A convex relaxation for multi-graph matching

Paul Swoboda*[,1,2], Dagmar Kainmüller[3,4], Ashkan Mokarian[3,4], Christian Theobalt[1,2], Florian Bernard[1,2]

[1]MPI Informatics [2]Saarland Informatics Campus [3]Berlin Institute of Health [4]MDC Berlin

Abstract

We present a convex relaxation for the multi-graph matching problem. Our formulation allows for partial pairwise matchings, guarantees cycle consistency, and our objective incorporates both linear and quadratic costs. Moreover, we also present an extension to higher-order costs. In order to solve the convex relaxation we employ a message passing algorithm that optimizes the dual problem. We experimentally compare our algorithm on established benchmark problems from computer vision, as well as on large problems from biological image analysis, the size of which exceed previously investigated multi-graph matching instances.

1. Introduction

Finding correspondences between images or shapes is a long-standing problem in computer vision and computer graphics research. Such problems are of high relevance for various applications, among them tracking, segmentation or shape modelling. However, many formulations of the correspondence problem, such as the well-known *quadratic assignment problem* (QAP), are known to be NP-hard. Most correspondence problems can be interpreted as an instance of the *graph matching* problem, where the objective is to establish correspondences between the nodes of two given graphs, such that the edges of both graphs are matched consistently. The *multi-graph matching* (MGM) problem generalizes graph matching to simultaneously establishing correspondences between more than two graphs. For multiple matchings, the notion of cycle consistency arises: assume that X_{pq} is the assignment matrix between graph p and q. The condition $X_{pr}X_{rq} = X_{pq}\ \forall p, q, r$ is called cycle consistency, see Figure 1 for an illustration.

Multi-matching problems are, among others, relevant for multi-view reconstruction, tracking of objects in videos or shape collection alignment. Generally, computing correspondences via multi-graph matching results in higher-quality solutions in comparison to matches computed by a series of graph matching problems only. The reason is that spurious matches introduced by noise in the data can be corrected, since each correspondence between two graphs depends on other matches via cycle consistency.

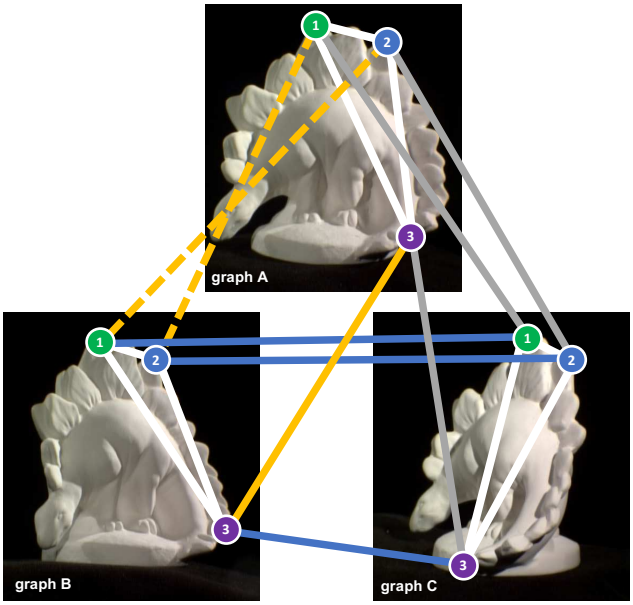

Figure 1. Illustration of cycle consistency in multi-graph matching (best viewed in color). Each graph A, B, C comprises three nodes (green, blue, purple) and three edges (white lines). The true correspondence is indicated by the node colour and node labels 1, 2, 3. Matchings between pairs of graphs are shown by coloured lines (A↔B in yellow, A↔C in gray, and B↔C in blue). Wrong matchings are indicated by dashed lines. The multi-matching A1↔B2↔C2↔A2 is not cycle consistent.

While matching problems between two graphs are well-studied and have received attention since more than fifty years [6, 18, 28, 20, 9, 36, 35, 48, 30, 26, 12, 1, 16, 11, 4, 46, 49, 14, 21, 2, 19], the multi-graph matching problem is less well-studied and hence offers great potential for improvements, both on the theoretical and practical side. In this work we propose a novel multi-graph matching approach that has the following main contributions:

*Email of corresponding author: pswoboda@mpi-inf.mpg.de

Contributions. In contrast to most prior work, our approach is based on a principled and theoretically well-grounded convex optimization approach that (i) *jointly* optimizes a general quadratic multi-graph matching objective while considering cycle consistency constraints, (ii) provides primal/dual gaps w.r.t. a strong relaxation, (iii) is independent of the initialization, (iv) is scalable to large-scale problems due to the use of state-of-the-art message passing techniques, and (v) can be easily extended to the multi-hypergraph matching problem. To our knowledge, there exists no solver in the literature that combines these desirable characteristics.

2. Related work

We review relevant algorithmic prior work for the graph matching and the multi-graph matching problem below.

Graph-matching. The simplest version of graph matching is the linear assignment problem (LAP) that can be solved in polynomial time with the Hungarian [24] or Auction [6] algorithm. For quadratic costs, the graph matching problem is also known as the quadratic assignment problem (QAP) [18]. It is considered to be one of the practically most difficult NP-hard problems [28]. Therefore, many heuristics and approximative algorithms have been proposed, among them algorithms based on Lagrangian relaxation [36, 47, 35], semidefinite programming [30, 16], and other techniques from convex optimization [12, 1, 11, 4]. Apart from that, primal heuristics have been proposed that are based on spectral techniques [20, 9], path-following [46, 49, 14], loopy belief propagation [2], and ADMM [19]. For survey papers that give an overview of techniques used in the combinatorial optimization community we refer to [28, 21]. Higher-order variants of graph matching, known as *hypergraph matching*, have also been considered, e.g. in [25, 10].

Our algorithm can be considered as an extension of the message passing techniques proposed in [47, 35] from the graph matching problem to the more difficult multi-graph matching problem.

Multi-graph matching. Various techniques have been applied for solving multi-graph matching problems. The method [32] holds a tensor that represents all pairwise matchings simultaneously. This way, cycle consistency is satisfied, but their approach is not scalable. A fast algorithm for MGM based on clustering was proposed in [37], where, however, only linear costs are considered.

In [44], the authors alternatingly optimize the individual graph matching problems and enforce cycle consistency repeatedly to obtain progressively better MGM solutions. The work [38] proposes a smooth nonconvex rank-constrained formulation of the multi-matching problem and utilize block coordinate descent on the resulting problem. Other approaches include extensions of random walk based methods [29], factorized graph matching [49] or matrix factorization [45]. The authors of [43] propose to alternatingly use existing graph matching solvers such that ultimately cycle consistency is achieved. The work [42, 41] also use existing graph matching solvers and gradually extend the problem by adding cycle consistency constraints until a feasible multi-graph matching is obtained. However, the works [43, 42, 41] do not use an overall optimization formulation.

