A new look at reweighted message passing

Vladimir Kolmogorov
vnk@ist.ac.at

Abstract

We propose a new family of message passing techniques for MAP estimation in graphical models which we call *Sequential Reweighted Message Passing* (SRMP). Special cases include well-known techniques such as *Min-Sum Diffusion* (MSD) and a faster *Sequential Tree-Reweighted Message Passing* (TRW-S). Importantly, our derivation is simpler than the original derivation of TRW-S, and does not involve a decomposition into trees. This allows easy generalizations. The new family of algorithms can be viewed as a generalization of TRW-S from pairwise to higher-order graphical models. We test SRMP on several real-world problems with promising results.

1 Introduction

This paper is devoted to the problem of minimizing a function of discrete variables represented as a sum of *factors*, where a factor is a term depending on a certain subset of variables. The problem is also known as MAP-MRF inference in a graphical model. Due to the generality of the definition, it has applications in many areas. Probably, the most well-studied case is when each factor depends on at most two variables (*pairwise MRFs*). Many inference algorithms have been proposed. A prominent approach is to try to solve a natural linear programming (LP) relaxation of the problem, sometimes called *Schlesinger LP* [1]. A lot of research went into developing efficient solvers for this special LP, as detailed below.

One of the proposed techniques is *Min-Sum Diffusion* (MSD) [1]. It has a very short derivation, but the price for this simplicity is efficiency: MSD can be significantly slower than more advanced techniques such as *Sequential Tree-Reweighted Message Passing* (TRW-S) [2]. The derivation of TRW-S in [2] uses additionally a decomposition of the graph into trees (as in [3]), namely into *monotonic chains*. This makes generalizing TRW-S to other cases harder (compared to MSD).

We consider a simple modification of MSD which we call *Anisotropic MSD*; it is equivalent to a special case of the *Convex Max-Product* (CMP) algorithm [4]. We then show that with a particular choice of weights and the order of processing nodes Anisotropic MSD becomes equivalent to TRW-S (in the case of pairwise graphical models). This gives an alternative derivation of TRW-S that does involve a decomposition into chains, and allows an almost immediate generalization of TRW-S to higher-order graphical models.

Note that generalized TRW-S has been recently presented in [5]. However, we argue that their generalization is more complicated: it introduces more notation and definitions related to a decomposition of the graphical model into *monotonic junction chains*, imposes weak assumptions on the graph $(\mathcal{F}, J)$ (this graph is defined in the next section), uses some restriction on the order of processing factors, and proposes a special treatment of nested factors to improve efficiency. All this makes generalized TRW-S in [5] more difficult to understand and implement.

We believe that our new derivation may have benefits even for pairwise graphical models. The family of SRMP algorithms is more flexible compared to TRW-S; as discussed in the conclusions, this may prove to be useful in certain scenarios.

Related work Besides [5], the closest related works are probably [6] and [4]. The first one presented a generalization of pairwise MSD to higher-order graphical models, and also described a family of LP relaxations specified by a set of pairs of nested factors for which the marginalization constraint needs to be enforced. We use this framework in our paper. The work [4] presented a family of *Convex Message Passing* algorithms, which we use as one of our building blocks.

Another popular message passing algorithm is *MPLP* [7, 8, 9]. Like MSD, it has a simple formulation (we give it in section 3 alongside with MSD). However, our tests indicate that MPLP can be significantly slower than SRMP.

Algorithms discusses so far perform a block-coordinate ascent on the objective function (and may get stuck in a suboptimal point [2, 1]). Many other techniques with similar properties have been proposed, e.g. [10, 11, 12, 4, 13, 14].

A lot of research also went into developing algorithms that are guaranteed to converge to an optimal solution of the LP. Examples include subgradient techniques [15, 16, 17, 18], smoothing the objective with a temperature parameter that gradually goes to zero [19], proximal projections [20], Nesterov schemes [21, 22], an augmented Lagrangian method [23, 24], a proximal gradient method [25] (formulated for the general LP in [26]), a bundle method [27], a mirror descent method [28], and a "smoothed version of TRW-S" [29].

2 Background and notation

We closely follow the notation of Werner [6]. Let V be the set of nodes. For node $v \in V$ let $\mathcal{X}_v$ be the finite set of possible labels for v. For a subset $\alpha \subseteq V$ let $\mathcal{X}_\alpha = \otimes_{v\in\alpha}\mathcal{X}_v$ be the set of labelings of α, and let $\mathcal{X} = \mathcal{X}_V$ be the set of labelings of V. Our goal is to minimize the function

$$f(\boldsymbol{x} \mid \bar\theta) = \sum_{\alpha\in\mathcal{F}} \bar\theta_\alpha(\boldsymbol{x}_\alpha)\,, \quad \boldsymbol{x} \in \mathcal{X} \tag{1}$$

where $\mathcal{F} \subset 2^V$ is a set of non-empty subsets of V (also called *factors*), $\boldsymbol{x}_\alpha$ is the restriction of $\boldsymbol{x}$ to $\alpha \subseteq V$, and $\bar\theta$ is a vector with components $(\bar\theta_\alpha(\boldsymbol{x}_\alpha) \mid \alpha \in \mathcal{F}, \boldsymbol{x}_\alpha \in \mathcal{X}_\alpha)$.

Let J be a fixed set of pairs of the form (α, β) where $\alpha, \beta \in \mathcal{F}$ and $\beta \subset \alpha$. Note that $(\mathcal{F}, J)$ is a directed acyclic graph. We will be interested in solving the following relaxation of the problem:

$$\min_{\mu\in\mathcal{L}(J)} \sum_{\alpha\in\mathcal{F}}\sum_{\boldsymbol{x}_\alpha} \bar\theta_\alpha(\boldsymbol{x}_\alpha)\mu_\alpha(\boldsymbol{x}_\alpha) \tag{2}$$

where $\mu_\alpha(\boldsymbol{x}_\alpha) \in \mathbb{R}$ are the variables and $\mathcal{L}(J)$ is the J-based *local polytope* of $(V, \mathcal{F})$:

$$\mathcal{L}(J) = \left\{ \mu \geq 0 \;\middle|\; \begin{array}{l} \sum\limits_{\boldsymbol{x}_\alpha} \mu_\alpha(\boldsymbol{x}_\alpha) = 1 \qquad \forall\alpha \in \mathcal{F}, \boldsymbol{x}_\alpha \\ \sum\limits_{\boldsymbol{x}_\alpha : \boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \mu_\alpha(\boldsymbol{x}_\alpha) = \mu_\beta(\boldsymbol{x}_\beta) \\ \hfill \forall(\alpha,\beta) \in J, \boldsymbol{x}_\beta \end{array} \right\} \tag{3}$$

We use the following implicit restriction convention: for $\beta \subseteq \alpha$, whenever symbols $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}_\beta$ appear in a single expression they do not denote independent joint states but $\boldsymbol{x}_\beta$ denotes the restriction of $\boldsymbol{x}_\alpha$ to nodes in β. Sometimes we will emphasize this fact by writing $\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta$, as in the eq. (3).

An important case that is frequently used is when $|\alpha| \leq 2$ for all $\alpha \in \mathcal{F}$ (a *pairwise* graphical model). We will always assume that in this case $J = \{(\{i,j\},\{i\}), (\{i,j\},\{j\}) \mid \{i,j\} \in \mathcal{F}\}$.

When there are higher-order factors, one could define $J = \{(\alpha, \{i\}) \mid i \in \alpha \in \mathcal{F}, |\alpha| \geq 2\}$. Graph $(\mathcal{F}, J)$ is then known as a *factor graph*, and the resulting relaxation is sometimes called the *Basic LP relaxation* (BLP). It is known that this relaxation is tight if each term $\bar\theta_A$ is a submodular function [6]. A larger classes of functions that can be solved with BLP has been recently identified in [30, 31], who in fact completely characterized classes of *Valued Constraint Satisfaction Problems* for which the BLP relaxation is always tight.

For many practical problems, however, the BLP relaxation is not tight; then we can add extra edges to J to tighten the relaxation.

Reparameterization and dual problem For each $(\alpha, \beta) \in J$ let $m_{\alpha\beta}$ be a *message* from α to β; it's a vector with components $m_{\alpha\beta}(\boldsymbol{x}_\beta)$ for $\boldsymbol{x}_\beta \in \mathcal{X}_\beta$. Each message vector $m = (m_{\alpha\beta} | (\alpha, \beta) \in J)$ defines a new vector $\theta = \bar\theta[m]$ according to

$$\theta_\beta(\boldsymbol{x}_\beta) = \bar\theta_\beta(\boldsymbol{x}_\beta) + \sum_{(\alpha,\beta)\in I_\beta} m_{\alpha\beta}(\boldsymbol{x}_\beta) - \sum_{(\beta,\gamma)\in O_\beta} m_{\beta\gamma}(\boldsymbol{x}_\gamma) \tag{4}$$

where I_β and O_β denote respectively the set of incoming and outgoing edges for β:

$$I_\beta = \{(\alpha, \beta) \in J\} \qquad O_\beta = \{(\beta, \gamma) \in J\} \tag{5}$$

It is easy to check that $\bar\theta$ and θ define the same objective function, i.e. $f(\boldsymbol{x} \mid \bar\theta) = f(\boldsymbol{x} \mid \theta)$ for all labelings $\boldsymbol{x} \in \mathcal{X}$. Vector θ that satisfies such condition is called a *reparameterization* of $\bar\theta$ [3].

For each vector θ expression $\Phi(\theta) = \sum\limits_{\alpha\in\mathcal{F}} \min\limits_{\boldsymbol{x}_\alpha} \theta_\alpha(\boldsymbol{x}_\alpha)$ gives a lower bound on $\min_{\boldsymbol{x}} f(\boldsymbol{x} \mid \theta)$. For a vector of messages m let us define $\Phi(m) = \Phi(\theta[m])$; as follows from above, it is a lower bound on energy (1). To obtain the tightest bound, we need to solve the following problem:

$$\max_m \Phi(m) \tag{6}$$

It can be checked that this maximization problem is equivalent to the dual of (2) (see [6]).

3 Block-coordinate ascent

To maximize lower bound $\Phi(m)$, we will use a block-coordinate ascent strategy: select a subset of edges $J' \subseteq J$ and maximize $\Phi(m)$ over messages $(m_{\alpha\beta} | (\alpha, \beta) \in \widetilde{J})$ while keeping all other messages fixed. It is not difficult to show that such restricted maximization problem can be solved efficiently if graph $(\mathcal{F}, J')$ is a tree (or a forest); for pairwise graphical models this was shown in [11]. In this paper we restrict our attention to two special cases of star-shaped trees:

I. Take $J' = I'_\beta \subseteq I_\beta$ for a fixed factor $\beta \in \mathcal{F}$, i.e. (a subset of) incoming edges to β.

II. Take $J' = O'_\alpha \subseteq O_\alpha$ for a fixed factor $\alpha \in \mathcal{F}$, i.e. (a subset of) outgoing edges from α.

We will mostly focus on the case when we take **all** incoming or outgoing edges, but for generality we also allow proper subsets. The two procedures described below are special cases of the *Convex Max-Product* (CMP) algorithm [4] but formulated in a different way: the presentation in [4] did not use the notion of a reparameterization.

Case I: Anisotropic MSD Consider factor $\beta \in \mathcal{F}$ and a non-empty set of incoming edges $I'_\beta \subseteq I_\beta$. A simple algorithm for maximizing $\Phi(m)$ over messages in I'_β is *Min-Sum Diffusion* (MSD). For pairwise models MSD was discovered by Kovalevsky and Koval in the 70's and independently by Flach in the 90's (see [1]). Werner [6] then generalized it to higher-order relaxations.

