J_1 = J_{receive}$. After each iteration we reverse the factor order.

Parameters for Example 3, Multicut. Similarly to Example 1, it is sufficient to go only over all edge factors in the loop of Algorithm 2, since each coupling constraint contains exactly one cycle and one edge factor. Each edge factor e receives messages from all coupled cycle factors $J_{receive} = \mathcal{N}_{\mathbb{G}}(\{c \in C : e \in c\})$ and sends them to the same factors. As in Example 1, each cycle factor forms a trivial set in the partition in line 10 of Algorithm 2, the partition reads $\dot\cup_{c \in C : e \in c} \{c\}$. Weights are distributed uniformly with $w_e = \frac{1}{|c \in C : e \in c|}$. After each iteration the processing order of factors is reversed.

4.2. Obtaining Integer Solution

Eventually we want to obtain a primal solution $x \in X$ of (1), not a reparametrization θ^ϕ. We are not aware of any rounding technique which would work equally well for all possible instances of IRPS-LP problem. According to our experience, the most efficient rounding is problem specific. Below, we describe our choices for the Examples 1 – 3.

Rounding for Example 1 coincides with the one suggested in [48]: Assume we have already computed a primal integer solution x_v^* for all $v < u$ and we want to compute x_u^*. To this end, right before the message receiving step of Algorithm 2 for $i = u$ we assign

$$x_u^* \in \operatorname*{argmin}_{x_u} \theta_u(x_u) + \sum_{v<u:uv\in \mathsf{E}} \theta_{uv}(x_u, x_v^*)\,. \qquad (16)$$

Rounding for Example 2 is the same except that we select the best label x_u among those, which have not been assigned yet, to satisfy uniqueness constraints:

$$x_u^* \in \operatorname*{argmin}_{x_u : x_v^* \neq x_u \forall v<u} \theta_u^\phi(x_u) + \sum_{v<u:uv\in \mathsf{E}} \theta_{uv}^\phi(x_u, x_v^*)\,. \quad (17)$$

Rounding for Example 3. We use the efficient Kernighan&Lin heuristic [45] as implemented in [46]. Costs for the rounding are the reparametrized edge potentials.

5. Fixed Points and Comparison to Subgradient Method

Algorithm 2 does not necessarily converge to the optimum of (1). Instead, it may get stuck in suboptimal points, similar to those correspoding to the "weak tree agreement" [48] or "arc consistency" [74, 3] in CRFs from Example 1. Below we characterise these fixpoints precisely.

Definition 3 (Marginal Consistency)**.** *Given a reparametrization θ^ϕ, let for each factor $i \in \mathbb{F}$ a non-empty set $\mathbb{S}_i \subseteq \operatorname{argmin}_{x_i \in X_i} \langle \theta^\phi, x_i \rangle$, $i \in \mathbb{F}$ be given. Define $\mathbb{S} = \prod_{i\in\mathbb{F}} \mathbb{S}_i$. We call reparametrization θ^ϕ* marginally consistent for $\mathbb{S}$ on $ij \in \mathbb{E}$ *if*

$$A_{(i,j)}\left(\mathbb{S}_i\right) = A_{(j,i)}\left(\mathbb{S}_j\right)\,. \qquad (18)$$

If θ^ϕ is marginally consistent for $\mathbb{S}$ on all $ij \in \mathbb{E}$, we call θ^ϕ marginally consistent for $\mathbb{S}$.

Note that marginal consistency is necessary, but not sufficient for optimality of the relaxation (1). This can be seen in the case of CRFs (Example 1), where it exactly corresponds to arc-consistency. The latter is only necessary, but not sufficient for optimality [74].

Theorem 2. *If θ^ϕ is marginally consistent, the dual lower bound $D(\phi)$ cannot be improved by Algorithm 2.*

Comparison to Subgradient Method. Decomposition IRPS-LP and more general ones can be solved via the subgradient method [50]. Similar to Algorithm 2, it operates on dual variables ϕ and manipulates them by visiting each factor sequentially. Contrary to Algorithm 2, subgradient algorithms converge to the optimum. Moreover, on a per-iterations basis, computing subgradients is cheaper than using Algorithm 2, as only (13) needs to be computed, while Algorithm 2 needs to solve (15) additionally. However, for MAP-inference, the study [41] has shown that subgradient-based algorithms converge much slower than message passing algorithms like TRWS [48]. In Section 6 we confirm this for the graph matching problem as well.

The reason for this large empirical difference is that one iteration of the subgradient algorithm only updates those coordinates of dual variables ϕ that are affected by the current minimal labeling $x_i^* \in \operatorname{argmin}_{x_i\in X_i} \langle \theta_i^\phi, x_i \rangle$ (i.e. coordinates $k : (A_{(i,j)}^\top x_i^*)_k = 1$), while in Algorithm 2 *all* coordinates of ϕ are taken into account. Also message passing implicitly chooses the stepsize so as to achieve monotonical convergence in Algorithm 1, while subgradient based algorithms must rely on some stepsize rule.

6. Experimental Evaluation

Our experiments' goal is to illustrate applicability of the proposed technique, they are not an exhaustive evaluation. The presented algorithms are only basic variants, which can be further improved and tuned to the considered problems. Both issues are addressed in the specialized studies [68, 67]. Still, we show that the presented basic variants are already able to surpass state-of-the-art specialized solvers on challenging datasets.