In [16, 4], the authors consider a convex relaxation for MGM based on semidefinite programming. While the approach [16] relies on a variable lifting that makes the problem computationally expensive, the approach in [4] is lifting-free but only discussed for the case of full matchings.

Another line of works split the solution of the MGM problem into two steps: solving the individual pairwise graph matching problems first, and enforcing cycle consistency as post-processing. The works [27, 7, 50, 31, 22, 5] assume they are given individual matchings and then postprocess them via matrix factorization to obtain cycle-consistent matchings, which they call permutation synchronization. Similarly, in [3] the authors improve given matchings, but they do not obtain cycle-consistent matchings.

Organization. Section 3 contains our overall multi-graph matching approach. In Section 3.1 we formally state the MGM problem, in Section 3.2 we describe the general *Lagrange decomposition* framework for linear programming (LP) relaxations, and in Section 3.3 we present the MGM problem decomposition within this framework. To obtain a scalable solver for the resulting LP we propose to use *message passing*, where we describe the messages in Section 3.4, and the solver itself in Section 3.5. Since cycle consistency is enforced through a cubic number of constraints, in Section 3.6 we propose a *dual cutting plane algorithm* to include only the required constraints in a working set. We discuss extensions to the multi-hypergraph matching problem in Section 3.7. Finally, in Section 4 we experimentally evaluate our solver on problems from computer vision and biomedical image analysis. We provide additional details in Appendix A. Code and datasets are available from `https://github.com/LPMP/LPMP`.

3. Lagrangian MGM relaxation

In this section we will first present the multi-graph matching problem with quadratic costs. Next, we review the Lagrange decomposition framework [34] and show how it can be applied to decompose the MGM into efficiently solvable subproblems. We also review the message passing algorithm from [34] for general decompositions and detail how our MGM decomposition can be optimized by this method. Last, we describe a dual cutting plane algorithm for cycle consistency constraints.

Since the problem decomposition is complex, the nota-

tion necessary to describe it is so as well. To aid the reader we consistently use symbols for indices which refer to the same type of object. The used index variables are summarised in Table 1.

Table 1. Notational conventions

symbol	meaning
j, k	subproblem
s, t	vector and matrix indices
p, q, r	index of pairwise GM problems
i, ℓ	temporary indices for sums
$\{\cdot\}^{[pq]}$	matching from p to q, $p < q$
$\overline{\{\cdot\}}^{[pq]}$	matching from q to p, $p < q$
d	number of graphs
m_p	number of nodes in graph p

3.1. Problem formulation

We phrase the problem of multi-graph matching as jointly solving pairwise graph matching problems between all pairs of graphs under additional cycle consistency constraints. Although our approach is applicable to considering subsets of pairwise graph matchings, for notational convenience we phrase the MGM problem as the matching of all possible pairs of graphs. We assume that the cost for matching the p-th graph and the q-th graph, where $p, q \in [d] := \{1, \dots, d\}$ for d being the total number of graphs, is given by $(x^{[pq]})^\top W^{[pq]} x^{[pq]}$, so that the MGM problem reads

$$\min_{\{X^{[pq]} \in \mathbb{P}_{m_p m_q}\}} \quad \sum_{p,q \in [d]} (x^{[pq]})^\top W^{[pq]} x^{[pq]} \tag{1}$$

$$\text{s.t.} \quad X^{[pq]} X^{[qr]} \leq X^{[pr]}\,, \tag{2}$$

where we define $x^{[pq]} := \mathrm{vec}(X^{[pq]})$ and the set of $m \times n$ (partial) permutation matrices $\mathbb{P}_{mn}$ is defined as

$$\mathbb{P}_{mn} = \{X \in \{0,1\}^{m \times n} : X\mathbf{1}_n \leq \mathbf{1}_m, X^\top \mathbf{1}_m \leq \mathbf{1}_n\}\,. \tag{3}$$

Note that we write all indices that refer to pairs (or triplets) of graphs in the MGM problem in brackets, e.g. $W^{[pq]}$.

Proposition 1. *Let $(X^{[pq]})_{p,q \in [d]}$ be a set of partial matchings. Then constraints* (2) *cut off all non-cycle-consistent elements.*

We give a minimal example showing when constraints 2 are active in Example 1 in the Appendix.

3.2. Lagrange decomposition

We will solve Problem (1) in a Lagrange decomposition framework. To this end we recapitulate the framework in [34], where the class of *Integer Relaxed Pairwise Separable Linear Programs* (IRPS-LP) is defined. IRPS-LPs are a special case of dual decomposition [13].

Definition 1 (IRPS-LP [34])**.** *Let $N \in \mathbb{N}$ and let $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ be a graph with $\mathbb{V} = \{1, \dots, N\}$. For every $j \in \mathbb{V}$, let $d_j \in \mathbb{N}$, let $Y^j \subseteq \{0,1\}^{d_j}$, and let $\theta^j \in \mathbb{R}^{d_j}$. Let $\Lambda := \mathsf{conv}(Y^1) \times \cdots \times \mathsf{conv}(Y^N)$. For every $\{j,k\} = e \in \mathbb{E}$, let $m_e \in \mathbb{N}$, $A^{j,k} \in \{0,1\}^{m_e \times d_j}$ and $A^{k,j} \in \{0,1\}^{m_e \times d_k}$ such that*

$$\forall x \in Y^j : \quad A^{j,k} x \in \{0,1\}^{m_e}\,, \text{ and} \tag{4}$$

$$\forall x \in Y^k : \quad A^{k,j} x \in \{0,1\}^{m_e}\,. \tag{5}$$

Then, the LP written below is called integer relaxed pairwise separable *w.r.t. the graph $\mathbb{G}$.*

$$\min_{\mu \in \Lambda} \quad \sum_{j \in \mathbb{V}} \langle \theta^j, \mu^j \rangle \tag{6}$$

$$\text{subject to} \quad \forall \{j,k\} \in \mathbb{E} : \quad A^{j,k}\mu^j = A^{k,j}\mu^k\,. \tag{7}$$

Here, $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ define a general *problem decomposition graph* relevant for IRPS-LP that shall not be confused with the graphs that we aim to match. Every $j \in \mathbb{V}$ defines a subproblem, and every edge $jk \in \mathbb{E}$ defines a dependency of subproblems. Def. 1 is more specific than a general Lagrange decomposition, since, firstly, the subproblems are assumed to be binary, and secondly, the linear constraints (7) that describe the dependence of subproblems are defined by 01-matrices that map 01-vectors to 01-vectors. IRPS-LPs are amenable to efficient optimization by the message passing framework of [34].