We will consider a generalization of this algorithm which we call *Anisotropic MSD* (AMSD). It is given in Fig. 1(a). In the first step it computes marginals for parents α of β and "moves" them to factor β; we call it a *collection* step. It then "propagates" obtained vector θ_β back to the parents with weights $\omega_{\alpha\beta}$. Here ω is some probability distribution over $I'_\beta \cup \{\beta\}$, i.e. a non-negative vector with $\sum_{(\alpha,\beta)\in I'_\beta} \omega_{\alpha\beta} + \omega_\beta = 1$. If $\omega_\beta = 0$ then vector θ_β will become zero, otherwise some "mass" will be kept at β

(a) $\texttt{AMSD}(\beta, I'_\beta, \omega)$

1. For each $(\alpha,\beta) \in I'_\beta$ compute

$$\delta_{\alpha\beta}(\boldsymbol{x}_\beta) = \min_{\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \theta_\alpha(\boldsymbol{x}_\alpha) \tag{7}$$

and set

$$\theta_\alpha(\boldsymbol{x}_\alpha) \;:=\; \theta_\alpha(\boldsymbol{x}_\alpha) - \delta_{\alpha\beta}(\boldsymbol{x}_\beta) \tag{8a}$$
$$\theta_\beta(\boldsymbol{x}_\beta) \;:=\; \theta_\beta(\boldsymbol{x}_\beta) + \delta_{\alpha\beta}(\boldsymbol{x}_\beta) \tag{8b}$$

2. Set $\widehat{\theta}_\beta := \theta_\beta$. For each $(\alpha,\beta) \in I'_\beta$ set

$$\theta_\alpha(\boldsymbol{x}_\alpha) \;:=\; \theta_\alpha(\boldsymbol{x}_\alpha) + \omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}_\beta) \tag{9a}$$
$$\theta_\beta(\boldsymbol{x}_\beta) \;:=\; \theta_\beta(\boldsymbol{x}_\beta) - \omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}_\beta) \tag{9b}$$

(b) $\texttt{AMPLP}(\alpha, O'_\alpha, \rho)$

1. For each $(\alpha,\beta) \in O'_\alpha$ set

$$\theta_\alpha(\boldsymbol{x}_\alpha) \;:=\; \theta_\alpha(\boldsymbol{x}_\alpha) + \theta_\beta(\boldsymbol{x}_\beta) \tag{10a}$$
$$\theta_\beta(\boldsymbol{x}_\beta) \;:=\; 0 \tag{10b}$$

2. Set $\widehat{\theta}_\alpha := \theta_\alpha$. For each $(\alpha,\beta) \in O'_\alpha$ compute

$$\delta_{\alpha\beta}(\boldsymbol{x}_\beta) = \min_{\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) \tag{11}$$

and set

$$\theta_\alpha(\boldsymbol{x}_\alpha) \;:=\; \theta_\alpha(\boldsymbol{x}_\alpha) - \rho_{\alpha\beta}\delta_{\alpha\beta}(\boldsymbol{x}_\beta) \tag{12a}$$
$$\theta_\beta(\boldsymbol{x}_\beta) \;:=\; \rho_{\alpha\beta}\delta_{\alpha\beta}(\boldsymbol{x}_\beta) \tag{12b}$$

Figure 1: **Anisotropic MSD and MPLP updates for factors β and α respectively.** *ω is a probability distribution over $I'_\beta \cup \{\beta\}$, and ρ is a probability distribution over $O'_\alpha \cup \{\alpha\}$. All updates should be done for all possible $\boldsymbol{x}_\alpha$, $\boldsymbol{x}_\beta$ with $\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta$.*

(namely, $\omega_\beta \widehat{\theta}_\beta$). [1] The following fact can easily be shown (see Appendix A).

Proposition 1 *Procedure* $\texttt{AMSD}(\beta, I'_\beta, \omega)$ *maximizes* $\Phi(m)$ *over* $(m_{\alpha\beta} \mid (\alpha,\beta) \in I'_\beta)$.

If I'_β contains a single edge $\{(\alpha,\beta)\}$ and ω is a uniform distribution over $I'_\beta \cup \{\beta\}$ (i.e. $\omega_{\alpha\beta} = \omega_\beta = \frac{1}{2}$) then the procedure becomes equivalent to MSD. [2] If $I'_\beta = I_\beta$ then the procedure becomes equivalent to the version of CMP described in [4, Algorithm 5] (for the case when $(\mathcal{F}, J)$ is a factor graph; we need to take $\omega_{\alpha\beta} = c_\alpha / \hat{c}_\beta$).

The work [4] used a fixed distribution ω for each factor. We will show, however, that allowing non-fixed distributions may lead to significant gains in the performance. As we will see in section 4, a particular scheme together with a particular order of processing factors will correspond to the TRW-S algorithm [2] (in the case of pairwise models), which is often faster than MSD/CMP.

Case II: Anisotropic MPLP Let us now consider factor $\alpha \in \mathcal{F}$ and a non-empty set of outgoing edges $O'_\alpha \subseteq O_\alpha$. This case can be tackled using the *MPLP* algorithm [7, 9].

Analogously to Case I, it can be generalized to *Anisotropic MPLP* (AMPLP) - see Figure 1(b). In the first step ("collection") vectors θ_β for children β of α are "moved" to factor α. We then compute min-marginals of α and "propagate" them to children β with weights $\rho_{\alpha\beta}$. Here ρ is some probability distribution over $O'_\alpha \cup \{\alpha\}$. The following can easily be shown (see Appendix B).

Proposition 2 *Procedure* $\texttt{AMPLP}(\beta, O'_\alpha, \rho)$ *maximizes* $\Phi(m)$ *over* $(m_{\alpha\beta} \mid (\alpha,\beta) \in O'_\alpha)$.

The updates given in [7, 8, 9] correspond to $\texttt{AMPLP}(\beta, O_\alpha, \rho)$ where ρ is a uniform probability distribution over O_α (with $\rho_\alpha = 0$). By analogy with Case I, we conjecture that a different weighting (that depends on the order of processing factors) could lead to faster convergence. However, we leave this as a question for future research, and focus on Case I instead.

For completeness, in Appendix C we give an implementation of AMPLP via messages; it is slightly different from implementations in [7, 8, 9] since we store explicitly vectors θ_β for factors β that have at least one incoming edge $(\alpha,\beta) \in J$.

4 Sequential Reweighted Message Passing

In this section we consider a special case of anisotropic MSD updates which we call a *Sequential Reweighted Message Passing* (SRMP). To simplify the presentation, we will assume that $|O_\alpha| \neq 1$ for all $\alpha \in \mathcal{F}$. (This is not a severe restriction: if there is factor α with a single child β then we can reparameterize θ to get $\min_{\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \theta_\alpha(\boldsymbol{x}_\alpha) = 0$ for all $\boldsymbol{x}_\beta$, and then remove factor α; this will not affect the relaxation.)

Let $\mathcal{S} \subset \mathcal{F}$ be the set of factors that have at least one incoming edge. Let us select some total order $\preceq$ on $\mathcal{S}$. SRMP will alternate between a forward pass (processing factors $\beta \in \mathcal{S}$ in the order $\preceq$) and a backward pass (processing these factors in the reverse order), with $I'_\beta = I_\beta$. Next, we discuss how we select distributions ω over $I_\beta \cup \{\beta\}$ for $\beta \in \mathcal{S}$. We will use different distributions during forward and backward passes; they will be denoted as ω^+ and ω^- respectively.

[1] Alternatively, update $\texttt{AMSD}(\beta, I'_\beta, \omega)$ can be defined as follows: reparameterize vectors θ_β and $\{\theta_\alpha \mid (\alpha,\beta) \in I'_\beta\}$ to get

$$\begin{aligned} \theta_\beta(\boldsymbol{x}_\beta) &= \omega_\beta \widehat{\theta}_\beta(\boldsymbol{x}_\beta) \\ \min_{\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \theta_\alpha(\boldsymbol{x}_\alpha) &= \omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}_\beta) \qquad \forall (\alpha,\beta) \in I'_\beta \end{aligned}$$

for some vector $\widehat{\theta}_\beta$.

[2] MSD algorithm given in [6] updates just a single message $m_{\alpha\beta}$ for some edge $(\alpha,\beta) \in J$; this corresponds to $\texttt{AMSD}(\beta, I'_\beta, \omega)$ with $|I'_\beta| = 1$. If $I'_\beta = I_\beta$ then $\texttt{AMSD}(\beta, I'_\beta, \omega)$ with the uniform distribution ω is equivalent to performing single-edge MSD updates until convergence. Such version of MSD for pairwise energies was mentioned in [1, Remark 4], although without an explicit formula.

Let I_β^+ be the set of edges $(\alpha,\beta)\in I_\beta$ such that α is accessed after calling $\mathtt{AMSD}(\beta, I_\beta, \omega^+)$ in the forward pass and before calling $\mathtt{AMSD}(\beta, I_\beta, \omega^-)$ in the subsequent backward pass. Formally,

$$I_\beta^+ = \left\{ (\alpha,\beta)\in J \;\middle|\; \begin{array}{l} (\alpha\in\mathcal{S} \text{ AND } \alpha \succ \beta) \;\; \text{OR} \\ (\exists (\alpha,\gamma)\in J \text{ s.t. } \gamma\succ\beta) \end{array} \right\} \tag{13a}$$

Similarly, let O_β^+ be the set of edges $(\beta,\gamma)\in J$ such as γ is processed after β in the forward pass:

$$O_\beta^+ \;\;=\;\; \{(\beta,\gamma)\in J \,|\, \gamma\succ\beta\} \tag{13b}$$

(Note that $\gamma\in\mathcal{S}$ since γ has an incoming edge, so the comparison $\gamma\succ\beta$ is valid.) We propose the following formula as the default weighting for SRMP in the forward pass:

$$\omega_{\alpha\beta}^+ = \begin{cases} \frac{1}{|O_\beta^+| + \max\{|I_\beta^+|, |I_\beta - I_\beta^+|\}} & \text{if } (\alpha,\beta)\in I_\beta^+ \\ 0 & \text{if } (\alpha,\beta)\in I_\beta - I_\beta^+ \end{cases} \tag{14}$$

It can be checked that the weight $\omega_\beta^+ = 1 - \sum_{(\alpha,\beta)\in I_\beta}\omega_{\alpha\beta}^+$ is non-negative, so this is a valid weighting. We define sets I_β^-, O_β^- and weights $\omega_{\alpha\beta}^-$ for the backward pass in a similar way; the only difference to (13), (14) is that "$\succ$" is replaced with "$\prec$":

$$I_\beta^- = \left\{ (\alpha,\beta)\in J \;\middle|\; \begin{array}{l} (\alpha\in\mathcal{S} \text{ AND } \alpha \prec \beta) \;\; \text{OR} \\ (\exists (\alpha,\gamma)\in J \text{ s.t. } \gamma\prec\beta) \end{array} \right\} \tag{15a}$$

$$O_\beta^- \;\;=\;\; \{(\beta,\gamma)\in J \,|\, \gamma\prec\beta\} \tag{15b}$$

$$\omega_{\alpha\beta}^- = \begin{cases} \frac{1}{|O_\beta^-| + \max\{|I_\beta^-|, |I_\beta - I_\beta^-|\}} & \text{if } (\alpha,\beta)\in I_\beta^- \\ 0 & \text{if } (\alpha,\beta)\in I_\beta - I_\beta^- \end{cases} \tag{16}$$

Note that $I_\beta = I_\beta^+ \cup I_\beta^-$. Furthermore, in the case of pairwise models sets I_β^+ and I_β^- are disjoint.