6.1. Graph Matching

Solvers. We compare against two state-of-the-art algorithms: (i) the subgradient based dual decomposition solver [70] abbreviated by **DD** and (ii) the recent "hungarian belief propagation" message passing algorithm [78], abbreviated as **HBP**. While the authors of [78] have embedded their solver in a branch-and-bound routine to produce exact solutions, we have reimplemented their message passing component but did not use branch and bound to make the comparison fair. Both algorithms **DD** and **HBP** outperformed alternative solvers [18, 76, 53, 69, 63, 29, 79, 38, 54, 16] at the time of their publication, hence we have not tested against them. We call our solver **AMP**.

Datasets. We selected three challenging datasets. The first two are the standard benchmark datasets `car` and `motor`, both used in [55], containing 30 pairs of cars and 20 pairs of motorbikes with keypoints to be matched 1:1. The images are taken from the VOC PASCAL 2007 challenge [21]. Costs are computed from features as in [55]. Instances are densely connected graphs with 20 – 60 nodes. The third one is the novel `worms` datasets [40], containing 30 problem instances coming from bioimaging. The problems are made of sparsely connected graphs with up to 600 nodes and up to 1500 labels. To our knowledge, the `worms` dataset contains the largest

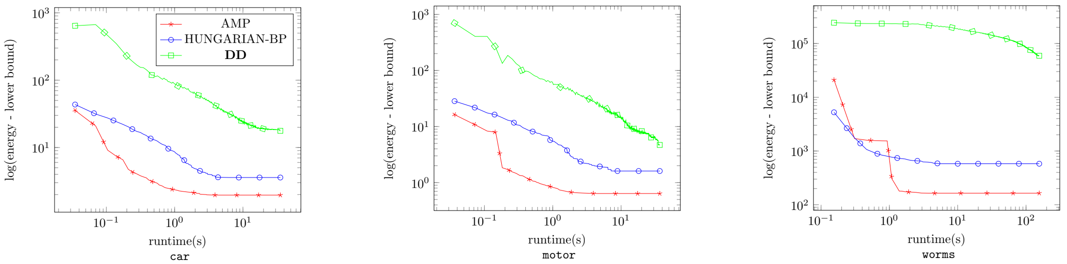

Figure 2. Runtime plots comparing averaged $\log(\text{primal energy} - \text{dual lower bound})$ values on `car`, `motor` and `worms` graph matching datasets. Both axes are logarithmic.

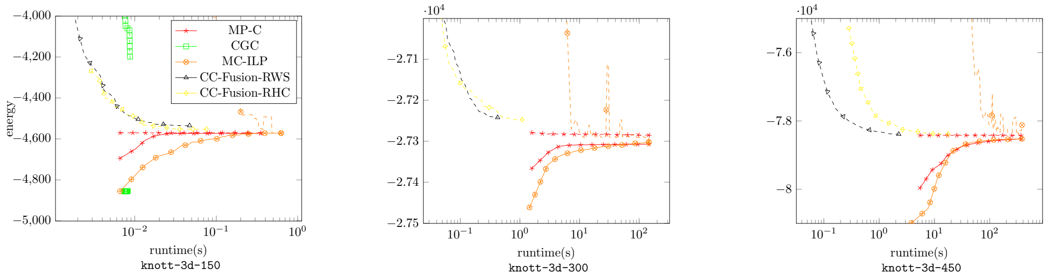

Figure 3. Runtime plots comparing averaged primal/dual values on the three `knott-3d-`$\{150|300|450\}$ multicut datasets. Values are averaged over all instances in the dataset. The x-axis are logarithmic. Continuous lines are dual lower bounds while corresponding dashed lines show primal solutions obtained by rounding.

graph matching instances ever considered in the literature. For runtime plots showing averaged logarithmic primal/dual gap over all instances of each dataset see Fig. 2.

Results. Our solver **AMP** consistently outperforms **HBP** and **DD** w.r.t. primal/dual gap and anytime performance Most markedly on the largest `worms` dataset, the subgradient based algorithm **DD** struggles hard to decrease the primal/dual gap, while **AMP** gives reasonable results.

6.2. Multicuts

Solvers. We compare against state-of-the-art multicut algorithms implemented in the OpenGM [41] library, namely (i) the branch-and-cut based solver **MC-ILP** [44] utilizing the ILP solver CPLEX [2], (ii) the heuristic primal "fusion move" algorithm **CC-Fusion** [9] with random hierarchical clustering and random watershed proposal generator, denoted by the suffixes **-RHC** and **-RWS** and (iii) the heuristic primal "Cut, Glue & Cut" solver **CGC** [10]. Those solvers were shown to outperform other multicut algorithms [9]. Algorithm **MC-ILP** provides both upper and lower bounds, while **CC-Fusion** and **CGC** are purely primal algorithms. We call our message passing solver with cycle constraints added in a cutting plane fashion **MP-C**.