In what follows, we will refer to subproblems $j \in \mathbb{V}$ by the distinctive names we give to the free variables $x^j \in Y^j$ they optimize over. It will be clear from context when we use subproblem variables x^j to refer to the subproblem j.

3.3. Multi-graph matching decomposition

We will propose a decomposition of Problem (1) as IRPS-LP. In Fig. 2 we illustrate the subproblem decomposition. Our decomposition consists of three types of subproblems: (i) *matching subproblems* that account for matching nodes from one graph to the other, (ii) *quadratic cost subproblems* that account for matching edges from one graph to another, and (iii) *cycle consistency subproblems* that constrain matchings from three distinct graphs to be valid multi-matchings. In what follows, we will use the following notation rule: Let a pairwise graph matching problem between graphs p and q be given, where w.l.o.g. $p < q$. There are two matching directions, with which we will associate two sets of variables: Given nodes (resp. edges) in p, match to nodes (resp. edges) in q. We write variables related to this forward direction as $\{\cdot\}^{[pq]}$, where the respective variable is inserted in place of $\{\cdot\}$. For the reverse direction, i.e. matching from q to p, we distinguish variables by writing them as $\overline{\{\cdot\}}^{[pq]}$.

Matching subproblems. As the matching subproblems are analogous for all pairwise GM problems p, q, we fix p, q

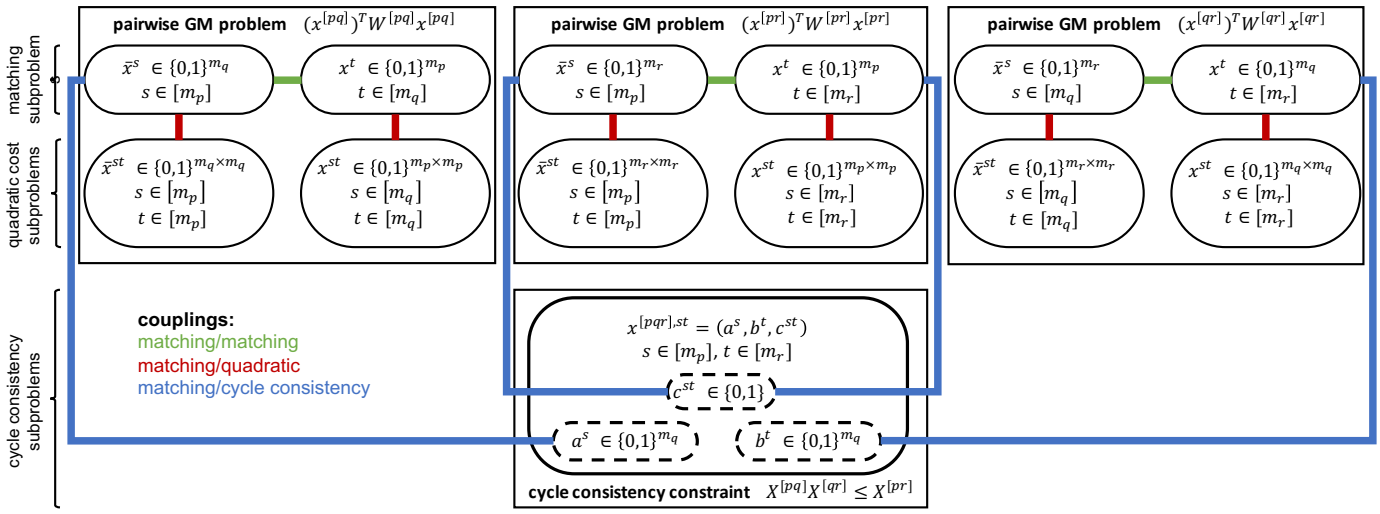

Figure 2. Overview of the subproblem decomposition and their couplings for a triplet of pairwise GM problems $[pq], [pr], [qr]$, along with the $[pqr]$ cycle consistency constraint (best viewed in color). The rounded rectangles correspond to (some of) the nodes of $\mathbb{V}$ in Def. 1, and the coloured lines correspond to (some of) the edges of $\mathbb{E}$, respectively.

and omit the superscripts p, q for the sake of an easier explanation (e.g. we use X instead of $X^{[pq]}$). We write the $(m_p \times m_q)$-dimensional partial matching matrix X in terms of matrix rows and columns as

$$X = \begin{bmatrix} X_{1,*} \\ \vdots \\ X_{m_p,*} \end{bmatrix} = \begin{bmatrix} X_{*,1} & \dots & X_{*,m_q} \end{bmatrix} . \tag{8}$$

For every row with index $s \in [m_p]$ of X we define a subproblem with feasible set $\overline{Y}^s = \{x \in \{0,1\}^{m_q} : \langle x, \mathbf{1} \rangle \leq 1\}$, and for every column with index $t \in [m_q]$ we define a factor with feasible set $Y^t = \{x \in \{0,1\}^{m_p} : \langle x, \mathbf{1} \rangle \leq 1\}$. It follows that $X \in \mathbb{P}_{m_p m_q}$ is equivalent to (9) in conjunction with (10):

$$(X_{s,*})^\top \in \overline{Y}^s \text{ for } s \in [m_p], \text{ and} \tag{9}$$

$$X_{*,t} \in Y^t \text{ for } t \in [m_q] . \tag{10}$$

We add each row and column of X as a subproblem to $\mathbb{V}$, so that we have one m_q-dimensional variable $\overline{x}^s \in \overline{Y}^s$ for each $[s] \in m_p$, and one m_p-dimensional variable $x^t \in Y^t$ for each $[t] \in m_q$. In order to ensure that the variables $\{\overline{x}^s\}, \{x^t\}$ form a valid $X \in \mathbb{P}_{m_p m_q}$, they are coupled via the equality constraints

$$\begin{bmatrix} (\overline{x}^1)^\top \\ \vdots \\ (\overline{x}^{m_p})^\top \end{bmatrix} = \begin{bmatrix} x^1 & \dots & x^{m_q} \end{bmatrix} . \tag{11}$$