Our motivation for the choice (14) is as follows. First of all, we claim that weight $\omega_{\alpha\beta}^+$ for edges $(\alpha,\beta)\in I_\beta - I_\beta^+$ can be set to zero without loss of generality. Indeed, if $\omega_{\alpha\beta}^+ = c > 0$ then we can transform $\omega_{\alpha\beta}^+ := 0$, $\omega_\beta^+ := \omega_\beta^+ + c$ without affecting the behaviour of the algorithm.

We also decided to set weights $\omega_{\alpha\beta}^+$ to the same value for all edges $(\alpha,\beta)\in I_\beta^+$; let us call it λ. We must have $\lambda \le \frac{1}{|I_\beta^+|}$ to guarantee that $\omega_\beta^+ \ge 0$.

If O_β^+ is non-empty then we should leave some "mass" at β, i.e. choose $\lambda < \frac{1}{|I_\beta^+|}$; this is needed for ensuring that we get a *local arc consistency* upon convergence of the lower bound (see section 5.3). This was the reason for adding $|O_\beta^+|$ to the demoninator of the expression in (14).

Expression $\max\{|I_\beta^+|, |I_\beta - I_\beta^+|\}$ in (14) was chosen to make SRMP equivalent to TRW-S in the case of the pairwise models (this equivalence is discussed later in this section).

Remark 1 *Setting $\lambda = \frac{1}{1+|I_\beta|}$ would give the CMP algorithm (with uniform weights ω) that processes factors in $\mathcal{S}$ using forward and backward passes. The resulting weight is usually smaller than the weight in eq.* (14)*.*

We conjecture that generalized TRW-S [5] is also a special case of SRMP. In particular, if $J = \{(\alpha,\{i\}) \mid i \in \alpha \in \mathcal{F}, |\alpha| \ge 2\}$ then setting $\lambda = \frac{1}{\max\{|I_\beta - I_\beta^-|, |I_\beta - I_\beta^+|\}}$ would give GTRW-S with a uniform distribution over junction chains (and assuming that we take the longest possible chains). The resulting weight is the same or smaller than the weight in eq. (14)*.*

Remark 2 *We tried one other choice, namely setting $\lambda = \frac{1}{|I_\beta^+|}$ in the case of pairwise models. This gives the same or larger weights compared to* (14)*. Somewhat surprisingly to us, in our preliminary tests this appeared to perform slightly worse than the choice* (14)*. A possible informal explanation is as follows: operation $\mathtt{AMSD}(\beta, I_\beta, \omega^+)$ sends the mass away from β, and it may never come back. It is thus desirable to keep some mass at β, especially when $|I_\beta^-| \gg |I_\beta^+|$.*

Implementation via messages A standard approach for implementing message passing algorithms is to store original vectors $\bar\theta_\alpha$ for factors $\alpha\in\mathcal{F}$ and messages $m_{\alpha\beta}$ for edges $(\alpha,\beta)\in J$ that define current reparameterization $\theta = \bar\theta[m]$ via eq. (4). This leads to Algorithm 1. As in Fig. 1, all updates should be done for all possible $\boldsymbol{x}_\alpha$, $\boldsymbol{x}_\beta$ with $\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta$. As usual, for numerical stability messages can be normalized by an additive constant so that $\min_{\boldsymbol{x}_\beta} m_{\alpha\beta}(\boldsymbol{x}_\beta) = 0$ for all $(\alpha,\beta)\in J$; this does not affect the behaviour of the algorithm.

Algorithm 1 Sequential Reweighted Message Passing (SRMP).

```
initialization: set $m_{\alpha\beta}(\boldsymbol{x}_\beta) = 0$ for all $(\alpha,\beta)\in J$
repeat until some stopping criterion
    for each factor $\beta\in\mathcal{S}$ do in the order $\preceq$
        for each edge $(\alpha,\beta)\in I_\beta^-$ update $m_{\alpha\beta}(\boldsymbol{x}_\beta) :=$
            $$\min_{\boldsymbol{x}_\alpha\sim\boldsymbol{x}_\beta}\left\{\bar\theta_\alpha(\boldsymbol{x}_\alpha) + \sum_{(\gamma,\alpha)\in I_\alpha} m_{\gamma\alpha}(\boldsymbol{x}_\alpha) - \sum_{(\alpha,\gamma)\in O_\alpha, \gamma\ne\beta} m_{\alpha\gamma}(\boldsymbol{x}_\gamma)\right\} \tag{17}$$
        compute $\theta_\beta(\boldsymbol{x}_\beta) =$
            $$\bar\theta_\beta(\boldsymbol{x}_\beta) + \sum_{(\alpha,\beta)\in I_\beta} m_{\alpha\beta}(\boldsymbol{x}_\beta) - \sum_{(\beta,\gamma)\in O_\beta} m_{\beta\gamma}(\boldsymbol{x}_\gamma)$$
        for each edge $(\alpha,\beta)\in I_\beta^+$ update
            $m_{\alpha\beta}(\boldsymbol{x}_\beta) := m_{\alpha\beta}(\boldsymbol{x}_\beta) - \omega_{\alpha\beta}^+\theta_\beta(\boldsymbol{x}_\beta)$
    end for
    reverse ordering $\preceq$, swap $I_\beta^+ \leftrightarrow I_\beta^-$, $\omega_{\alpha\beta}^+ \leftrightarrow \omega_{\alpha\beta}^-$
end repeat
```

Note that update (17) is performed only for edges $(\alpha,\beta)\in I_\beta^-$ while step 1 in $\mathtt{AMSD}(\beta, I_\beta, \omega^+)$ requires updates for edges $(\alpha,\beta)\in I_\beta$. This discrepancy is justified by the proposition below.

Proposition 3 *Starting with the second pass, update* (17) *for edge $(\alpha,\beta)\in I_\beta - I_\beta^-$ would not change vector $m_{\alpha\beta}$.*

Proof. We will refer to this update of β as "update II", and to the previous update of β during the preceding backward pass as "update I". For each edge $(\alpha,\gamma)\in J$ with $\gamma\neq\beta$ we have $\gamma\succ\beta$ (since $(\alpha,\beta)\notin I_\beta^-$), therefore such γ will not be processed between updates I and II. Similarly, if $\alpha\in\mathcal{S}$ then $\alpha\succ\beta$ (again, since $(\alpha,\beta)\notin I_\beta^-$), so α also will not be processed. Therefore, vector θ_α is never modified between updates I and II. Immediately after update I we have $\min\limits_{\boldsymbol{x}_\alpha\sim\boldsymbol{x}_\beta}\theta_\alpha(\boldsymbol{x}_\alpha)=0$ for all $\boldsymbol{x}_\beta$, and so vector $\delta_{\alpha\beta}$ in (7) during update II would be zero. This implies the claim. □

For pairwise graphical models skipping unnecessary updates reduces the amount of computation by approximately a factor of 2. Note that the argument of the proposition does not apply to the very first forward pass of Algorithm 1. Therefore, this pass is not equivalent to AMSD updates, and the lower bound may potentially decrease during the first pass.

Alternative implementation At the first glance Algorithm 1 may appear to be different from the TRW-S algorithm [2] (in the case of pairwise models). For example, if the graph is a chain then after the first iteration messages in TRW-S will converge, while in SRMP they will keep changing (in general, they will be different after forward and backward passes). To show a connection to TRW-S, we will describe an alternative implementation of SRMP with the same update rules as in TRW-S. We will assume that $|\alpha|\leq 2$ for $\alpha\in\mathcal{F}$ and $J=\{(\{i,j\},\{i\}),(\{i,j\},\{j\})\,|\,\{i,j\}\in\mathcal{F}\}$.

The idea is to use messages $\widehat{m}_{(\alpha,\beta)}$ for $(\alpha,\beta)\in J$ that have a different intepretation. Current reparameterization θ will be determined from $\bar{\theta}$ and $\widehat{m}$ using a two-step procedure: (i) compute $\widehat{\theta}=\bar{\theta}[\widehat{m}]$ via eq. (4); (ii) compute

$$\theta_\alpha(\boldsymbol{x}_\alpha)=\widehat{\theta}_\alpha(\boldsymbol{x}_\alpha)+\sum_{(\alpha,\beta)\in O_\alpha}\omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}_\beta)\qquad\forall\alpha\in\mathcal{F},|\alpha|=2$$

$$\theta_\beta(\boldsymbol{x}_\beta)=\omega_\beta\widehat{\theta}_\beta(\boldsymbol{x}_\beta)\qquad\forall\beta\in\mathcal{F},|\beta|=1$$

where $\omega_{\alpha\beta},\omega_\beta$ are the weights used in the last update for β. Update rules with this interpretation are given in Appendix D; if the weights are chosen as in (14) and (16) then these updates are equivalent to those in [2].

Extracting primal solution We used the following scheme for extracting a primal solution $\boldsymbol{x}$. In the beginning of a forward pass we mark all nodes $i\in V$ as "unlabeled". Now consider procedure `AMSD`(β,I_β,ω^+) (lines 4-6 in Algorithm 1). We assign labels to all nodes in $i\in\beta$ as follows: (i) for each $(\alpha,\beta)\in I_\beta$ compute "restricted" messages $m^\star_{\alpha\beta}(\boldsymbol{x}_\beta)$ using the following modification of eq. (17): instead of minimizing over all labelings $\boldsymbol{x}_\alpha\sim\boldsymbol{x}_\beta$, we minimize only over those labelings $\boldsymbol{x}_\alpha$ that are consistent with currently labeled nodes $i\in\alpha$; (ii) compute $\theta^\star_\beta(\boldsymbol{x}_\beta)=\bar{\theta}_\beta(\boldsymbol{x}_\beta)+\sum_{(\alpha,\beta)\in I_\beta}m^\star_{\alpha\beta}(\boldsymbol{x}_\beta)$ for labelings $\boldsymbol{x}_\beta$ consistent with currently labeled nodes $i\in\beta$, and choose a labeling with the smallest cost $\theta^\star_\beta(\boldsymbol{x}_\beta)$. It can be shown that for pairwise graphical models this procedure is equivalent to the one given in [2].

We use the same procedure in the backward pass. We observed that a forward pass usually produces the same labeling as the previous forward pass (and similarly for backward passes), but forward and backward passes often given different results. Accordingly, we run this extraction procedure every third iteration in both passes, and keep track of the best solution found so far. (We implemented a similar procedure for MPLP, but it performed worse than the method in [32] - see Fig. 2(a,g).)

Order of processing factors An important question is how to choose the order $\preceq$ on factors in $\mathcal{S}$. Assume that nodes in V are totally ordered: $V=\{1,\ldots,n\}$. We used the following rule for factors $\alpha,\beta\subseteq V$ proposed in [5]: (i) first sort by the minimum node in α and β; (ii) if $\min\alpha=\min\beta$ then sort by the maximum node. For the remaining cases we added some arbitrarily chosen rules.

Thus, the only parameter to SRMP is the order of nodes. The choice of this order is an important issue which is not addressed in this paper. Note, however, that in many applications there is a natural order on nodes which often works well. In all of our tests we processed the nodes in the order they were given.

5 J-consistency and convergence properties

It is known that fixed points of the MSD algorithm on graph $(\mathcal{F},J)$ are characterized by the *local arc consistency* condition w.r.t. J, or *J-consistency* for short. In this section we show a similar property for SRMP.