Datasets. We have selected three datasets `knott-3d-`$\{150|300|450\}$ from OpenGM [41]. The problems come from electron microscopy of brain tissue, for which we wish to obtain a neuron segmentation. Each dataset contains 8 instances with $\leq$ 972, 5896 and 17074 nodes and $\leq$ 5656, 36221, and 107060 edges respectively.

Results. For plots showing dual bounds and primal solution objectives over time see Figure 3. Our algorithm **MP-C** combines advantages of LP-based techniques awith those of primal heuristics: It delivers high dual lower bounds faster than **MC-ILP**. Its has fast primal convergence speed and delivers primal solutions comparable/superior to **CGC**'s and **CC-Fusion**'s.

7. Acknowledgments

The authors would like to thank Vladimir Kolmogorov for helpful discussions. The first authors was supported by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 616160, the second author by DFG Grant GRK 1653 and the last author by the European Research Council under the European Unions Horizon 2020 research and innovation programme (grant agreement No 647769) and the DFG Grant "ERBI" SA 2640/1-1.

References

[1] https://en.wikipedia.org/wiki/ARPANET. 1

[2] IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/. 1, 8

[3] Marginal consistency: Upper-bounding partition functions over commutative semirings. *IEEE TPAMI*, 37(7):1455–1468, 2015. 7

[4] A. Alush and J. Goldberger. Break and conquer: Efficient correlation clustering for image segmentation. In E. R. Hancock and M. Pelillo, editors, *SIMBAD*, volume 7953 of *Lecture Notes in Computer Science*, pages 134–147. Springer, 2013. 3

[5] B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. Probabilistic image segmentation with closedness constraints. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, *ICCV*, pages 2611–2618. IEEE Computer Society, 2011. 3

[6] B. Andres, T. Kröger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Köthe, and F. A. Hamprecht. Globally optimal closed-surface segmentation for connectomics. In A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *ECCV (3)*, volume 7574 of *Lecture Notes in Computer Science*, pages 778–791. Springer, 2012. 3

[7] A. Arasu, C. R, and D. Suciu. Large-scale deduplication with constraints using dedupalog. In Y. E. Ioannidis, D. L. Lee, and R. T. Ng, editors, *ICDE*, pages 952–963. IEEE Computer Society, 2009. 3

[8] C. Arora and A. Globerson. Higher order matching for consistent multiple target tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 177–184, 2013. 2

[9] T. Beier, F. A. Hamprecht, and J. H. Kappes. Fusion moves for correlation clustering. In *CVPR*, pages 3507–3516. IEEE Computer Society, 2015. 3, 8

[10] T. Beier, T. Kröger, J. H. Kappes, U. Köthe, and F. A. Hamprecht. Cut, glue & cut: A fast, approximate solver for multicut partitioning. In *CVPR. Proceedings*, 2014. 3, 8

[11] O. Bilde and J. Krarup. Sharp lower bounds and efficient algorithms for the simple plant location problem. *Annals of Discrete Mathematics*, 1:79–97, 1977. 1

[12] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 1

[13] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis. *The Quadratic Assignment Problem*, pages 1713–1809. Springer US, Boston, MA, 1999. 3

[14] Y. Chen, S. Sanghavi, and H. Xu. Clustering sparse graphs. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NIPS*, pages 2213–2221, 2012. 3

[15] F. Chierichetti, N. Dalvi, and R. Kumar. Correlation clustering in mapreduce. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 641–650, New York, NY, USA, 2014. ACM. 3

[16] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV (5)*, volume 6315 of *Lecture Notes in Computer Science*, pages 492–505. Springer, 2010. 3, 7

[17] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1):87–115, 1993. 3, 4

[18] J. Duchi, D. Tarlow, G. Elidan, and D. Koller. Using combinatorial optimization within max-product belief propagation. In *NIPS*, pages 369–376. MIT Press, 2006. 2, 3, 7

[19] M. Elsner and E. Charniak. You talking to me? a corpus and algorithm for conversation disentanglement. In K. McKeown, J. D. Moore, S. Teufel, J. Allan, and S. Furui, editors, *ACL*, pages 834–842. The Association for Computer Linguistics, 2008. 3

[20] M. Elsner and W. Schudy. Bounding and comparing methods for correlation clustering beyond ILP. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 19–27, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. 3

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 7

[22] M. L. Fisher and D. S. Hochbaum. Database location in computer networks. *Journal of the ACM (JACM)*, 27(4):718–735, 1980. 1

[23] M. L. Fisher, R. Jaikumar, and L. N. Van Wassenhove. A multiplier adjustment method for the generalized assignment problem. *Management Science*, 32(9):1095–1103, 1986. 1

[24] M. L. Fisher and P. Kedia. Optimal solution of set covering/partitioning problems using dual heuristics. *Management science*, 36(6):674–688, 1990. 2

[25] D. Gamarnik, D. Shah, and Y. Wei. Belief propagation for min-cost network flow: Convergence & correctness. In *SODA*, pages 279–292. SIAM, 2010. 1

[26] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. 4

[27] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *ACM Trans. Knowl. Discov. Data*, 1(1):4, 2007. 3

[28] A. Globerson and T. S. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, pages 553–560, 2007. 1, 2, 6, 13, 14

[29] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(4):377–388, 1996. 3, 7

[30] M. Guignard. A Lagrangean dual ascent algorithm for simple plant location problems. *European Journal of Operational Research*, 35(2):193–200, 1988. 1