The constraints (11) correspond to constraints (6) between $\overline{Y}^s$ and Y^t, as they can be expressed as

$$A^{s,t} \overline{x}^s = A^{t,s} x^t . \tag{12}$$

Quadratic cost subproblems. Again, as the quadratic costs for all pairwise GM problems p, q are analogous, we fix p, q and omit the superscripts p, q for the sake of an easier explanation. In order to linearize the quadratic cost $x^\top W x$ from (1), where $x \in \mathbb{R}^{m_p m_q}$ and $W \in \mathbb{R}^{m_p m_q \times m_p m_q}$, we first observe that

$$x^\top W x = \begin{bmatrix} x^1 \\ \vdots \\ x^{m_q} \end{bmatrix}^\top \begin{bmatrix} W^{(11)} & \dots & W^{(1 m_q)} \\ \vdots & \ddots & \vdots \\ W^{(m_q 1)} & \dots & W^{(m_q m_q)} \end{bmatrix} \begin{bmatrix} x^1 \\ \vdots \\ x^{m_q} \end{bmatrix} , \tag{13}$$

where $x^s \in Y^s$ for $s \in [m_q]$ and $W^{(st)} \in \mathbb{R}^{m_p \times m_p}$,

$$= \sum_{s,t=1}^{m_q} (x^s)^\top W^{(st)} x^t . \tag{14}$$

The blockwise decomposition in equation (14) defines unary and pairwise potentials similarly as in a Markov Random Field (MRF) as follows: Each diagonal element in block $W^{(ss)}$ for $s \in [m_q]$ defines a unary cost $\theta^s = 0.5 \cdot \operatorname{diag}(W^{(ss)})$, so that we can write the cost function $0.5 \cdot (x^s)^\top W^{(ss)} x^s$ for each unary s-factor ($s \in [m_q]$) as $\langle x^s, \theta^s \rangle$ (the factor 0.5 accounts for the decomposition in variables $\{\cdot\}$ and $\overline{\{\cdot\}}$).

For each non-diagonal block $W^{(st)}$ with $s, t \in [m_q], s < t$ we define a pairwise factor with feasible set $Y^{st} = \{x \in \{0,1\}^{m_p \times m_p} : \mathbf{1}^\top x \mathbf{1} = 1\}$. We connect unary variables $x^s \in Y^s$ and $x^t \in Y^t$ with pairwise variable $x^{st} \in Y^{st}$ via constraints as follows:

$$\begin{aligned} A^{s,st} x^s = A^{st,s} \operatorname{vec}(x^{st}) \quad &\Leftrightarrow \quad x^s = x^{st} \mathbf{1} \\ A^{t,st} x^t = A^{st,t} \overrightarrow{(x^{st})} \quad &\Leftrightarrow \quad x^t = (x^{st})^\top \mathbf{1} \end{aligned} \tag{15}$$

The pairwise costs are $\theta^{st} = 0.5 \cdot (W^{(st)} + (W^{(ts)})^\top)$, so that we can write the quadratic cost function $0.5 \cdot ((x^s)^\top W^{(st)} x^t + (x^t)^\top W^{(ts)} x^s)$ in terms of each pairwise st-factor ($s, t \in [m_q], s < t$) as the *linear* term $\langle x^{st}, \theta^{st} \rangle$. Analoguously, we define the costs for the variables $\overline{x}^s, \overline{x}^t, \overline{x}^{st}$, $s, t \in [m_p]$. Note that this construction corresponds to the local polytope [39].

Cycle consistency subproblems. Since the cycle consistency subproblems couple the individual pairwise graph matching problems, in this paragraph we cannot drop the superscripts p, q, r, so that we e.g. write $X^{[pq]}$ instead of X, and $x^{[pq],s} \in Y^{[pq],s}$ instead of $x^s \in Y^s$.

Let now the triplet of matchings $X^{[pq]}, X^{[qr]}$ and $X^{[pr]}$ be given. The element-wise matrix inequality $X^{[pq]} X^{[qr]} \leq X^{[pr]}$ comprises $m_p m_r$ scalar inequalities. Let us consider the scalar inequality at position $(s, t) \in [m_p] \times [m_r]$, which reads

$$X^{[pq]}_{s,*} X^{[qr]}_{*,t} = \sum_{i \in [m_q]} X^{[pq]}_{s,i} X^{[qr]}_{i,t} \leq X^{[pr]}_{st} \,. \tag{16}$$

Accordingly, we define the feasible set

$$Y^{[pqr],st} = \{x^{[pqr],st} = (a, b, c) \in \{0,1\}^{m_q \times m_q \times 1} : \langle a, b \rangle \leq c\} \,. \tag{17}$$

For any p, q, r, s, t, the matching constraints $A^{j,k} \mu^j = A^{k,j} \mu^k$ from (7) translate into

(i) $\overline{x}^{[pq],s} = a$ for $\overline{x}^{[pq],s} \in \overline{Y}^{[pq],s}$ from (9)

(ii) $x^{[qr],t} = b$ for $x^{[qr],t} \in Y^{[qr],t}$ from (10)

(iii) $\overline{x}^{[pr],s}_t = c$ for $\overline{x}^{[pr],s}_t \in \overline{Y}^{[pr],s}$ from (9), and

(iv) $x^{[pr],t}_s = c$ for $x^{[pr],t}_s \in Y^{[pr],t}$ from (10),

where $x^{[pqr],st} = (a, b, c) \in Y^{[pqr],st}$.

Note that here we explicitly indicate the indices of the pairwise graph matching problems for the feasible sets in (9) and (10), e.g. we write $\overline{Y}^{[pq],s}$ to denote $\overline{Y}^s$ in (9) for given p, q.

Remark 2. *Only one of the constraints (iii) and (iv) is necessary. We include both in our formulation, since constraints will translate into Lagrangian variables and for our algorithm it will be advantageous to have this overcomplete representation since it leads to more frequent updates.*

3.4. Messages

As already indicated above, instead of directly solving the primal problem (6), we solve its dual. Specifically, we consider the space of reparametrized cost functions $\overline{\theta}$ that are equivalent to θ, where we require that for every primal μ admissible to (6) it holds that $\langle \mu, \theta \rangle = \langle \mu, \overline{\theta} \rangle$. Such reparametrizated cost functions can be obtained as follows: For any two dependent subproblems $\{j, k\} = e \in \mathbb{E}$ with associated constraint matrices $A^{j,k} \in \{0,1\}^{m_e \times d_j}$, $A^{k,j} \in \{0,1\}^{m_e \times d_k}$ (see Def. 1), we can change the costs θ^j and θ^k by an arbitrary vector $\Delta \in \mathbb{R}^{m_e}$ according to the *update rules*

$$\hat{\theta}^j := \theta^j + (A^{j,k})^\top \Delta \tag{18}$$

$$\hat{\theta}^k := \theta^k - (A^{k,j})^\top \Delta \ . \tag{19}$$

We refer to any update of θ according to the rules (18)–(19) as *message passing*. Message passing does not change the cost of any primal feasible solution, as

$$\begin{aligned} &\langle \hat{\theta}^j, \mu^j \rangle + \langle \hat{\theta}^k, \mu^k \rangle \\ &= \langle \theta^j + (A^{j,k})^\top \Delta, \mu^j \rangle + \langle \theta^k - (A^{k,j})^\top \Delta, \mu^k \rangle && (20) \\ &= \langle \theta^j, \mu^j \rangle + \langle \theta^k, \mu^k \rangle + \langle \Delta, A^{j,k} \mu^j - A^{k,j} \mu^k \rangle && (21) \\ &\overset{(7)}{=} \langle \theta^j, \mu^j \rangle + \langle \theta^k, \mu^k \rangle \ . && (22) \end{aligned}$$

Message passing does, however, change the *dual lower bound* $L(\theta)$ to (6) given by

$$L(\theta) := \sum_{j \in \mathbb{V}} \min_{x \in Y^j} \langle \theta^j, x \rangle \ . \tag{23}$$

The maximum of $L(\theta)$ over all costs obtainable by message passing is equal to the minimum of (6), by linear programming duality. We seek to alter the costs θ by means of message passing so as to maximize the lower bound $L(\theta)$.