We will work with relations $\mathcal{R}_\alpha\subseteq\mathcal{X}_\alpha$. For two relations $\mathcal{R}_\alpha,\mathcal{R}_{\alpha'}$ of factors α,α' with $\alpha\subset\alpha'$ or $\alpha\supset\alpha'$ we denote

$$\pi_{\alpha'}(\mathcal{R}_\alpha)=\{\boldsymbol{x}_{\alpha'}\,|\,\exists\boldsymbol{x}_\alpha\in\mathcal{R}_\alpha\text{ s.t. }\boldsymbol{x}_{\alpha'}\sim\boldsymbol{x}_\alpha\}\tag{18}$$

If $\alpha'\subset\alpha$ then $\pi_{\alpha'}(\mathcal{R}_\alpha)$ is usually called a *projection* of $\mathcal{R}_\alpha$ to α'. For a vector $\theta_\alpha=(\theta_\alpha(\boldsymbol{x}_\alpha)\mid\boldsymbol{x}_\alpha\in\mathcal{X}_\alpha)$ we define relation $\langle\theta_\alpha\rangle\subseteq\mathcal{X}_\alpha$ via

$$\langle\theta_\alpha\rangle=\arg\min_{\boldsymbol{x}_\alpha}\theta_\alpha(\boldsymbol{x}_\alpha)\tag{19}$$

Definition 4 *Vector θ is said to be J-consistent if there exist non-empty relations $(\mathcal{R}_\beta\subseteq\langle\theta_\beta\rangle\mid\beta\in\mathcal{F})$ such that $\pi_\beta(\mathcal{R}_\alpha)=\mathcal{R}_\beta$ for each $(\alpha,\beta)\in J$.*

The main result of this section is the following theorem; it shows that J-consistency is a natural stopping criterion for SRMP.

Theorem 5 *Let $\theta^t=\bar{\theta}[m^t]$ be the vector produced after t iterations of the SRMP algorithm, with $\theta^0=\bar{\theta}$. Then the following holds for $t>0$.* [3]
(a) If θ^t is J-consistent then $\Phi(\theta^{t'})=\Phi(\theta^t)$ for all $t'>t$.
(b) If θ^t is not J-consistent then $\Phi(\theta^{t'})>\Phi(\theta^t)$ for some $t'>t$.
(c) If θ^* is a limit point of the sequence $(\theta^t)_t$ then θ^* is J-consistent (and also $\lim_{t\to\infty}\Phi(\theta^t)=\Phi(\theta^*)$).

[3] The condition $t>0$ is added since updates in the very first iteration of SRMP are not equivalent to AMSD updates, as discussed in the previous section.

Remark 3 *Note that the sequence $(\theta^t)_t$ has at least one limit point θ^* if, for example, vectors θ^t are bounded. We conjecture that these vectors always stay bounded, but leave this as an open question.*

Remark 4 *For other message passing algorithms such as MSD it was conjectured in [1] that the messages m^t converge to a fixed point m^* for $t \to \infty$. We would like to emhpasize that this is not the case for the SRMP algorithm, as discussed in the previous section; in general, the messages would be different after backward and forward passes. In this respect SRMP differs from other proposed message passing techniques such as MSD, TRW-S and MPLP. However, a weaker convergence property given in Theorem 5(c) still holds. (A similar property has been proven for the pairwise TRW-S algorithm [2], except that we do not prove that the vectors stay bounded.)*

The remainder of this section is devoted to the proof of Theorem 5. The proof will be applicable not just to SRMP, but to other sequences of updates $\texttt{AMSD}(\beta, I_\beta, \omega)$ that satisfy certain conditions. The first condition is that the updates consist of the same iteration that is repeatedly applied infinitely many times, and this iteration visits each factor $\beta \in \mathcal{F}$ with $I_\beta \neq \varnothing$ at least once. The second condition concerns zero components of distributions ω; it will hold, in particular, if there are no such components. Details are given below.

5.1 Proof of Theorem 5(a)

The statement is a special case of the following well-known fact: if θ is J-consistent then applying any number of tree-structured block-coordinate ascent steps (such as AMSD and AMPLP) will not increase the lower bound. For completeness, a proof of this fact is given below.

Proposition 6 *Suppose that θ is J-consistent with relations $(\mathcal{R}_\beta \subseteq \langle \theta_\beta \rangle \mid \beta \in \mathcal{F})$, and $J' \subseteq J$ is a subset of edges such that graph $(\mathcal{F}, J')$ is a forest. Applying a block-coordinate ascent step (w.r.t. J') to θ preserves J-consistency (with the same relations $(\mathcal{R}_\beta \mid \beta \in \mathcal{F})$) and does not change the lower bound $\Phi(\theta)$.*

Proof. We can assume w.l.o.g. that $J = J'$: removing edges in $J - J'$ does not affect the claim.

Consider LP relaxation (2). We claim that there exists a feasible vector μ such that $supp(\mu_\alpha) = \mathcal{R}_\alpha$ for all $\alpha \in \mathcal{F}$, where $supp(\mu_\alpha) = \{\boldsymbol{x}_\alpha \mid \mu_\alpha(\boldsymbol{x}_\alpha) > 0\}$ is the support of probability distribution μ_α. Such vector can be constructed as follows. First, for each connected component of $(\mathcal{F}, J)$ we pick an arbitrary factor α in this component and choose some distribution μ_α with $supp(\mu_\alpha) = \mathcal{R}_\alpha$ (e.g. a uniform distribution over $\mathcal{R}_\alpha$). Then we repeatedly choose an edge $(\alpha, \beta) \in J$ with exactly one "assigned" endpoint, and choose a probability distribution for the other endpoint. Namely, if μ_α is assigned then set μ_β via $\mu_\beta(\boldsymbol{x}_\beta) = \sum_{\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \mu_\alpha(\boldsymbol{x}_\alpha)$; we then have $supp(\mu_\beta) = \pi_\beta(supp(\mu_\alpha)) = \pi_\beta(\mathcal{R}_\alpha) = \mathcal{R}_\beta$. If μ_β is assigned then we choose some probability distribution $\hat{\mu}_\alpha$ with $supp(\hat{\mu}_\alpha) = \mathcal{R}_\alpha$, compute its marginal probability $\hat{\mu}_\beta(\boldsymbol{x}_\beta) = \sum_{\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \hat{\mu}_\alpha(\boldsymbol{x}_\alpha)$ and then set $\mu_\alpha(\boldsymbol{x}_\alpha) = \frac{\mu_\beta(\boldsymbol{x}_\beta)}{\hat{\mu}_\beta(\boldsymbol{x}_\beta)} \hat{\mu}_\alpha(\boldsymbol{x}_\alpha)$ for labelings $\boldsymbol{x}_\alpha \in \mathcal{R}_\alpha$; for other labelings $\mu_\alpha(\boldsymbol{x}_\alpha)$ is set to zero. The fact that $\pi_\beta(\mathcal{R}_\alpha) = \mathcal{R}_\beta$ implies that μ_α is a valid probability distribution with $supp(\mu_\alpha) = \mathcal{R}_\alpha$. The claim is proved.

Using standard LP duality for (2), it can be checked that condition $\mathcal{R}_\alpha \subseteq \langle \theta_\alpha \rangle$ for all $\alpha \in \mathcal{F}$ is equivalent to the complementary slackness conditions for vectors μ and $\theta = \bar{\theta}[m]$ (where μ is the vector constructed above and m is the vector of messages corresponding to θ). Therefore, m is an optimal dual vector for (2). This means that applying a block-coordinate ascent step to $\theta = \bar{\theta}[m]$ results in a vector $\theta' = \bar{\theta}[m']$ which is optimal as well: $\Phi(\theta') = \Phi(\theta)$. The complementary slackness conditions must hold for θ', so $\mathcal{R}_\alpha \subseteq \langle \theta'_\alpha \rangle$ for all $\alpha \in \mathcal{F}$. □

5.2 Proof of Theorem 5(b,c)

Consider a sequence of AMSD updates from Fig. 1 where $I'_\beta = I_\beta$. One difficulty in the analysis is that some components of distributions ω may be zeros. We will need to impose some restrictions on such components. Specifically, we will require the following:

R1 *For each call $\texttt{AMSD}(\beta, I_\beta, \omega)$ with $O_\beta \neq \varnothing$ there holds $\omega_\beta > 0$.*

R2 *Consider a call $\texttt{AMSD}(\beta, I_\beta, \omega)$ with $(\alpha, \beta) \in I_\beta$, $\omega_{\alpha\beta} = 0$. This call "locks" factor α, i.e. this factor and its children (except for β) cannot be processed anymore until it is "unlocked" by calling $\texttt{AMSD}(\beta, I_\beta, \omega')$ with $\omega'_{\alpha\beta} > 0$.*

R3 *The updates are applied in iterations, where each iteration calls $\texttt{AMSD}(\beta, I_\beta, \omega)$ for each $\beta \in \mathcal{F}$ with $I_\beta \neq \varnothing$ at least once.*

Restriction R2 can also be formulated as follows. For each factor $\alpha \in \mathcal{F}$ let us keep a variable $\Gamma_\alpha \in O_\alpha \cup \{\varnothing\}$. In the beginning we set $\Gamma_\alpha := \varnothing$ for all $\alpha \in \mathcal{F}$, and after calling $\texttt{AMSD}(\beta, I_\beta, \omega)$ we update these variables as follows: for each $(\alpha, \beta) \in I_\beta$ set $\Gamma_\alpha := (\alpha, \beta)$ if $\omega_{\alpha\beta} = 0$, and $\Gamma_\alpha := \varnothing$ otherwise. Condition R2 means that calling $\texttt{AMSD}(\beta, I_\beta, \omega)$ is possible only if (i) $\Gamma_\beta = \varnothing$, and (ii) for each $(\alpha, \beta) \in I_\beta$ there holds $\Gamma_\alpha \in \{(\alpha, \beta), \varnothing\}$.

It can be seen that the sequence of updates in SRMP (starting with the second pass) satisfies conditions of the theorem; a proof of this fact is similar to that of Proposition 3.

Theorem 7 *Consider a sequence of AMSD updates satisfying conditions R1-R3. If vector θ° is not J-consistent then applying the updates to θ° will increase the lower bound after at most $T = 1 + \sum_{\alpha \in \mathcal{F}} |\mathcal{X}_\alpha| + \sum_{(\alpha,\beta) \in J} |\mathcal{X}_\beta|$ iterations.*

Theorem 7 immediately implies part (b) of Theorem 5. Using an argument from [2], we can also prove part (c) as follows.

Let $(\theta^{t(k)})_k$ be a subsequence of $(\theta^t)_t$ such that $\lim_{k \to \infty} \theta^{t(k)} = \theta^*$. We have

$$\lim_{t \to \infty} \Phi(\theta^t) = \lim_{k \to \infty} \Phi(\theta^{t(k)}) = \Phi(\theta^*) \qquad (20)$$

where the first equality holds since the sequence $(\Phi(\theta^t))_t$ is monotonic, and the second equality is by continuity of function $\Phi : \Omega \to \mathbb{R}$, where by Ω we denoted the space of vectors of the form $\theta = \bar{\theta}[m]$.

We need to show that θ^* is J-consistent. Suppose that this is not the case. Define mapping $\pi : \Omega \to \Omega$ as follows: we take vector $\theta \in \Omega$ and apply T iterations of SRMP to it. By Theorem 7 we have $\Phi(\pi(\theta^*)) > \Phi(\theta^*)$. Clearly, π and thus $\Phi \circ \pi$ are continuous mappings, therefore

$$\lim_{k\to\infty} \Phi(\pi(\theta^{t(k)})) = \Phi\left(\pi\left(\lim_{k\to\infty}\theta^{t(k)}\right)\right) = \Phi(\pi(\theta^*))$$

This implies that $\Phi(\pi(\theta^{t(k)})) > \Phi(\theta^*)$ for some index k. Note that $\pi(\theta^{t(k)}) = \theta^t$ for $t = T + t(k)$. We obtained that $\Phi(\theta^t) > \Phi(\theta^*)$ for some $t > 0$; this contradicts eq. (20) and monotonicity of the sequence $(\Phi(\theta^t))_t$.

We showed that Theorem 7 indeed implies Theorem 5(b,c). It remains to prove Theorem 7.