[31] M. Guignard and S. Kim. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical programming*, 39(2):215–228, 1987. 1

[32] M. Guignard and S. Kim. Lagrangean decomposition for integer programming: theory and applications. *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle*, 21(4):307–323, 1987. 1

[33] M. Guignard and M. B. Rosenwein. An application-oriented guide for designing Lagrangean dual ascent algorithms. *Euro-*

pean Journal of Operational Research, 43(2):197–205, 1989. 2

[34] M. Guignard and M. B. Rosenwein. Technical note-an improved dual based algorithm for the generalized assignment problem. *Operations Research*, 37(4):658–663, 1989. 1

[35] M. Guignard and M. B. Rosenwein. An application of Lagrangean decomposition to the resource-constrained minimum weighted arborescence problem. *Networks*, 20(3):345–359, 1990. 2

[36] Gurobi Optimization, Inc., 2015. `http://www.gurobi.com`. 1

[37] J. Jancsary and G. Matz. Convergent decomposition solvers for tree-reweighted free energies. In *AISTATS 2011*, 2011. 2

[38] B. Jiang, J. Tang, C. Ding, and B. Luo. A local sparse model for matching problem. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3790–3796. AAAI Press, 2015. 3, 7

[39] J. K. Johnson, D. M. Malioutov, and A. S. Willsky. Lagrangian relaxation for map estimation in graphical models. In *In Allerton Conf. Communication, Control and Computing*, 2007. 2

[40] D. Kainmueller, F. Jug, C. Rother, and G. Myers. Active graph matching for automatic joint segmentation and annotation of C. elegans. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 81–88. Springer, 2014. 7

[41] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*, 115(2):155–184, 2015. 1, 2, 7, 8

[42] J. H. Kappes, B. Savchynskyy, and C. Schnörr. A bundle approach to efficient MAP-inference by Lagrangian relaxation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1688–1695. IEEE, 2012. 1

[43] J. H. Kappes, M. Speth, B. Andres, G. Reinelt, and C. Schnörr. Globally optimal image partitioning by multicuts. In *EMMCVPR*. Springer, Springer, 2011. 3, 4

[44] J. H. Kappes, M. Speth, G. Reinelt, and C. Schnörr. Higher-order segmentation via multicuts. *CoRR*, abs/1305.6387, 2013. 3, 4, 8

[45] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell Systems Technical Journal*, 49(2), 1970. 7

[46] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *ICCV*, 2015. 7

[47] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo. Higher-order correlation clustering for image segmentation. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *NIPS*, pages 1530–1538, 2011. 3

[48] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1568–1583, 2006. 1, 2, 5, 6, 7, 13, 14

[49] V. Kolmogorov. A new look at reweighted message passing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(5):919–930, 2015. 1, 2, 5, 6, 13, 14

[50] N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):531–552, March 2011. 1, 7

[51] C. Lemaréchal. Lagrangian decomposition and nonsmooth optimization: Bundle algorithm, prox iteration, augmented Lagrangian. *Nonsmooth Optimization: Methods and Applications*, pages 201–216, 1992. 1

[52] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for Markov random field optimization. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1392–1405, 2010. 1

[53] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, pages 1482–1489. IEEE Computer Society, 2005. 3, 7

[54] M. Leordeanu, M. Hebert, and R. Sukthankar. An integer projected fixed point method for graph matching and MAP inference. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *NIPS*, pages 1114–1122. Curran Associates, Inc., 2009. 3, 7

[55] M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *International Journal of Computer Vision*, 96(1):28–45, 2012. 7

[56] T. Meltzer, A. Globerson, and Y. Weiss. Convergent message passing algorithms - a unifying view. In *UAI*, pages 393–401. AUAI Press, 2009. 2

[57] I. Necoara and J. A. Suykens. Application of a smoothing technique to decomposition in convex optimization. *IEEE Transactions on Automatic Control*, 53(11):2674–2679, 2008. 1

[58] V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, (July):104, 2001. 3

[59] C. Ribeiro and M. Minoux. Solving hard constrained shortest path problems by Lagrangean relaxation and branch-and-bound algorithms. *Methods of Operations Research*, 53:303–316, 1986. 1

[60] E. Sadikov, J. Madhavan, L. Wang, and A. Halevy. Clustering query refinements by user intent. In *World Wide Web Conference (WWW)*. ACM Press, April 2010. 3

[61] B. Savchynskyy, J. Kappes, S. Schmidt, and C. Schnörr. A study of Nesterov's scheme for Lagrangian decomposition and MAP labeling. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1817–1823. IEEE, 2011. 1

[62] B. Savchynskyy, S. Schmidt, J. H. Kappes, and C. Schnörr. Efficient MRF energy minimization via adaptive diminishing smoothing. In *UAI*, pages 746–755. AUAI Press, 2012. 2

[63] C. Schellewald and C. Schnörr. Probabilistic subgraph matching based on convex relaxation. In *EMMCVPR*, volume 3757 of *Lecture Notes in Computer Science*, pages 171–186. Springer, 2005. 3, 7

[64] M. I. Schlesinger and K. V. Antoniuk. Diffusion algorithms and structural recognition optimization problems. *Cybernetics and Systems Analysis*, 47(2):175–192, 2011. 6