Elementary message updates. We call a message update elementary, if it acts on a pair of factors $\{j, k\} \in \mathbb{E}$ and reparametrizes factors j and k by a message Δ as in (18) and (19). An elementary message is required to monotonically decreases the lower bound $L(\theta)$, and additionally is maximal w.r.t. a partial order, as described in [34]. Since in our case all elementary messages can be mechanically derived by following [34], we give the corresponding updates between the matching/quadratic/cycle consistency subproblem factors in Table 2. We denote the message computation by $\Delta = \mathrm{msg}(j, k)$ and the reparametrization by $\mathrm{repam}(\Delta, j, k)$. Our overall algorithm will proceed by passing a series of reweighted elementary messages.

3.5. Message passing algorithm

Algorithm 1 shows a forward pass of the general message passing algorithm for IRPS-LP. It proceeds by sequentially visiting a subset of subproblems in a given order. For each visited factor j it first receives elementary message updates from a subset of neighboring subproblems $R_j^{\rightarrow}$. Second, it sends messages to another set of neighboring subproblems $S_j^{\rightarrow}$ via scaled elementary message passing updates with weights $\omega_j^{\rightarrow}$. In the backward pass, we reverse

$j \in \mathbb{V}$	$k \in \mathbb{V}$	$\Delta = \text{msg}(j,k)$
		matching/matching
$\overline{x}^s \in \overline{Y}^s$	$x^t \in Y^t$	$\overline{\theta}^s(\overline{x}^s_t) - \min_{x \in \overline{Y}^s \setminus \{\overline{x}^s_t\}} \overline{\theta}^s(x)$
$x^t \in Y^t$	$\overline{x}^s \in \overline{Y}^s$	$\theta^t(x^t_s) - \min_{x \in Y^t \setminus \{x^t_s\}} \theta^t(x)$
		matching/quadratic
$x^s \in Y^s$	$x^{st} \in Y^{st}$	$\theta^s - \min_{x \in Y^s} \theta^s(x)$
$x^{st} \in Y^{st}$	$x^s \in Y^s$	$\min_{\rightarrow} \theta^{st}$
$x^{st} \in Y^{st}$	$x^t \in Y^t$	$\min_{\downarrow} \theta^{st}$
$\overline{x}^s \in \overline{Y}^s$	$\overline{x}^{st} \in \overline{Y}^{st}$	$\overline{\theta}^s - \min_{x \in \overline{Y}^s} \overline{\theta}^s(x)$
$\overline{x}^{st} \in \overline{Y}^{st}$	$\overline{x}^s \in \overline{Y}^s$	$\min_{\rightarrow} \overline{\theta}^{st}$
$\overline{x}^{st} \in \overline{Y}^{st}$	$x^t \in \overline{Y}^t$	$\min_{\downarrow} \overline{\theta}^{st}$
		matching/cycle consistency
$\overline{x}^{[pq],s} \in \overline{Y}^{[pq],s}$	$(a,b,c) \in Y^{[pqr],st}$	$\overline{\theta}^{[pq],s} - \min_{x \in \overline{Y}^{[pq],s}} \overline{\theta}^{[pq],s}(x)$
$x^{[qr],t} \in Y^{[qr],t}$	$(a,b,c) \in Y^{[pqr],st}$	$\theta^{[qr],t} - \min_{x \in Y^{[qr],t}} \theta^{[qr],t}(x)$
$\overline{x}^{[pr],s} \in \overline{Y}^{[pr],s}$	$(a,b,c) \in Y^{[pqr],st}$	$\overline{\theta}^{[pr],s}(\overline{x}^{[pr],s}_t) - \min_{x \in \overline{Y}^{[pr],s} \setminus \{\overline{x}^{[pr],s}_t\}} \overline{\theta}^{[pr],s}(x)$
$x^{[pr],t} \in Y^{[pr],t}$	$(a,b,c) \in Y^{[pqr],st}$	$\theta^{[pr],t}(x^{[pr],t}_s) - \min_{x \in Y^{[pr],t} \setminus \{x^{[pr],t}_s\}} \theta^{[pr],t}(x)$
$(a,b,c) \in Y^{[pqr],st}$	$\overline{x}^{[pq],s} \in \overline{Y}^{[pq],s}$	$(\min(a_i, a_i + b_i + c, \min_{j \neq i}\{a_i + b_j\}) - \min(0, c, \min_j\{b_j\}))_{i=1,\dots,m_q}$
$(a,b,c) \in Y^{[pqr],st}$	$x^{[qr],t} \in Y^{[qr],t}$	$(\min(b_i, a_i + b_i + c, \min_{j \neq i}\{a_i + b_j\}) - \min(0, c, \min_i\{a_i\}))_{j=1,\dots,m_p}$
$(a,b,c) \in Y^{[pqr],st}$	$\overline{x}^{[pr],s} \in \overline{Y}^{[pr],s}$	$\min(z, z + \min_i\{a_i + b_i\}) - \min(0, \min_i\{a_i\}, \min_j\{b_j\}, \min_{i \neq j}\{a_i + b_j\})$
$(a,b,c) \in Y^{[pqr],st}$	$x^{[pr],t} \in Y^{[pr],t}$	$\min(z, z + \min_i\{a_i + b_i\}) - \min(0, \min_i\{a_i\}, \min_j\{b_j\}, \min_{i \neq j}\{a_i + b_j\})$

Table 2. Elementary message updates. Notation $\min_{\rightarrow} A$ denotes row-wise minimum of matrix A, while $\min_{\downarrow}(A)$ denotes column-wise minimum of A.

the order of visited factors so that in Alg. 1 we replace $(R^{\rightarrow}, S^{\rightarrow}, \omega^{\rightarrow})$ by $(R^{\leftarrow}, S^{\leftarrow}, \omega^{\leftarrow})$.

For notational convenience below, we define $\mathcal{N}_j := \{k : \{j,k\} \in \mathbb{E}\}$ as the neighbours of the j-th subproblem in the subproblem graph $(\mathbb{V}, \mathbb{E})$.