Remark 5 *Note that while Theorem 5(a,b) holds for any sequence of AMSD updates satisfying conditions R1-R3, we believe that this is not the case for Theorem 5(c). If, for example, the updates for factors β used varying distributions ω whose components $\omega_{\alpha\beta}$ would tend to zero then the increase of the lower bound could become exponentially smaller with each iteration, and $\Phi(\theta^t)$ might not converge to $\Phi(\theta^*)$ for a J-consistent vector θ^*. In the argument above it was essential that the sequence of updates was repeating, and the weights $\omega^+_{\alpha\beta}$, $\omega^-_{\alpha\beta}$ used in Algorithm 1 were kept constant.*

5.3 Proof of Theorem 7

The proof will be based on the following fact.

Lemma 8 *Suppose that operation* AMSD(β, I_β, ω) *does not increase the lower bound. Let θ and θ' be respectively the vector before and after the update, and $\delta_{\alpha\beta}$, $\widehat{\theta}_\beta$ be the vectors defined in steps 1 and 2. Then for any $(\alpha,\beta) \in I_\beta$ there holds*

$$\langle\delta_{\alpha\beta}\rangle = \pi_\beta(\langle\theta_\alpha\rangle) \tag{21a}$$

$$\langle\widehat{\theta}_\beta\rangle = \langle\theta_\beta\rangle \cap \bigcap_{(\alpha,\beta)\in I_\beta} \langle\delta_{\alpha\beta}\rangle \tag{21b}$$

$$\langle\theta'_\alpha\rangle = \langle\theta_\alpha\rangle \cap \pi_\alpha(\langle\widehat{\theta}_\beta\rangle) \quad \textit{if } \omega_{\alpha\beta} > 0 \tag{21c}$$

$$\langle\theta'_\alpha\rangle \cap \pi_\alpha(\langle\widehat{\theta}_\beta\rangle) = \langle\theta_\alpha\rangle \cap \pi_\alpha(\langle\widehat{\theta}_\beta\rangle) \quad \textit{if } \omega_{\alpha\beta} = 0 \tag{21d}$$

Proof. Adding a constant to vectors θ_γ does not affect the claim, so we can assume w.l.o.g. that $\min_{\boldsymbol{x}_\gamma} \theta_\gamma(\boldsymbol{x}_\gamma) = 0$ for all factors γ. This means that $\langle\theta_\gamma\rangle = \{\boldsymbol{x}_\gamma \mid \theta_\gamma(\boldsymbol{x}_\gamma) = 0\}$. We denote $\widehat{\theta}$ to be the vector after the update in step 1. By the assumption of this lemma and by Proposition 1 we have $\Phi(\widehat{\theta}) \le \Phi(\theta') = \Phi(\theta) = 0$.

Eq. (21a) follows directly from the definition of $\delta_{\alpha\beta}$. Also, we have $\min_{\boldsymbol{x}_\beta} \delta_{\alpha\beta}(\boldsymbol{x}_\beta) = \min_{\boldsymbol{x}_\alpha} \widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) = 0$ for all $(\alpha,\beta) \in I_\beta$.

By construction, $\widehat{\theta}_\beta = \theta_\beta + \sum_{(\alpha,\beta)\in I_\beta} \delta_{\alpha\beta}$. All terms of this sum are non-negative vectors, thus vector $\widehat{\theta}_\beta$ is also non-negative. We must have $\min_{\boldsymbol{x}_\beta} \widehat{\theta}_\beta(\boldsymbol{x}_\beta) = 0$, otherwise we would have $\Phi(\widehat{\theta}) > \Phi(\theta)$ - a contradiction. Thus,

$$\begin{aligned}\langle\widehat{\theta}_\beta\rangle &= \{\boldsymbol{x}_\beta \mid \widehat{\theta}_\beta(\boldsymbol{x}_\beta) = 0\} \\ &= \{\boldsymbol{x}_\beta \mid \theta_\beta(\boldsymbol{x}_\beta) = 0 \text{ AND } \delta_{\alpha\beta}(\boldsymbol{x}_\beta) = 0 \;\; \forall(\alpha,\beta) \in I_\beta\}\end{aligned}$$

which gives (21b).

Consider $(\alpha,\beta) \in I_\beta$. By construction, $\theta'_\alpha(\boldsymbol{x}_\alpha) = \widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) + \omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}_\beta)$. We know that (i) vectors θ_α, $\widehat{\theta}_\alpha$ and $\widehat{\theta}_\beta$ are non-negative, and their minina is 0; (ii) there holds $\boldsymbol{x}_\beta \in \langle\widehat{\theta}_\beta\rangle \subseteq \langle\delta_{\alpha\beta}\rangle$, so for each $\boldsymbol{x}_\alpha$ with $\boldsymbol{x}_\beta \in \langle\widehat{\theta}_\beta\rangle$ we have $\delta_{\alpha\beta}(\boldsymbol{x}_\beta) = 0$ and therefore $\widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) = \theta_\alpha(\boldsymbol{x}_\alpha)$. These facts imply (21c) and (21d). □

From now on we assume that applying T iterations to vector θ° does not increase the lower bound; we need to show that θ° is J-consistent.

Let us define relations $\mathcal{R}_\alpha \subseteq \mathcal{X}_\alpha$ for $\alpha \in \mathcal{F}$ and $\mathcal{R}_{\alpha\beta} \in \mathcal{X}_\beta$ for $(\alpha,\beta) \in J$ using the following procedure. In the beginning we set $\mathcal{R}_\alpha = \langle\theta^\circ_\alpha\rangle$ for all $\alpha \in \mathcal{F}$ and $\mathcal{R}_{\alpha\beta} = \mathcal{X}_\beta$ for all $(\alpha,\beta) \in J$. After calling AMSD(β, I_β, ω) we update these relations as follows:

- Set $\mathcal{R}'_\beta := \langle\widehat{\theta}_\beta\rangle$ where $\widehat{\theta}_\beta$ is the vector in step 2 of procedure AMSD(β, I_β, ω).
- For each $(\alpha,\beta) \in I_\beta$ set $\mathcal{R}'_{\alpha\beta} := \langle\widehat{\theta}_\beta\rangle$. If $\omega_{\alpha\beta} > 0$ then set $\mathcal{R}'_\alpha := \langle\theta'_\alpha\rangle$ where θ'_α is the vector after the update, otherwise set $\mathcal{R}'_\alpha := \mathcal{R}_\alpha \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$.

Finally, we update $\mathcal{R}_\beta := \mathcal{R}'_\beta$ and $\mathcal{R}_{\alpha\beta} := \mathcal{R}'_{\alpha\beta}$.

Lemma 9 *The following invariants are preserved for each $\alpha \in \mathcal{F}$ during the first T iterations:*
(a) If $O_\alpha \neq \varnothing$ and $\Gamma_\alpha = \varnothing$ then $\mathcal{R}_\alpha = \langle\theta_\alpha\rangle$.
(b) If $O_\alpha = \varnothing$ then either $\mathcal{R}_\alpha = \langle\theta_\alpha\rangle$ or $\theta_\alpha = \mathbf{0}$.
(c) If $\Gamma_\alpha = (\alpha,\beta)$ then $\mathcal{R}_\alpha = \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}_{\alpha\beta})$.
Also, for each $(\alpha,\beta) \in J$ the following is preserved:
(d) $\mathcal{R}_\beta \subseteq \mathcal{R}_{\alpha\beta}$. If $O_\beta = \varnothing$ then $\mathcal{R}_\beta = \mathcal{R}_{\alpha\beta}$.
(e) $\mathcal{R}_\alpha \subseteq \pi_\alpha(\mathcal{R}_{\alpha\beta})$.
Finally, relations $\mathcal{R}_\alpha$ and $\mathcal{R}_{\alpha\beta}$ either shrink or stay the same, i.e. they never acquire new elements.

Proof. Checking that properties (a)-(e) hold after initialization is straightforward. Let us show that a call to procedure AMSD(β, I_β, ω) preserves them. We use the notation as in Lemma 8. Similarly, we denote $\mathcal{R}_\gamma, \mathcal{R}_{\alpha\beta}, \Gamma_\alpha$ and $\mathcal{R}'_\gamma, \mathcal{R}'_{\alpha\beta}, \Gamma'_\alpha$ to be the corresponding quantities before and after the update.

Monotonicity of $\mathcal{R}_\beta$ and $\mathcal{R}_{\alpha\beta}$. Let us show that $\mathcal{R}'_\beta = \langle\widehat{\theta}_\beta\rangle \subseteq \mathcal{R}_\beta$ and also $\mathcal{R}'_{\alpha\beta} = \langle\widehat{\theta}_\beta\rangle \subseteq \mathcal{R}_{\alpha\beta}$ for all $(\alpha,\beta) \in I_\beta$. By parts (a,b) of the induction hypothesis for factor β two cases are possible:

- $\langle\theta_\beta\rangle = \mathcal{R}_\beta$. Then $\langle\widehat{\theta}_\beta\rangle \subseteq \langle\theta_\beta\rangle = \mathcal{R}_\beta \subseteq \mathcal{R}_{\alpha\beta}$ where the first inclusion is by (21a) and the second inclusion is by the first part of (d).
- $O_\beta = \varnothing$ and $\theta_\beta = \mathbf{0}$. By the second part of (d) we have $\mathcal{R}_\beta = \mathcal{R}_{\alpha\beta}$ for all $(\alpha,\beta) \in I_\beta$, so we just need to show that $\langle\widehat{\theta}_\beta\rangle \subseteq \mathcal{R}_\beta$. There must exist $(\alpha,\beta) \in I_\beta$ with $\Gamma_\alpha = \varnothing$. For such (α,β) we have $\langle\theta_\alpha\rangle = \mathcal{R}_\alpha \subseteq \pi_\alpha(\mathcal{R}_{\alpha\beta})$ by (a) and (e). This implies that $\pi_\beta(\langle\theta_\alpha\rangle) \subseteq \mathcal{R}_{\alpha\beta}$. Finally, $\langle\widehat{\theta}_\beta\rangle \subseteq \langle\delta_{\alpha\beta}\rangle = \pi_\beta(\langle\theta_\alpha\rangle) \subseteq \mathcal{R}_{\alpha\beta} = \mathcal{R}_\beta$

where we used (21b), (21a) and the second part of (d).

Monotonicity of $\mathcal{R}_\alpha$ for $(\alpha,\beta)\in I_\beta$. Let us show that $\mathcal{R}'_\alpha \subseteq \mathcal{R}_\alpha$. If $\omega_{\alpha\beta}=0$ then $\mathcal{R}'_\alpha = \mathcal{R}_\alpha \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$, so the claim holds. Assume that $\omega_{\alpha\beta}>0$. By (21c) we have $\mathcal{R}'_\alpha = \langle\theta'_\alpha\rangle = \langle\theta_\alpha\rangle \cap \pi_\alpha(\langle\widehat{\theta}_\beta\rangle)$. We already showed that $\langle\widehat{\theta}_\beta\rangle = \mathcal{R}'_{\alpha\beta} \subseteq \mathcal{R}_{\alpha\beta}$, therefore $\mathcal{R}'_\alpha \subseteq \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}_{\alpha\beta})$. By (a,c) we have either $\mathcal{R}_\alpha = \langle\theta_\alpha\rangle$ or $\mathcal{R}_\alpha = \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}_{\alpha\beta})$; in each case $\mathcal{R}'_\alpha \subseteq \mathcal{R}_\alpha$.

To summarize, we showed monotonicity for all relations (relations that are not mentioned above do not change).

Invariants (a,b). If $\omega_\beta = 0$ then $\theta'_\beta = \mathbf{0}$ (and $O_\beta = \varnothing$ due to restriction R2). Otherwise $\langle\theta'_\beta\rangle = \langle\omega_\beta\widehat{\theta}_\beta\rangle = \langle\widehat{\theta}_\beta\rangle = \mathcal{R}'_\beta$. In both cases properties (a,b) hold for factor β after the update.