[65] D. Sontag and T. S. Jaakkola. Tree block coordinate descent for map in graphical models. In *AISTATS*, volume 5 of *JMLR Proceedings*, pages 544–551, 2009. 14

[66] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001. 3

[67] P. Swoboda and B. Andres. A message passing algorithm for the minimum cost multicut problem. In *CVPR*, 2017. 4, 7

[68] P. Swoboda, C. Rother, H. Abu Alhaija, D. Kainmueller, and B. Savchynskyy. Study of Lagrangean decomposition and dual ascent solvers for graph matching. In *CVPR*, 2017. 7

[69] P. H. S. Torr. Solving Markov random fields using semi definite programming. In *In: AISTATS*, 2003. 3, 7

[70] L. Torresani, V. Kolmogorov, and C. Rother. A dual decomposition approach to feature correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(2):259–271, 2013. 3, 7

[71] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008. 3

[72] H. Wang and D. Koller. Subproblem-tree calibration: A unified approach to max-product message passing. In *30th International Conference on Machine Learning (ICML-13)*, pages 190–198, 2013. 2, 14

[73] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001. 2

[74] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(7):1165–1179, 2007. 2, 3, 7, 13, 14

[75] T. Werner. Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1474–1488, 2010. 2

[76] J. Yarkony, C. C. Fowlkes, and A. T. Ihler. Covering trees and lower-bounds on quadratic assignment. In *CVPR*, pages 887–894. IEEE Computer Society, 2010. 3, 7

[77] J. Yarkony, A. Ihler, and C. C. Fowlkes. *Fast Planar Correlation Clustering for Image Segmentation*, pages 568–581. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 3

[78] Z. Zhang, Q. Shi, J. McAuley, W. Wei, Y. Zhang, and A. van den Hengel. Pairwise matching through max-weight bipartite belief propagation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 3, 7

[79] F. Zhou and F. D. la Torre. Factorized graph matching. In *CVPR*, pages 127–134. IEEE Computer Society, 2012. 3, 7

8. Supplementary Material

Proof of Proposition 1

Proposition. $\sum_{i\in\mathbb{F}}\langle\theta_i,\mu_i\rangle = \sum_{i\in\mathbb{F}}\langle\theta_i^{\phi},\mu_i\rangle$, *whenever* $\mu_1,\ldots,\mu_k$ *obey the coupling constraints.*

Proof. $\sum_{i\in\mathbb{F}}\langle\theta^{\phi},\mu_i\rangle = \sum_{i\in\mathbb{F}}\langle\theta,\mu_i\rangle + \underbrace{\sum_{ij\in\mathbb{E}}\langle\phi_{(i,j)},A_{(i,j)}\mu_i\rangle + \langle\phi_{(j,i)},A_{(j,i)}\mu_j\rangle}_{(*)} = \sum_{i\in\mathbb{F}}\langle\theta,\mu_i\rangle$, where $(*) = 0$ due to $\phi_{(i,j)} = -\phi_{j,i)}$ and $A_{(i,j)}\mu_i = A_{(j,i)}\mu_j$. □

Proof of Proposition 2

Proposition. *Let* $\operatorname{conv}(X_i) = \{\mu_i : A_i\mu_i \leq b_i\}$ *with* $A_i \in \mathbb{R}^{n\times m}$. *Let the messages in problem* (15) *have size* $n_1,\ldots,n_{|J|}$. *Then* (15) *is a linear program with* $O(n+n_1+\ldots+n_{|J|})$ *variables and* $O(m+n_1+\ldots+n_{|J|})$ *constraints.*

Proof. From LP-duality we know that $\mu_i^* \in \operatorname{argmin}_{\mu_i : A\mu_i \leq b_i}\langle c,\mu_i\rangle$ iff $\exists y \geq 0 : A_i^\top y = c_i$ and $\langle b_i - A_i\mu_i^*, y\rangle = 0$. Hence, (15) can be rewritten as

$$\begin{array}{cl} \displaystyle\max_{y\geq 0,\Delta_{(i,j_1)},\ldots,\Delta_{(i,j_l)}} & \langle\delta,\theta^{\phi+\Delta}\rangle \\ \text{s.t.} & \langle b_i - A_i\mu_i^*, y\rangle = 0 \\ & A_i^\top y = \theta^{\phi+\Delta} \\ & \Delta_{(i,j)}(s)\begin{cases}\leq 0, & \nu_i(s) = 0\\ \geq 0, & \nu_i(s) = 1\end{cases} \\ & \quad\text{where } \nu_i := A_{(i,j)}\mu_i^* \end{array} \tag{19}$$

$\theta^{\phi+\Delta}$ is a linear expression and μ^* is constant during the computation, hence (19) is a LP. □

Proof of Lemma 1 and Lemma 2

Lemma. *Let* $ij \in \mathbb{E}$ *be a pair of factors related by the coupling constraints and* $\phi_{(i,j)}$ *be a corresponding dual vector. Let* $x_i^* \in \operatorname*{argmin}_{x_i\in X_i}\langle\theta_i^{\phi},x_i\rangle$ *and* $\Delta_{(i,j)}$ *satisfy*

$$\Delta_{(i,j)}(s)\begin{cases}\geq 0, & \nu(s) = 1\\ \leq 0, & \nu(s) = 0\end{cases}, \text{ where } \nu := A_{(i,j)}x_i^*\,. \tag{20}$$

Then $x_i^* \in \operatorname*{argmin}_{x_i\in X_i}\langle\theta_i^{\phi+\Delta},x_i\rangle$ *implies* $D(\phi) \leq D(\phi+\Delta)$.