For solving the MGM formulation above with Algorithm 1, we specify the free parameters as follows:

- $\mathbb{V}_{\text{update}}$ corresponds to all matching subproblems.
- Order on $\mathbb{V}_{\text{update}}$: We order the graph matching subproblems (1) lexicographically w.r.t. indices $p, q \in [d]$. For a given graph matching problem between p and q we order the associated matching subproblems by first considering the column matching subproblems $x^{[pq],1}, \dots, x^{[pq],m_q}$ followed by the row matching subproblems $\overline{x}^{[pq],1}, \dots, \overline{x}^{[pq],m_p}$. Moreover, we define
- $R^{\rightarrow}_j := \begin{cases} \mathcal{N}_j \setminus \{x^{[pq],st} : s < t\}, & j = x^{[pq],t} \\ \mathcal{N}_j \setminus \{\overline{x}^{[pq],st} : s < t\}, & j = \overline{x}^{[pq],t} \end{cases}$, and
- $S^{\rightarrow}_j := \begin{cases} \mathcal{N}_j \setminus \{x^{[pq],st} : s > t\}, & j = x^{[pq],t} \\ \mathcal{N}_j \setminus \{\overline{x}^{[pq],st} : s > t\}, & j = \overline{x}^{[pq],t} \end{cases}$, and
- $\omega^{\rightarrow}_j := \frac{1}{\#\{S^{\rightarrow}_j\}}$.

We define $R^{\leftarrow}_j := S^{\rightarrow}_j$ and $S^{\leftarrow}_j := R^{\rightarrow}_j$, i.e. we swap the "$<$" by "$>$" and vice versa for $R^{\rightarrow}_j$ and $S^{\rightarrow}_j$.

Algorithm 1: Forward pass of message passing for IRPS-LP

for $j \in \mathbb{V}_{update} \subset \mathbb{V}$ *in ascending order* **do**
  Receive messages:
  for $k \in R^{\rightarrow}_j \subset \{k \in \mathbb{V} : \{j,k\} \in \mathbb{E}\}$ **do**
    $\Delta = \text{msg}(k,j)$;
    $\text{repam}(\Delta, k, j)$;
  end
  Send messages:
  for $k \in S^{\rightarrow}_j \subset \{k \in \mathbb{V} : \{j,k\} \in \mathbb{E}\}$ **do**
    $\Delta_k = \text{msg}(j,k)$;
  end
  for $k \in S^{\rightarrow}_j \subset \{k \in \mathbb{V} : \{j,k\} \in \mathbb{E}\}$ **do**
    $\text{repam}(\omega^{\rightarrow}_{j,k} \cdot \Delta_k, k, j)$;
  end
end

3.6. Cutting planes for cycle consistency

There are $\mathcal{O}(m^2 d^3)$, $m = \max_{p \in [d]}\{m_p\}$, cycle consistency subproblems, namely one for each triplet of graphs

$p, q, r \in [d]$ and each pair of nodes $s \in [m_p]$, $t \in [m_r]$. Hence, it is not practical to add all of them at once. Since many of them will not be necessary to achieve the LP-optimum, we pursue a cutting plane approach in which we only add those cycle consistency subproblems that are guaranteed to increase the dual lower bound $L(\theta)$. Specifically, we begin the optimization without any cycle consistency subproblems. When no progress occurs or after some number of iterations, we start adding cycle consistency subproblems. To this end, we first enumerate all graph matching triplets $\{p, q, r\}, p, q, r \in [d]$. For each triplet we enumerate all associated cycle consistency subproblems $x^{[pqr],st}$ and test how much the dual lower bound would increase if we add $x^{[pqr],st}$. We record the increase and add the K best cycle consistency subproblems, where K is a fixed number of subproblems to add. The guaranteed increase of the dual lower bound from addition of subproblem $x^{[pqr],st}$ can be computed with Algorithm 2, see Appendix A.

3.7. Multi-hypergraph matching

Our framework can easily be extended to the hypergraph matching case. For the third-order case, we have 3-tensors $W'^{[pq]}$ instead of a matrices $W^{[pq]}$ in (1). In other words, we have a multi-linear symmetric form $W'^{[pq]} \in \mathbb{R}^{m_p m_q \times m_p m_q \times m_p m_q}$ instead of a matrix $W^{[pq]} \in \mathbb{R}^{m_p m_q \times m_p m_q}$ as in (1). To account for this higher order cost formulation we introduce third-order cost subproblems and connect them to quadratic subproblems, exactly as done for MRFs, see [40, 17]. While the hypergraph matching formulation could be used to optimize over a more complicated cost formulation, we use it to tighten our LP-relaxation, as done for MRFs [40]. This is equivalent to having third-order cost subproblems in the Lagrange decomposition with zero cost. Since adding all possible third-order cost subproblems would be computationally prohibitive, we employ the cutting plane approach proposed in [33] which uses reductions to max-cut problems to find violated cycle inequalities. The found cycles are subsequently triangulated to yield third-order subproblems in our formulation.

3.8. Runtime

The runtime per iteration for the basic relaxation is linear in the number of non-zero entries $\#\{ij : W_{ij} \neq 0\}$ and the number of triplet constraints, since the respective operations in Table 2 can be computed in corresponding time. When we additionally tighten our problem, the corresponding message passing operations can be naively performed in time $\mathcal{O}(m_p^3)$ for $p \in [d]$ and each third-order cost subproblem. More efficient message passing operations for zero-cost third-order subproblems are described in [23], where an expected running time of $\mathcal{O}(m_p^2 \log(m_p))$ is given.

4. Experiments

In this section we provide an experimental evaluation of our algorithm, for which we consider two variants:

MP: Our message passing Algorithm 1 with the cycle consistency cutting plane routine from Section 3.6. We obtain a primal solution from the dual solution using permutation synchronization [27] applied on the dual costs after they have been rounded based on solving an LAP.

MP-T: As the MP-algorithm above, but with additional tightening as described in Section 3.7.

4.1. Synthetic MGM problems

Using the experimental protocol from the authors of [41], we generate four different configurations of synthetic MGM problems (`complete`, `density`, `deform`, `outlier`), where for each of them we consider the number of graphs d to vary from 4 to 16. For details on the problem generation we refer to [41]. We compare our MP/MP-T algorithms to RRWM [8], composition-based affinity optimisation (CAO) [41], MatchOpt (mOpt) [44], permutation synchronisation (mSync) [27], and the recent state-of-the-art DS$\star$ method [4]. The results are shown in Fig. 3. Our MP-T approach performs similar to DS$\star$ on the `complete` and `density` instances, and much better compared to the other methods. Note that in contrast to DS$\star$, our approach explicitly considers outliers, and as such our methodology is particularly well-suited in setups with a large portion of outliers (see `outlier` case). Apart from `complete`, where MP is already tight, the tightening (Section 3.7) significantly improves the results, as can be seen when comparing MP with MP-T. We believe it is an advantage of our method that it can be extended to optimizing tighter LP-relaxations, while this would be difficult to do in more ad-hoc approaches [8, 42, 44, 4, 27].