Properties (a,b) also cannot become violated for factors α with $(\alpha,\beta)\in I_\beta$. Indeed, (b) does not apply to such factors, and (a) will apply only if $\omega_{\alpha\beta}>0$, in which case we have $\mathcal{R}' = \langle\theta'_\alpha\rangle$, as required. We proved that (a,b) are preserved for all factors.

Invariant (c). Consider edge $(\alpha,\beta)\in I_\beta$ with $\Gamma'_\alpha = (\alpha,\beta)$ (i.e. with $\omega_{\alpha\beta}=0$). We need to show that $\mathcal{R}'_\alpha = \langle\theta'_\alpha\rangle \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$. By construction, $\mathcal{R}'_\alpha = \mathcal{R}_\alpha \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$, and by (21d) we have have $\langle\theta'_\alpha\rangle \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta}) = \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$ (note that $\mathcal{R}'_{\alpha\beta} = \langle\widehat{\theta}_\beta\rangle$). Thus, we need to show that $\mathcal{R}_\alpha \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta}) = \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$. Two cases are possible:

- $\Gamma_\alpha = \varnothing$. By (a) we have $\mathcal{R}_\alpha = \langle\theta_\alpha\rangle$, which implies the claim.
- $\Gamma_\alpha = (\alpha,\beta)$. By (c) we have $\mathcal{R}_\alpha = \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}_{\alpha\beta})$, so we need to show that $\langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}_{\alpha\beta}) \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta}) = \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$. This holds since $\mathcal{R}'_{\alpha\beta} \subseteq \mathcal{R}_{\alpha\beta}$.

Invariants (d,e). These invariants cannot become violated for edges $(\alpha',\beta')\in J$ with $\beta'\neq\beta$ since relation $\mathcal{R}_{\alpha'\beta'}$ does not change and relations $\mathcal{R}_{\alpha'}, \mathcal{R}_{\beta'}$ do not grow. Checking that that (d,e) are preserved for edges $(\alpha,\beta)\in I_\beta$ is straightforward. We just discuss one case (all other cases follow directly from the construction). Suppose that $\omega_{\alpha\beta}>0$. Then $\mathcal{R}'_\alpha = \langle\theta'_\alpha\rangle = \langle\theta_\alpha\rangle \cap \pi_\alpha(\mathcal{R}'_{\alpha\beta})$ (by (21c)); this implies that $\mathcal{R}'_\alpha \subseteq \pi_\alpha(\mathcal{R}'_{\alpha\beta})$, as desired. □

We are now ready to prove Theorem 7, i.e. that vector θ° is J-consistent. As we showed, all relations never grow, so after fewer than T iterations we will encounter an iteration during which the relations do not change. Let $(\mathcal{R}_\alpha \mid \alpha\in\mathcal{F})$ and $(\mathcal{R}_{\alpha\beta} \mid (\alpha,\beta)\in J)$ be the relations during this iteration. There holds $\mathcal{R}_\alpha \subseteq \langle\theta^\circ_\alpha\rangle$ for all $\alpha\in\mathcal{F}$ (since we had equalities after initialization and then the relations have either shrunk or stayed the same). Consider edge $(\alpha,\beta)\in J$. At some point during the iteration we call `AMSD`(β, I_β, ω). By analyzing this call we conclude that $\mathcal{R}_{\alpha\beta} = \mathcal{R}_\beta$. From (21a), (21b) and Lemma 9(a) we get $\mathcal{R}_\beta = \langle\widehat{\theta}_\beta\rangle \subseteq \langle\delta_{\alpha\beta}\rangle = \pi_\beta(\langle\theta_\alpha\rangle) = \pi_\beta(\mathcal{R}_\alpha)$. From Lemma 9(e) we obtain that $\mathcal{R}_\alpha \subseteq \pi_\alpha(\mathcal{R}_\beta)$.

We showed that $\mathcal{R}_\beta \subseteq \pi_\beta(\mathcal{R}_\alpha)$ and $\mathcal{R}_\alpha \subseteq \pi_\alpha(\mathcal{R}_\beta)$; this implies that $\mathcal{R}_\beta = \pi_\beta(\mathcal{R}_\alpha)$. Finally, relation $\mathcal{R}_\beta$ (and thus $\mathcal{R}_\alpha$) is non-empty since $\mathcal{R}_\beta = \langle\widehat{\theta}_\beta\rangle$.

6 Experimental results

In this section we compare SRMP, CMP (namely, updates from Fig. 1(a) with the uniform distributions ω) and MPLP. Note that there are many other inference algorithms, see e.g. [33] for a recent comprehensive comparison. Our goal is not to replicate this comparison but instead focus on techniques from the same family (namely, coordinate ascent message passing algorithms), since they represent an important branch of MRF optimization algorithms.

We implemented the three methods in the same framework, trying to put the same amount of effort into optimizing each technique. For MPLP we used equations given in Appendix C. On the protein and second-order stereo instances discussed below our implementation was about 10 times faster than the code in [32].[4] For all three techniques factors were processed in the order described in section 4 (but for CMP and MPLP we used only forward passes).

Unless noted otherwise, graph $(\mathcal{F},J)$ was constructed as follows: (i) add to $\mathcal{F}$ all possible intersections of existing factors; (ii) add edges (α,β) to J such that $\alpha,\beta\in\mathcal{F}$, $\beta\subset\alpha$, and there is no "intermediate" factor $\gamma\in\mathcal{F}$ with $\beta\subset\gamma\subset\alpha$. For some problems we also experimented with the BLP relaxation (defined in section 2); although it is weaker in general, message passing operations can potentially be implemented faster for certain factors.

Instances We used the data cited in [34] and a subset of data from [5][5]. These are energies of order 2,3 and 4. Note that for energies of order 2 (i.e. for pairwise energies) SRMP is equivalent to TRW-S; we included this case for completeness. The instances are summarized below.

(a) Potts stereo vision. We took 4 instances from [34]; each node has 16 possible labels.

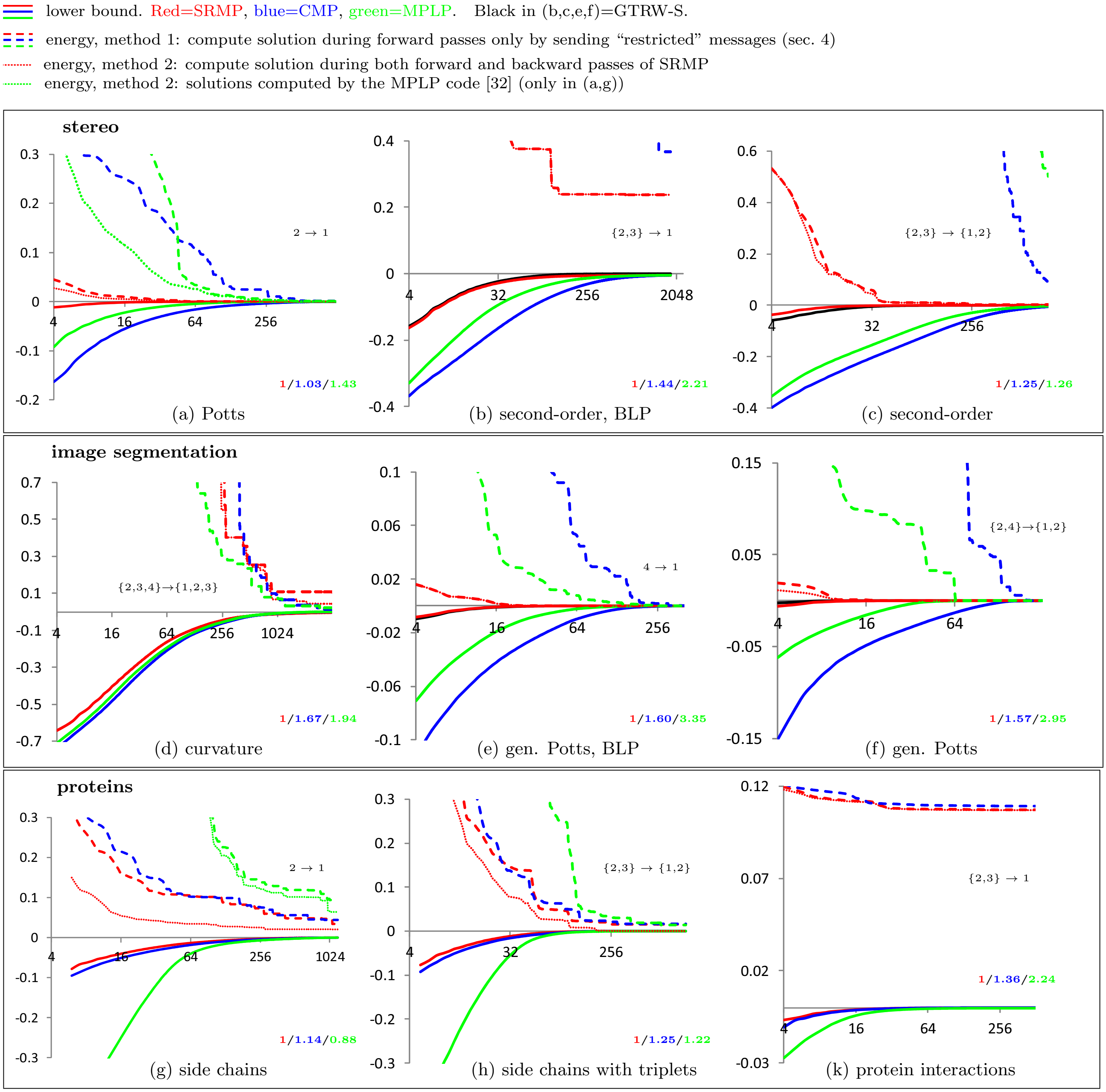
(b,c) Stereo with a second order smoothness prior [35]. We used the version described in [5]; the energy has unary terms and ternary terms in horizontal and vertical directions. We ran it on scaled-down stereo pairs "Tsukuba" and "Venus" (with 8 and 10 labels respectively).

(d) Constraint-based curvature, as described in [5]. Nodes correspond to faces of the dual graph, and have 2 possible labels. We used "cameraman" and "lena" images of size 64×64.

(e,f) Generalized Potts model with 4 labels, as described in [5]. The energy has unary terms and 4-th order terms corresponding to 2×2 patches. We used 3 scaled-down images ("lion", "lena" and "cameraman").

[4] The code in [32] appears to use the same message update routines for all factors. We instead implemented separate routines for different factors (namely Potts, general pairwise and higher-order) as virtual functions in C++. In addition, we precompute the necessary tables for edges $(\alpha,\beta)\in J$ with $|\alpha|\geq 3$ to allow faster computation. We made sure that our code produced the same lower bounds as the code in [32].

[5] We excluded instances with specialized high-order factors used in [5]. They require customized message passing routines, which we have not implemented. As discussed in [5], for such energies MPLP has an advantage; SRMP and CMP could be competitive with MPLP only if the factor allows efficient incremental updates exploiting the fact that after sending message $\alpha\to\beta$ only message $m_{\alpha\beta}$ changes for factor α.



(a) Potts (b) second-order, BLP (c) second-order

(d) curvature (e) gen. Potts, BLP (f) gen. Potts

(g) side chains (h) side chains with triplets (k) protein interactions

Figure 2: Average lower bound and energy vs. time. **X axis**: SRMP iterations in the log scale. Notation **1**/a/b means that 1 iteration of SRMP takes the same time as a iterations of CMP, or b iterations of MPLP. Note that an SRMP iteration has two passes (forward and backward), while CMP and MPLP iterations have only forward passes. However in each pass SRMP updates only a subset of messages (half in the case of pairwise models). **Y axis**: lower bound/energy, normalized so that the initial lower bound (with zero messages) is -1, and the best computed lower bound is 0. Line $A \to B$ gives information about set J: $A = \{|\alpha| \; : \; (\alpha, \beta) \in J\}$, $B = \{|\beta| \; : \; (\alpha, \beta) \in J\}$.