Proof. Let $x_j^* \in \operatorname{argmin}_{x_j\in X_j}\langle\theta_j^{\phi},x_j\rangle$ be a solution of (13) at which the dual lower bound (11) is attained before the update and $x_j^{**} \in \operatorname{argmin}_{x_j\in X_j}\langle\theta^{\phi} - A_{(j,i)}^\top\Delta_{(i,j)}^*, x_j\rangle$ be an integral solution at which the dual lower bound is attained after ϕ has been updated. Variable x_i^* as chosen in (13) is optimal for θ^{ϕ} and for $\theta^{\phi+\Delta}$ by construction. We need to prove

$$\begin{aligned}&\langle\theta_i^{\phi},x_i^*\rangle + \sum_{j\in J}\langle\theta_j^{\phi},x_j^*\rangle\\ &\leq \langle\theta_i^{\phi}+\sum_{j\in J}A_{(i,j)}^\top\Delta_{(i,j)}^*, x_i^*\rangle + \sum_{j\in J}\langle\theta_j^{\phi}-A_{(j,i)}^\top\Delta_{(i,j)}^*, x_j^{**}\rangle\,.\end{aligned} \tag{21}$$

We shuffle all terms with variables $\Delta_{(i,j)}^*$, $j\in J$ to the right side and all other terms to the left side.

$$\begin{aligned}&\langle\theta_i^{\phi},x_i^* - x_i^*\rangle + \sum_{j\in J}\langle\theta_j^{\phi},x_j^* - x_j^{**}\rangle\\ &\leq \langle\sum_{j\in}A_{(i,j)}^\top\Delta_{(i,j)}^*, x_i^*\rangle - \sum_{j\in J}\langle A_{(j,i)}^\top\Delta_{(i,j)}^*, x_j^{**}\rangle\end{aligned} \tag{22}$$

All terms on the left side are smaller than zero due to the choice of x_j^* being minimizers w.r.t. θ_j^{ϕ}. Hence, it will be enough to prove the above inequality when assuming the left side to be zero. We rewrite the scalar products by transposing $A_{(i,j)}^\top$ and $A_{(j,i)}^\top$.

$$0 \leq \sum\nolimits_{j\in J}\left\{\langle\Delta_{(i,j)}^*, A_{(i,j)}x_i^* - A_{(j,i)}x_j^{**}\rangle\right\} \tag{23}$$

Due to $A_{(j,i)}x_j^{**} \in \{0,1\}^{\dim(\phi_{(i,j)})}$ and $A_{(i,j)}x_i^* \in \{0,1\}^{\dim(\phi_{(i,j)})}$ by Definition 1 and $\Delta_{(i,j)}^* \lesseqgtr 0$ whenever $A_{(i,j)}x_i^* \lesseqgtr 0$, the result follows. □

Lemma. *Let* $\Delta \in AD(\theta_i^{\phi}, x_i^*, J)$ *then* $D(\phi) \leq D(\phi+\Delta)$.

Proof. Analoguous to the proof of Lemma 1. □

Proof of Theorem 1

Theorem. *Algorithm 2 monotonically increasis the dual lower bound* (11).

Proof. We prove that (i) the receiving messages and (ii) the sending messages step improve (11).
(i) Directly apply Lemma 1. (ii) The difficulty here is that we compute descent directions from the current dual variables ϕ in parallel and then apply all of them simultaneously. By Lemma 2, the send message step is non-decreasing when called for each set $J_1,\ldots,J_l$ in Algorithm 2. The dual lower bound $L(\phi)$ is concave, hence we apply Jensen's inequality and note that $\omega_1+\ldots+\omega_l \leq 1$ to obtain the result. □

Proof of Theorem 2

Theorem. *If* θ^{ϕ} *is marginally consistent, the dual lower bound* $D(\phi)$ *cannot be improved by Algorithm 2.*

First, we need two technical lemmata.

Lemma 3. *Let* $X \subset \{0,1\}^n$, $A \in \{0,1\}^{K\times n}$ *and* $Ax \in \{0,1\}^K$ $\forall x \in X$. *Let* $x^* \in X$ *be given and define* $\nu^* := Ax^*$. *Let* $\varDelta \in \mathbb{R}^K$ *be given such that* $\varDelta(s)\begin{cases}\geq 0, & \nu^*(s)=1\\ \leq 0, & \nu^*(s)=0\end{cases}$. *Then (i)* $x^* \in \mathrm{argmin}_{x\in X}\langle -\varDelta, Ax\rangle$ *and (ii) for* $x^{**} \in \mathrm{argmin}_{x\in X}\langle -\varDelta, Ax\rangle$, $\nu^{**} = Ax^{**}$ *it holds that* $\varDelta(s)=0$ *whenever* $\nu^*(s) \neq \nu^{**}(s)$.