4.2. CMU House & Hotel

In this experiment we consider the CMU `house` and `hotel` sequences, which are image sequences that come with annotated ground truth. In order to obtain *challenging* MGM problems, we consider a setting where 40% of the points are outliers (the total number of points is 10 per image). For this, we have followed the protocol of [41], where further details are described. We consider the same set of MGM algorithms as in Section 4.1. The results of this experiment are shown in Fig. 4. In both datasets, our method (MP-T) achieves a significantly higher precision compared to all other methods, while also achieving a better recall. This again confirms the robustness of our approach.

4.3. C. elegans

Here we consider the large-scale `worms` dataset of [15]. The goal is to find corresponding nuclei of C. elegans, a famous model organism in biology. The dataset contains 30

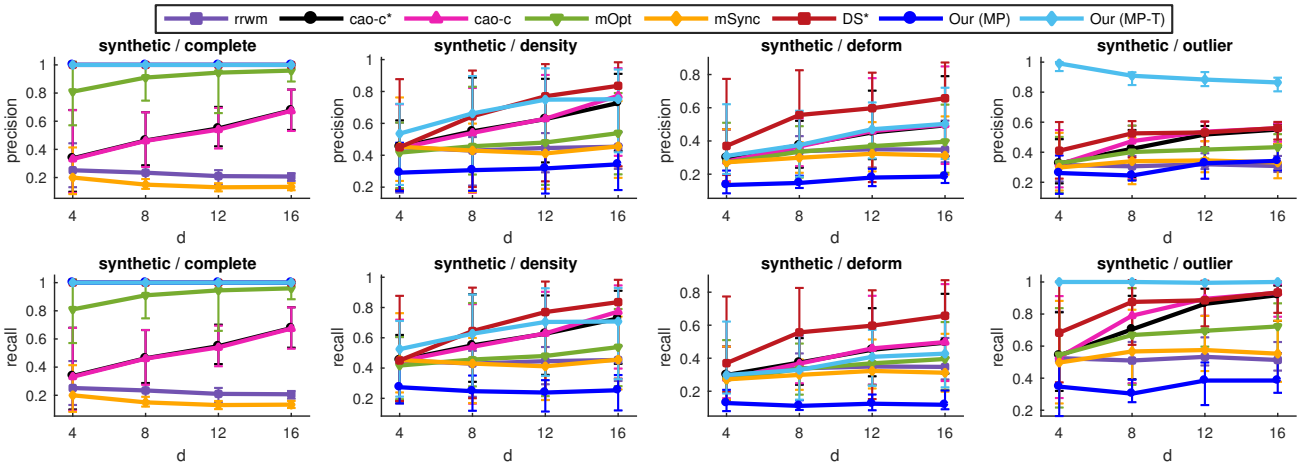

Figure 3. Results on synthetic data (best viewed in color). Note that in the first column (`complete`) the methods `DS*`, `MP` and `MP-T` achieve a perfect matching in all cases.

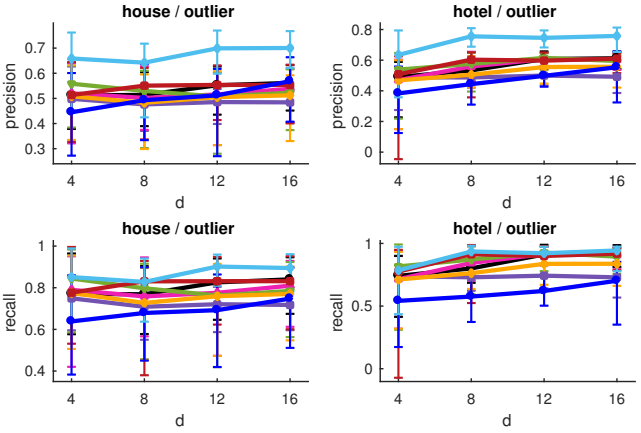

Figure 4. Results on CMU `house` and `hotel` multi-graph matching problems (best viewed in color). Refer to Fig. 3 for the legend.

three-dimensional microscopy images of individual worms and segmentations of their nuclei, where each 3D image depicts one worm that has 558 nuclei. As such, the resulting nuclei matching problems are among the largest graph matching instances ever investigated in the literature (see [35]; note that the pairwise problems we establish for this work differ slightly from the worm matching problems of [35] in that they are $30{\cdot}29/2$ worm-to-worm matching problems, as opposed to 30 atlas-to-worm matching problems). We derive a range of MGM problems by selecting subsets of worms of varying cardinality. The results are summarized in Table 3. It can be seen that with successively larger numbers of graphs the precision and recall are improved. We would like to stress that the largest instances have 36 million optimization variables and methods [8, 42, 44, 4] do not scale well enough. Method [27] does not allow for quadratic costs, hence we cannot report competing algorithmic results for this dataset.

d	2	4	5	6	7	8	9	10
W	0.8M	4M	8M	12M	17M	22M	29M	36M
time	3	20	35	50	90	110	130	165
prec.	.71	.70	.73	.73	.75	.76	.78	.79
recall	.41	.41	.42	.41	.42	.42	.43	.43

Table 3. Quantitative results for the `worms` dataset solved with [35] for $d = 2$ and `MP` for the multi-graph case $d > 2$. We give the number of non-zero entries $\#\{ij : W_{ij} \neq 0\}$, the time in minutes for solving and obtained precision and recall.

5. Conclusion

We have presented a principled and theoretically well-grounded convex relaxation for the multi-graph matching problem based on a Lagrange decomposition. We have phrased MGM as simultaneously solving pairwise graph matching problems that communicate with each other based on cycle consistency constraints. Our proposed formulation is general as it can handle linear, quadratic, and higher-order matching costs, while at the same time considering cycle consistency constraints. Due to the convex formulation it is independent of the initialization, and due to the duality principle we obtain primal/dual gaps that can serve as optimality certificates. Moreover, we have demonstrated that by using additional higher-order terms one can obtain a tighter relaxation. In order to computationally solve the dual problem, we use an efficient algorithm based on message passing. In our experiments we considered standard computer vision benchmark problems, as well as problems from biomedical image analysis. The experimental results demonstrate the merits of our approach.

Acknowledgments

This work was partially funded by the ERC Consolidator Grant 4DRepLy (770784).

References

[1] Yonathan Aflalo, Alexander Bronstein, and Ron Kimmel. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences*, 112(10):2942–2947, 2015.

[2] Dragomir Anguelov, Praveen Srinivasan, Hoi-cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 33–40. MIT Press, 2005.

[3] Federica Arrigoni, Eleonora Maset, and Andrea Fusiello. Synchronization in the symmetric inverse semigroup. In *ICIAP*, 2017.