(g) Side chain prediction in proteins. We took 30 instances from [34].

(h) A tighter relaxation of energies in (g) obtained by adding zero-cost triplets of nodes to $\mathcal{F}$. These triplets were generated by the MPLP code [32] that implements the techniques in [36, 37]. [6]

(k) Protein-protein interactions, with binary labels (8 in-

[6] The set of edges J was set in the same way as in the code [32]: for each triplet $\alpha = \{i, j, k\}$ we add to J edges from α to the factor $\{i, j\}$, $\{j, k\}$, $\{i, k\}$. If one of these factors (say, $\{i, j\}$) is not present in the original energy then we did **not** add edges $(\{i, j\}, \{i\})$ and $(\{i, j\}, \{j\})$.

stances from [38]). We also tried reparameterized energies given in [34]; the three methods performed very similarly (not shown here).

Summary of results From the plots in Fig. 2 we make the following conclusions. On problems with a highly regular graph structure (a,b,c,e,f) SRMP clearly outperforms CMP and MPLP. On the protein-related problems (g,h,k) SRMP and CMP perform similarly, and outperform MPLP. On the remaining problem (d) the three techniques are roughly comparable, although SRMP converges to a worse solution.

On a subset of problems (b,c,e,f) we also ran the GTRW-S code from [5], and found that its behaviour is very similar to SRMP - see Fig. 2.[7] We do not claim any speed improvement over GTRW-S; instead, the advantage of SRMP is a simpler and a more general formulation (as discussed in section 4, we believe that with particular weights SRMP becomes equivalent to GTRW-S).

7 Conclusions

We presented a new family of algorithms which includes CMP and TRW-S as special cases. The derivation of SRMP is shorter than that of TRW-S; this should facilitate generalizations to other cases. We developed such a generalization for higher-order graphical models, but we also envisage other directions. An interesting possibility is to treat edges in a pairwise graphical model with different weights depending on their "strengths"; SRMP provides a natural way to do this. (In TRW-S modifying a weight of an individual edge is not easy: these weights depend on probabilities of monotonic chains that pass through this edge, and changing them would affect other edges as well.) In certain scenarios it may be desirable to perform updates only in certain parts of the graphical model (e.g. to recompute the result after a small change of the model); again, SRMP may be more suitable for that. [29] presented a "smoothed version of TRW-S" for pairwise models; our framework may allow an easy generalization to higher-order graphical models. We thus hope that our paper will lead to new research directions in the area of approximate MAP-MRF inference.

A Proof of Proposition 1

We can assume w.l.o.g. that $\mathcal{F} = \{\alpha \mid (\alpha,\beta) \in I'_\beta\} \cup \{\beta\}$ and $J = I_\beta = I'_\beta = \{(\alpha,\beta) \mid \alpha \in \mathcal{F} - \{\beta\}\}$; removing other factors and edges will not affect the claim.

Let θ be the output of $\texttt{AMSD}(\beta, I_\beta, \omega)$. We need to show that $\Phi(\theta) \geq \Phi(\theta[m])$ for any message vector m.

It follows from the description of the AMSD procedure that θ satisfies the following for any $\boldsymbol{x}_\beta$:

$$\min_{\boldsymbol{x}_\alpha \sim \boldsymbol{x}_\beta} \theta_\alpha(\boldsymbol{x}_\alpha) = \omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}_\beta) \qquad \forall(\alpha,\beta)\in J \tag{22a}$$

$$\theta_\beta(\boldsymbol{x}_\beta) = \omega_\beta\widehat{\theta}_\beta(\boldsymbol{x}_\beta) \tag{22b}$$

Let $\boldsymbol{x}^*_\beta$ be a minimizer of $\widehat{\theta}_\beta(\boldsymbol{x}_\beta)$. From (22) we get

$$\begin{aligned}\Phi(\theta) &= \sum_{(\alpha,\beta)\in J} \min_{\boldsymbol{x}_\alpha} \theta_\alpha(\boldsymbol{x}_\alpha) + \min_{\boldsymbol{x}_\beta} \theta_\beta(\boldsymbol{x}_\beta) \\ &= \sum_{(\alpha,\beta)\in J} \omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}^*_\beta) + \omega_\beta\widehat{\theta}_\beta(\boldsymbol{x}^*_\beta) \; = \; \widehat{\theta}_\beta(\boldsymbol{x}^*_\beta)\end{aligned}$$

As for the value of $\Phi(\theta[m])$, it is equal to

$$\begin{aligned}&\sum_{(\alpha,\beta)\in J} \min_{\boldsymbol{x}_\alpha} [\theta_\alpha(\boldsymbol{x}_\alpha) - m(\boldsymbol{x}_\beta)] + \min_{\boldsymbol{x}_\beta} \left[\theta_\beta(\boldsymbol{x}_\beta) + \sum_{(\alpha,\beta)\in J} m(\boldsymbol{x}_\beta)\right] \\ &= \sum_{(\alpha,\beta)\in J} \min_{\boldsymbol{x}_\beta} \left[\omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}_\beta) - m(\boldsymbol{x}_\beta)\right] + \min_{\boldsymbol{x}_\beta} \left[\omega_\beta\widehat{\theta}_\beta(\boldsymbol{x}_\beta) + \sum_{(\alpha,\beta)\in J} m(\boldsymbol{x}_\beta)\right] \\ &\leq \sum_{(\alpha,\beta)\in J} \left[\omega_{\alpha\beta}\widehat{\theta}_\beta(\boldsymbol{x}^*_\beta) - m(\boldsymbol{x}^*_\beta)\right] + \left[\omega_\beta\widehat{\theta}_\beta(\boldsymbol{x}^*_\beta) + \sum_{(\alpha,\beta)\in J} m(\boldsymbol{x}^*_\beta)\right] \\ &= \widehat{\theta}_\beta(\boldsymbol{x}^*_\beta)\end{aligned}$$

B Proof of Proposition 2

We can assume w.l.o.g. that $\mathcal{F} = \{\beta \mid (\alpha,\beta) \in O'_\alpha\} \cup \{\alpha\}$ and $J = O_\alpha = O'_\alpha = \{(\alpha,\beta) \mid \beta \in \mathcal{F} - \{\alpha\}\}$; removing other factors and edges will not affect the claim.

Let θ be the output of $\texttt{AMPLP}(\alpha, O_\alpha, \rho)$. We need to show that $\Phi(\theta) \geq \Phi(\theta[m])$ for any message vector m.

Let $\boldsymbol{x}^*_\alpha$ be a minimizer of $\widehat{\theta}_\alpha$, and correspondingly $\boldsymbol{x}^*_\beta$ be its restriction to factor $\beta \in \mathcal{F}$.

Proposition 10 *(a) $\boldsymbol{x}^*_\alpha$ is a minimizer of $\theta_\alpha(\boldsymbol{x}_\alpha) = \widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) - \sum_{(\alpha,\beta)\in J} \rho_{\alpha\beta}\delta_{\alpha\beta}(\boldsymbol{x}_\beta)$.*
(b) $\boldsymbol{x}^*_\beta$ is a minimizer of $\theta_\beta(\boldsymbol{x}_\beta) = \rho_{\alpha\beta}\delta_{\alpha\beta}(\boldsymbol{x}_\beta)$ for each $(\alpha,\beta) \in J$.

Proof. The fact that $\boldsymbol{x}^*_\beta$ is a minimizer of $\delta_{\alpha\beta}(\boldsymbol{x}_\beta)$ follows directly the definition of vector $\delta_{\alpha\beta}$ in $\texttt{AMPLP}(\alpha, O_\alpha, \rho)$. To show (a), we write the expression as

$$\begin{aligned}&\widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) - \sum_{(\alpha,\beta)\in J} \rho_{\alpha\beta}\delta_{\alpha\beta}(\boldsymbol{x}_\beta) \\ &\quad = \rho_\alpha\widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) + \sum_{(\alpha,\beta)\in J} \rho_{\alpha\beta}\left[\widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) - \delta_{\alpha\beta}(\boldsymbol{x}_\beta)\right]\end{aligned}$$

and observe that $\boldsymbol{x}^*_\alpha$ is a minimizer of each term on the RHS; in particular, $\min_{\boldsymbol{x}_\alpha} \left[\widehat{\theta}_\alpha(\boldsymbol{x}_\alpha) - \delta_{\alpha\beta}(\boldsymbol{x}_\beta)\right] = \widehat{\theta}_\alpha(\boldsymbol{x}^*_\alpha) - \delta_{\alpha\beta}(\boldsymbol{x}^*_\beta) = 0$. □

Using the proposition, we can write

$$\begin{aligned}\Phi(\theta) &= \min_{\boldsymbol{x}_\alpha} \theta_\alpha(\boldsymbol{x}_\alpha) + \sum_{(\alpha,\beta)\in J} \min_{\boldsymbol{x}_\beta} \theta_\beta(\boldsymbol{x}_\beta) \\ &= \theta_\alpha(\boldsymbol{x}^*_\alpha) + \sum_{(\alpha,\beta)\in J} \theta_\beta(\boldsymbol{x}^*_\beta)\end{aligned}$$

[7] In the plots (b,c,e,f) we assume that one iteration of SRMP takes the same time as one iteration of GTRW-S. Note that in practice these times were different: SRMP iteration was about 50% slower than GTRW-S iteration on (e), but about 9 times faster on (f).

As for the value of $\Phi(\theta[m])$, it is equal to

$$\begin{aligned}
&\min_{\boldsymbol{x}_\alpha}\left[\theta_\alpha(\boldsymbol{x}_\alpha)-\sum_{(\alpha,\beta)\in J} m(\boldsymbol{x}_\beta)\right]+\sum_{(\alpha,\beta)\in J}\min_{\boldsymbol{x}_\beta}\left[\theta_\beta(\boldsymbol{x}_\beta)+m(\boldsymbol{x}_\beta)\right]\\
&\le\left[\theta_\alpha(\boldsymbol{x}^*_\alpha)-\sum_{(\alpha,\beta)\in J} m(\boldsymbol{x}^*_\beta)\right]+\sum_{(\alpha,\beta)\in J}\left[\theta_\beta(\boldsymbol{x}^*_\beta)+m(\boldsymbol{x}^*_\beta)\right]\\
&=\theta_\alpha(\boldsymbol{x}^*_\alpha)+\sum_{(\alpha,\beta)\in J}\theta_\beta(\boldsymbol{x}^*_\beta)
\end{aligned}$$

C Implementation of anisotropic MPLP via messages

For simplicity we assume that $O'_\alpha = O_\alpha$ (which is the case in our implementation).

We keep messages $m_{\alpha\beta}$ for edges $(\alpha,\beta)\in J$ that define current reparameterization $\theta=\bar\theta[m]$ via eq. (4). To speed up computations, we also store vector θ_β for all factors $\beta\in\mathcal{F}$ that have at least one incoming edge $(\alpha,\beta)\in J$.

The implementation of $\texttt{AMPLP}(\alpha,O_\alpha,\rho)$ is given below; all updates should be done for all labelings $\boldsymbol{x}_\alpha,\boldsymbol{x}_\beta$ with $\boldsymbol{x}_\alpha\sim\boldsymbol{x}_\beta$. In step 1 we essentially set messages $m_{\alpha\beta}$ to zero, and update θ_β accordingly. (We could have set $m_{\alpha\beta}(\boldsymbol{x}_\beta):=0$ explicitly, but this would have no effect.) In step 2 we move θ_β to factor α. Again, we could have set $\theta_\beta(\boldsymbol{x}_\beta):=0$ after this step, but this would have no effect.