Proof. Let $x \in X$ and define $\nu = Ax$. Then

$$\begin{aligned}&\langle -\varDelta, Ax\rangle\\ &\quad= \underbrace{\sum_{s:\nu^*(s)=1=\nu(s)} -\varDelta(s)}_{(*)} + \underbrace{\sum_{s:\nu(s)=1>0=\nu^*(s)} -\varDelta(s)}_{(**)}\\ &\qquad\qquad \geq \underbrace{\sum_{s:\nu^*(s)=1} -\varDelta(s)}_{(***)}\\ &\qquad\qquad\qquad\qquad = \langle -\varDelta, Ax^*\rangle \quad (24)\end{aligned}$$

because $(*) \geq (***)$ due to $\varDelta(s) \geq 0$ for $\nu^*(s)=1$ and $(**) \geq 0$ due to $\varDelta(s) \leq 0$ for $\nu^*(s)=0$. This proves (i) and (ii) is proven by observing that $(**) = 0$ and $(*) = (***)$ must also hold. □

Lemma 4. *Let* $x_i^*, x_i^{**} \in \mathrm{argmin}_{x_i\in X_i}\langle \theta^\phi, x_i\rangle$ *be two solutions to the* i*-th factor for the current reparametrization* θ^ϕ. *If* $\varDelta$ *is admissible w.r.t.* x_i^* *then* $\varDelta$ *is also admissible w.r.t.* x_i^{**}.

Proof. As both x_i^* and x_i^{**} are optimal to θ^ϕ and x_i^* is also optimal to $\theta^{\phi+\varDelta}$, we have $\langle \varDelta_{(i,j)}, A_{(j,i)}x_i^*\rangle \leq \langle \varDelta_{(i,j)}, A_{(j,i)}x_i^{**}\rangle$. By Lemma 3, (i) also $\langle -\varDelta_{(i,j)}, A_{(j,i)}x_i^*\rangle \leq \langle -\varDelta_{(i,j)}, A_{(j,i)}x_i^{**}\rangle$ holds, hence equality must hold. This shows $x_i^{**} \in \mathrm{argmin}_{x_i\in X_i}\langle \theta^{\phi+\varDelta}, x_i\rangle$. Second, Lemma 3, (ii) implies that $\varDelta(s)=0$ whenever $\nu^*(s) \neq \nu^{**}(s)$. This proves that $\varDelta_{(i,j)}(s)\begin{cases}\geq 0, & \nu^{**}(s)=1\\ \leq 0, & \nu^{**}(s)=0\end{cases}$, $\nu^{**} := A_{(i,j)}x_i^{**}$. □

Proof of Theorem 2. It is sufficient to show that for marginally consistent θ^ϕ for $\mathbb{S}$, the update $\varDelta$ computed by Algorithm 1 on an arbitrary factor $i \in \mathbb{F}$ and some set $J \subset \mathcal{N}_{\mathbb{G}}(i)$ has the following properties: (i) $L(\phi) = L(\phi+\varDelta)$, (ii) $\theta^{\phi+\varDelta}$ is marginally consistent for $\mathbb{S}$. For an easier proof, we only consider the case $J = \{j\}$. The general case can be proven analoguously.

(i) Let $x_i^* \in \mathbb{S}_i$, $x_j^* \in \mathbb{S}_j$ with $A_{(i,j)}x_i^* = A_{(j,i)}x_j^*$. We have to show that

$$\min_{x_i\in X_i}\langle \theta_i^\phi, x_i\rangle + \min_{x_j\in X_j}\langle \theta_j^\phi, x_j\rangle = \min_{x_i\in X_i}\langle \theta_i^{\phi+\varDelta}, x_i\rangle + \min_{x_j\in X_j}\langle \theta_j^{\phi+\varDelta}, x_j\rangle \quad (25)$$

Due to x_i^* optimal to $\theta_i^{\phi+\varDelta}$, since by Lemma 4 the update $\varDelta$ is admissible for x_i^*, it remains to show that $x_j^* \in \mathrm{argmin}_{x_j\in X_j}\langle \theta^{\phi+\varDelta}, x_j\rangle$. As $x_j^* \in \mathrm{argmin}_{x_j\in X_j}\langle \theta^\phi, x_j\rangle$, it is sufficient to prove that $x_j^* \in \mathrm{argmin}_{x_j\in X_j}\langle -\varDelta_{(i,j)}, A_{(j,i)}x_j\rangle$. This follows from Lemma 3 (i). We conclude by noting $\langle \theta_i^\phi, x_i^*\rangle + \langle \theta_j^\phi x_j\rangle = \langle \theta_i^{\phi+\varDelta}, x_i^*\rangle + \langle \theta_j^{\phi+\varDelta} x_j\rangle$.