[4] Florian Bernard, Christian Theobalt, and Michael Moeller. DS*: Tighter lifting-free convex relaxations for quadratic matching problems. In *CVPR*, 2018.

[5] Florian Bernard, Johan Thunberg, Jorge Goncalves, and Christian Theobalt. Synchronisation of partial multi-matchings via non-negative factorisations. *Pattern Recognition*, 92:146–155, 2019.

[6] Dimitri P Bertsekas. Network optimization: Continuous and discrete models. Athena Scientific, 1998.

[7] Yuxin Chen, Leonidas J Guibas, and Qi-Xing Huang. Near-optimal joint object matching via convex relaxation. In *ICML*, 2014.

[8] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. *ECCV*, 2010.

[9] T Cour, P Srinivasan, and J Shi. Balanced graph matching. *NIPS*, 2006.

[10] Olivier Duchenne, Francis Bach, In-So Kweon, and Jean Ponce. A tensor-based algorithm for high-order graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2383–2395, 2011.

[11] Nadav Dym, Haggai Maron, and Yaron Lipman. DS++ - A flexible, scalable and provably tight relaxation for matching problems. *ACM Transactions on Graphics (TOG)*, 36(6), 2017.

[12] Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d'Aspremont. Convex relaxations for permutation problems. In *NIPS*, 2013.

[13] Monique Guignard and Siwhan Kim. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical programming*, 39(2):215–228, 1987.

[14] B Jiang, J Tang, C Ding, and B Luo. Binary constraint preserving graph matching. In *CVPR*, 2017.

[15] Dagmar Kainmueller, Florian Jug, Carsten Rother, and Eugene W Myers. Active Graph Matching for Joint Segmentation and Annotation of C. Elegans. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pages 81–88. Springer, 2014.

[16] Itay Kezurer, Shahar Z Kovalsky, Ronen Basri, and Yaron Lipman. Tight relaxation of quadratic matching. *Comput. Graph. Forum*, 2015.

[17] Vladimir Kolmogorov. A new look at reweighted message passing. *IEEE transactions on pattern analysis and machine intelligence*, 37(5):919–930, 2015.

[18] Eugene L Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.

[19] D Khuê Lê-Huu and Nikos Paragios. Alternating direction graph matching. In *CVPR*, 2016.

[20] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[21] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura Netto, Peter Hahn, and Tania Maia Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.

[22] Eleonora Maset, Federica Arrigoni, and Andrea Fusiello. Practical and efficient multi-view matching. In *ICCV*, 2017.

[23] Julian McAuley and Tibério Caetano. Exploiting within-clique factorizations in junction-tree algorithms. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 525–532, 2010.

[24] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, Mar. 1957.

[25] Quynh Nguyen, Antoine Gautier, and Matthias Hein. A flexible tensor block coordinate ascent scheme for hypergraph matching. 2015.

[26] Carl Olsson, Anders P Eriksson, and Fredrik Kahl. Solving large scale binary quadratic problems - spectral methods vs. semidefinite programming. *CVPR*, 2007.

[27] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. In *NIPS*, 2013.

[28] Panos M Pardalos, Franz Rendl, and Henry Wolkowicz. The quadratic assignment problem - a survey and recent developments. *DIMACS Series in Discrete Mathematics*, 1993.

[29] Han-Mu Park and Kuk-Jin Yoon. Consistent multiple graph matching with multi-layer random walks synchronization. *Pattern Recognition Letters*, 2018.

[30] Christian Schellewald and Christoph Schnörr. Probabilistic subgraph matching based on convex relaxation. In *EMMCVPR*, 2005.

[31] Yanyao Shen, Qixing Huang, Nati Srebro, and Sujay Sanghavi. Normalized spectral map synchronization. In *NIPS*, 2016.

[32] X Shi, H Ling, W Hu, and J Xing. Tensor power iteration for multi-graph matching. In *CVPR*, 2016.

[33] David Sontag, Do Kook Choe, and Yitao Li. Efficiently searching for frustrated cycles in MAP inference. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI-12)*, pages 795–804, Corvallis, Oregon, 2012. AUAI Press.

[34] Paul Swoboda, Jan Kuske, and Bogdan Savchynskyy. A dual ascent framework for Lagrangean decomposition of combinatorial problems. In *CVPR*, 2017.

[35] Paul Swoboda, Carsten Rother, Hassan Abu Alhaija, Dagmar Kainmüller, and Bogdan Savchynskyy. Study of Lagrangean decomposition and dual ascent solvers for graph matching. In *CVPR*, 2017.

[36] L Torresani and V Kolmogorov. A dual decomposition approach to feature correspondence. *TPAMI*, 35(2):259–271, 2013.

[37] Roberto Tron, Xiaowei Zhou, Carloes Esteves, and Kostas Daniilidis. Fast multi-image matching via density-based clustering. In *ICCV*, 2017.

[38] Qianqian Wang, Xiaowei Zhou, and Kostas Daniilidis. Multi-image semantic matching by mining consistent features. In *CVPR*, 2018.

[39] Tomas Werner. A linear programming approach to max-sum problem: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(7):1165–1179, 2007.

[40] Tomáš Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *Conf. on Computer Vision and Pattern Recognition, Anchorage, USA*, June 2008.

[41] J Yan, M Cho, H Zha, and X Yang. Multi-graph matching via affinity optimization with graduated consistency regularization. *TPAMI*, 38(6):1228–1242, 2016.

[42] Junchi Yan, Yin Li, Wei Liu, Hongyuan Zha, Xiaokang Yang, and Stephen Mingyu Chu. Graduated consistency-regularized optimization for multi-graph matching. *ECCV*, 2014.

[43] Junchi Yan, Yu Tian, Hongyuan Zha, Xiaokang Yang, Ya Zhang, and Stephen M Chu. Joint optimization for consistent multiple graph matching. *ICCV*, 2013.

[44] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu. Consistency-driven alternating optimization for multigraph matching: A unified approach. In *IEEE Transactions on Image Processing*, 2015.

[45] J Yan, H Xu, H Zha, X Yang, and H Liu. A matrix decomposition perspective to multiple graph matching. In *ICCV*, 2015.

[46] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *TPAMI*, 31(12):2227–2242, 2009.

[47] Zhen Zhang, Qinfeng Shi, Julian McAuley, Wei Wei, Yanning Zhang, and Anton van den Hengel. Pairwise matching through max-weight bipartite belief propagation. In *CVPR*, 2016.

[48] Q Zhao, S E Karisch, F Rendl, and H Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109, 1998.

[49] Feng Zhou and Fernando De la Torre. Factorized graph matching. *TPAMI*, 38(9):1774–1789, 2016.

[50] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *ICCV*, 2015.