To avoid the accumulation of numerical errors, once in a while we recompute stored values θ_β from current messages m.

```
1: for each $(\alpha,\beta)\in O_\alpha$ update $\theta_\beta(\boldsymbol{x}_\beta):=\theta_\beta(\boldsymbol{x}_\beta)-m_{\alpha\beta}(\boldsymbol{x}_\beta)$, set $\widehat\theta_\beta(\boldsymbol{x}_\beta):=\theta_\beta(\boldsymbol{x}_\beta)$
2: set $\theta_\alpha(\boldsymbol{x}_\alpha):=\bar\theta_\alpha(\boldsymbol{x}_\alpha)+\sum_{(\gamma,\alpha)\in I_\alpha}m_{\gamma\alpha}(\boldsymbol{x}_\alpha)+\sum_{(\alpha,\beta)\in O_\alpha}\theta_\beta(\boldsymbol{x}_\beta)$
3: for each $(\alpha,\beta)\in O_\alpha$ update
       $\theta_\beta(\boldsymbol{x}_\beta):=\rho_{\alpha\beta}\min_{\boldsymbol{x}_\alpha\sim\boldsymbol{x}_\beta}\theta_\alpha(\boldsymbol{x}_\alpha)$
       $m_{\alpha\beta}(\boldsymbol{x}_\beta):=\theta_\beta(\boldsymbol{x}_\beta)-\widehat\theta_\beta(\boldsymbol{x}_\beta)$
4: if we store $\theta_\alpha$ for $\alpha$ (i.e. if exists $(\gamma,\alpha)\in J$) then update $\theta_\alpha(\boldsymbol{x}_\alpha):=\theta_\alpha(\boldsymbol{x}_\alpha)-\sum_{(\alpha,\beta)\in O_\alpha}\theta_\beta(\boldsymbol{x}_\beta)$
```

D Equivalence to TRW-S

Consider a pairwise model. Experimentally we verified that SRMP is equivalent to the TRW-S algorithm [2], assuming that all chains in [2] are assigned the uniform probability and the weights in SRMP are set via eq. (14) and (16). More precisely, the lower bounds produced by the two were identical up to the last digit (eventhough the messages were different). In this section we describe how this equivalence could be established.

As discussed in section 4, the second alternative to implement SRMP is to keep messages $\widehat m$ that define the current reparameterization θ via the following two-stage procedure: (i) compute $\widehat\theta=\bar\theta[\widehat m]$ via eq. (4); (ii) compute

$$\begin{aligned}
\theta_\alpha(\boldsymbol{x}_\alpha)&=\widehat\theta_\alpha(\boldsymbol{x}_\alpha)+\sum_{(\alpha,\beta)\in O_\alpha}\omega_{\alpha\beta}\widehat\theta_\beta(\boldsymbol{x}_\beta) &&\forall\alpha\in\mathcal{F},|\alpha|=2\\
\theta_\beta(\boldsymbol{x}_\beta)&=\omega_\beta\widehat\theta_\beta(\boldsymbol{x}_\beta) &&\forall\beta\in\mathcal{F},|\beta|=1
\end{aligned}$$

where $\omega_{\alpha\beta},\omega_\beta$ are the weights used in the last update for β. We also store vectors $\widehat\theta_\beta$ for singleton factors $\beta=\{i\}\in\mathcal{S}$ and update them as needed.

Consider the following update for factor β:

```
for each $\alpha$ with $(\alpha,\beta)\in I_\beta$ update $\widehat m_{\alpha\beta}(\boldsymbol{x}_\beta):=$
    $\min_{\boldsymbol{x}_\alpha}\left\{\bar\theta_\alpha(\boldsymbol{x}_\alpha)+\sum_{(\alpha,\gamma)\in O_\alpha,\gamma\neq\beta}\left[\omega_{\alpha\gamma}\widehat\theta_\gamma(\boldsymbol{x}_\gamma)-\widehat m_{\alpha\gamma}(\boldsymbol{x}_\gamma)\right]\right\}$
update $\widehat\theta_\beta(\boldsymbol{x}_\beta):=\bar\theta_\beta(\boldsymbol{x}_\beta)+\sum_{(\alpha,\beta)\in I_\beta}\widehat m_{\alpha\beta}(\boldsymbol{x}_\beta)$
```

We claim that this corresponds to procedure $\texttt{AMSD}(\beta,I_\beta,\omega)$; a proof of this is left to the reader. This implies that Algorithm 1 is equivalent to repeating the following: (i) run the updates above for factors $\beta\in\mathcal{S}$ in order $\preceq$ using weights ω^+, but skip the updates of messages $\widehat m_{\alpha\beta}$ for $(\alpha,\beta)\in I_\beta-I^+_\beta$; (ii) reverse the ordering, swap $I^+_\beta\leftrightarrow I^-_\beta,\omega^+_{\alpha\beta}\leftrightarrow\omega^-_{\alpha\beta}$. [8]

It can now be seen that the updates given above are equivalent to those in [2]. Note, the order of operations is slightly different: in one step of the forward pass we send messages from node i to higher-ordered nodes, while in [2] messages are sent from lower-ordered nodes to i. However, is can be checked that the result in both cases is the same.

Acknowledgments

I thank Thomas Schoenemann for his help with the experimental section and providing some of the data. This work was in parts funded by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 616160.

References

[1] T. Werner, "A linear programming approach to max-sum problem: A review," *PAMI*, vol. 29(7), pp. 1165–1179, 2007.

[8] Skipping updates for $(\alpha,\beta)\in I_\beta-I^+_\beta$ is justified as in Proposition 3. Note, the very first forward pass of the described procedure is not equivalent to AMSD updates, but is equivalent to the updates in Algorithm 1; verifying this claim is again left to the reader.

[2] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *PAMI*, vol. 28(10), pp. 1568–1583, 2006.

[3] M. Wainwright, T. Jaakkola, and A. Willsky, "MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches," *Trans. on Information Theory*, vol. 51(11), pp. 3697–3717, 2005.

[4] T. Hazan and A. Shashua, "Norm-product belief propagation: Primal-dual message-passing for approximate inference," *IEEE Trans. on Information Theory*, vol. 56, no. 12, pp. 6294–6316, 2010.

[5] T. Schoenemann and V. Kolmogorov, "Generalized sequential tree-reweighted message passing," in *Advanced Structured Prediction*, S. Nowozin, P. V. Gehler, J. Jancsary, and C. Lampert, Eds. MIT Press, 2014, code: https://github.com/Thomas1205/Optimization-Toolbox.

[6] T. Werner, "Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction," *PAMI*, vol. 32, no. 8, pp. 1474–1488, 2010.

[7] A. Globerson and T. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations," in *NIPS*, 2007.

[8] D. Sontag, "Approximate inference in graphical models using lp relaxations," Ph.D. dissertation, MIT, Dept. of Electrical Engineering and Computer Science, 2010.

[9] D. Sontag, A. Globerson, and T. Jaakkola, "Introduction to dual decomposition for inference," in *Optimization for Machine Learning*, S. Sra, S. Nowozin, and S. Wright, Eds. MIT Press, 2011.

[10] M. P. Kumar and P. Torr, "Efficiently solving convex relaxations for MAP estimation," in *ICML*, 2008.

[11] D. Sontag and T. Jaakkola, "Tree block coordinate descent for MAP in graphical models," in *AISTATS*, vol. 8, 2009, pp. 544–551.

[12] T. Meltzer, A. Globerson, and Y. Weiss, "Convergent message passing algorithms - a unifying view," in *UAI*, 2009.

[13] Y. Zheng, P. Chen, and J.-Z. Cao, "MAP-MRF inference based on extended junction tree representation," in *CVPR*, 2012.

[14] H. Wang and D. Koller, "Subproblem-tree calibration: A unified approach to max-product message passing," in *ICML*, 2013.

[15] G. Storvik and G. Dahl, "Lagrangian-based methods for finding MAP," *IEEE Trans. on Image Processing*, vol. 9(3), pp. 469–479, Mar. 2000.

[16] M. I. Schlesinger and V. V. Giginyak, "Solution to structural recognition (MAX,+)-problems by their equivalent transformations," *Control Systems and Computers*, no. 1,2, 2007.

[17] N. Komodakis and N. Paragios, "Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles," in *ECCV*, 2008.

[18] ——, "Beyond pairwise energies: Efficient optimization for higher-order MRFs," in *CVPR*, 2009.

[19] J. Johnson, D. M. Malioutov, and A. S. Willsky, "Lagrangian relaxation for MAP estimation in graphical models," in *45th Annual Allerton Conference on Communication, Control and Computing*, 2007.

[20] P. Ravikumar, A. Agarwal, and M. J. Wainwright, "Message-passing for graph-structured linear programs: Proximal projections, convergence, and rounding schemes," *JMLR*, vol. 11, pp. 1043–1080, Mar. 2010.

[21] V. Jojic, S. Gould, and D. Koller, "Accelerated dual decomposition for MAP inference," in *ICML*, 2010.

[22] B. Savchynskyy, J. H. Kappes, S. Schmidt, and C. Schnörr, "A study of Nesterov's scheme for lagrangian decomposition and MAP labeling," in *CVPR*, 2011.

[23] A. F. T. Martins, M. A. T. Figueiredo, P. M. Q. Aguiar, N. A. Smith, and E. P. Xing, "An augmented Lagrangian approach to constrained MAP inference," in *ICML*, 2011.

[24] O. Meshi and A. Globerson, "An alternating direction method for dual MAP LP relaxation," in *ECML PKDD*, 2011, pp. 470–483.

[25] S. Schmidt, B. Savchynskyy, J. H. Kappes, and C. Schnörr, "Evaluation of a first-order primal-dual algorithm for MRF energy minimization," in *EMMCVPR*, 2011.

[26] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "An algorithm for minimizing the piecewise smooth Mumford-Shah functional," in *ICCV*, 2009.

[27] J. H. Kappes, B. Savchynskyy, and C. Schnörr, "A bundle approach to efficient MAP-inference by lagrangian relaxation," in *CVPR*, 2012.

[28] D. V. N. Luong, P. Parpas, D. Rueckert, and B. Rustem, "Solving MRF minimization by mirror descent," in *Advances in Visual Computing*, 2012, pp. 587–598.

[29] B. Savchynskyy, S. Schmidt, J. H. Kappes, and C. Schnörr, "Efficient MRF energy minimization via adaptive diminishing smoothing," in *UAI*, 2012, pp. 746–755.

[30] J. Thapper and S. Živný, "The power of linear programming for valued CSPs," in *FOCS*, 2012.

[31] V. Kolmogorov, "The power of linear programming for finite-valued CSPs: a constructive characterization," in *ICALP*, 2013.

[32] A. Globerson, D. Sontag, D. K. Choe, and Y. Li, "MPLP code, version 2," 2012, http://cs.nyu.edu/{~}dsontag/code/mplp_ver2.tgz.

[33] J. Kappes, B. Andres, C. Schnörr, F. Hamprecht, S. Nowozin, D. Batra, J. Lellman, N. Komodakis, S. Kim, B. Kausler, and C. Rother, "A comparative study of modern inference techniques for discrete energy minimization problems," in *CVPR*, 2013.

[34] http://cs.nyu.edu/{~}dsontag/code/README_v2.html.

[35] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon, "Global stereo reconstruction under second order smoothness priors," in *CVPR*, 2008.

[36] D. Sontag, T. Meltzer, A. Globerson, Y. Weiss, and T. Jaakkola, "Tightening LP relaxations for MAP using message passing," in *UAI*, 2008.

[37] D. Sontag, D. K. Choe, and Y. Li, "Efficiently searching for frustrated cycles in MAP inference," in *UAI*, 2012.

[38] http://www.cs.huji.ac.il/project/PASCAL/.