(ii) The computations in (i) show that $\mathbb{S}_i \subseteq \mathrm{argmin}_{x_i\in X_i}\langle \theta_i^{\phi+\varDelta}, x_i\rangle$ and $\mathbb{S}_j \subseteq \mathrm{argmin}_{x_j\in X_j}\langle \theta_j^{\phi+\varDelta}, x_j\rangle$. The reparametrizations of all other factors stay the same: $\theta_k^{\phi+\varDelta} = \theta_k^\phi$ for $k \in \mathbb{F}\backslash\{i,j\}$. Hence, $\theta^{\phi+\varDelta}$ is marginally consistent for $\mathbb{S}$ after the update. □

9. Special Cases: Graphical Model Solvers

We will show how Algorithm 2 subsumes known message-passing algorithms MSD [74], TRWS [48], SRMP [49] and MPLP [28] for MAP-inference with common graphical models, considered in Example 1.

Solver Primitives (13) and (15). As it can be seen, all factors in (5) are of the form $X_i = \{(1,0,\dots,0),(0,1,0,\dots,0),\dots,(0,\dots,0,1)\}$ and $\mathrm{conv}(X_i) = \{\mu \geq 0 : \langle \mathbb{1}, \mu\rangle = 1\}$ is a $\mathrm{dim}_(X_i)$-dimensional simplex.

In all message passing algorithms [48, 49, 74, 28], there are two types of invokations of Algorithm 1 together with solutions of the accompanying optimization problem (13) and (15):

Alg. 1 input	Factor Optimization (13)	Reparametrization adjustment (15)
$i = u \in \mathsf{V}$ $J = \{uv\}$ $uv \in \mathsf{E}$	$\min\limits_{x_u\in X_u}\{\theta_u^\phi(x_u)\}$	$\varDelta^*_{(u,uv)}(x_u) = \min\limits_{x'_u\in X_u}\theta_u^\phi(x'_u) - \theta_u^\phi(x_u)$
$i = uv \in \mathsf{E}$ $J = \{u\}$ $u \in \mathsf{V}$	$\min\{\theta_u^\phi(x_u,x_v)\}$ $(x_u,x_v) \in X_u \times X_v$	$\varDelta^*_{(uv,u)}(x_u) = \min\limits_{x'_{uv}\in X_{uv}}\theta_{uv}^\phi(x'_{uv}) - \min\limits_{x_v\in X_v}\{\theta_{uv}(x_u,x_v)\}$

MAP-inference Solvers. In Table 1 we state solvers MSD [74], TRWS [48], SRMP [49] and MPLP [28] as special cases of our framework. Factors are visited in the order they are read in.

Algorithm	Current factor	$J_{receive}$	$J_1 \dot\cup \ldots \dot\cup J_l$	ω
MSD [74]	$u \in \mathsf{V}$	$\mathcal{N}_{\mathbb{G}}(u)$	$\{uv\} \subset \mathcal{N}_{\mathbb{G}}(u)$	$\omega_1, \ldots = {}^1/_{\lvert\mathcal{N}_{\mathbb{G}}(u)\rvert}$
	$uv \in \mathsf{E}$	$\varnothing$	—	—
MPLP [28]	$u \in \mathsf{V}$	$\varnothing$	—	—
	$uv \in \mathsf{E}$	$\{u, v\}$	$\{u\}, \{v\}$	$\omega_1 = {}^1/_2 = \omega_2$
TRWS [48] SRMP [49]			forward pass:	
	$u \in \mathsf{V}$	$\{uv : v \in \mathcal{N}_{\mathsf{G}}(u), v < u\}$	$\{uv\} : v \in \mathcal{N}_{\mathsf{G}}(u), v > u$	$\omega_1, \ldots = {}^1/_{\max(\{v \in \mathcal{N}_{\mathsf{G}}(u) : v > u\}, \{v \in \mathcal{N}_{\mathsf{G}}(u) : v < u\})}$
			backward pass:	
	$u \in \mathsf{V}$	$\{uv : v \in \mathcal{N}_{\mathsf{G}}(u), v > u\}$	$\{uv\} : v \in \mathcal{N}_{\mathsf{G}}(u), v < u$	$\omega_1, \ldots = {}^1/_{\max(\{v \in \mathcal{N}_{\mathsf{G}}(u) : v > u\}, \{v \in \mathcal{N}_{\mathsf{G}}(u) : v < u\})}$
	$uv \in \mathsf{E}$	$\varnothing$	—	—

Table 1. [74, 48, 49, 28] as special cases of Algorithm 2.

Remark 1. *We have only treated the case of unary $\theta_u, u \in \mathsf{V}$ and pairwise potentials $\theta_{uv}, uv \in \mathsf{E}$ here. MPLP [28] and SRMP [49] can be applied to higher order potentials as well, which we do not treat here.SRMP [49] is a generalisation of TRWS [48] to the higher-order case.*

Remark 2. *There are convergent message-passing algorithms such that factors comprise trees [72, 65]. Their analysis is more difficult, hence we omit it here.*

Note that our framework generalizes upon [48, 49, 28, 74, 65, 72] in several ways: (i) Our factors need not be simplices or trees. (ii) Our messages need not be marginalization between unary/pairwise/triplet/. . . factors. (iii) We can compute message updates on more than one coupling constraint simultaneously, i.e. we may choose $J_1 \dot\cup \ldots \dot\cup J_l$ in Algorithm 2 to be different than singleton sets. (i) and (ii) affect LP-modeling, (iii) affects computational efficiency: By considering multiple messages at once in Procedure 1, we may be able to make larger updates Δ^*, resulting in faster convergence.