Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 25 (2007) 1301–1314
www.elsevier.com/locate/imavis

Evaluation of a convex relaxation to a quadratic assignment matching approach for relational object views

Christian Schellewald [a,*], Stefan Roth [b], Christoph Schnörr [a]

[a] CVGPR – Group, Department of Mathematics and Computer Science, University of Mannheim, D-68131 Mannheim, Germany
[b] Department of Computer Science, Brown University, Providence, RI 02912, USA

Received 30 March 2006; accepted 16 August 2006

Abstract

We introduce a convex relaxation approach for the quadratic assignment problem to the field of computer vision. Due to convexity, a favourable property of this approach is the absence of any tuning parameters and the computation of high-quality combinatorial solutions by solving a mathematically simple optimization problem. Furthermore, the relaxation step always computes a tight lower bound of the objective function and thus can additionally be used as an efficient subroutine of an exact search algorithm. We report the results of both established benchmark experiments from combinatorial mathematics and random ground-truth experiments using computer-generated graphs. For comparison, a deterministic annealing approach is investigated as well. Both approaches show similarly good performance. In contrast to the convex approach, however, the annealing approach yields no problem relaxation, and four parameters have to be tuned by hand for the annealing algorithm to become competitive.

Keywords: Quadratic assignment; Weighted graph matching; Combinatorial optimization; Convex programming; Object recognition

1. Introduction

1.1. Motivation

Visual object recognition is a central problem of computer vision research. A key question in this context is how to represent objects for the purpose of recognition by a computer vision system. Approaches range from view-based to 3D model-based, from object-centered to viewer-centered representations [1], each of which may have advantages under constraints related to specific applications. Psychophysical findings however provide evidence for *view-based* object representations [2] in human vision, and so we will focus on this representation in this paper. A common and powerful structure for representing object views is to define a set of local image features V along with pairwise relations $E \subset V \times V$ (spatial proximity and (dis)similarity measure) in terms of a weight function $w : E \to \mathbb{R}_+$, that is an undirected weighted graph $G = (V, E)$ (see Fig. 1). Accordingly, a core routine of any recognition system is to compare graphs in different images, or to match graphs against prototypical graphs in some object database.

Finding a good match between two graphs G, G' amounts to compute a permutation of the vertices of one graph so as to become similar to the other one (cf. Section 2). The relaxation approach we are concerned with to tackle this problem was developed by Anstreicher and Brixius [3,4] for the quadratic assignment problem (QAP). It is well-known that the *quadratic assignment problem* and general graph matching problems are NP-hard [5,6] and cannot be solved to optimality even for moderately sized problem instances. For the graph matching problem shown in Fig. 1, an exhaustive search has to check the impractical number of about $38! \approx 5 \cdot 10^{44}$ possible permutations of vertices.

* Corresponding author.
E-mail addresses: c.schellewald@psyreon.de (C. Schellewald), roth@cs.brown.edu (S. Roth), schnoerr@uni-mannheim.de (C. Schnörr).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.08.005

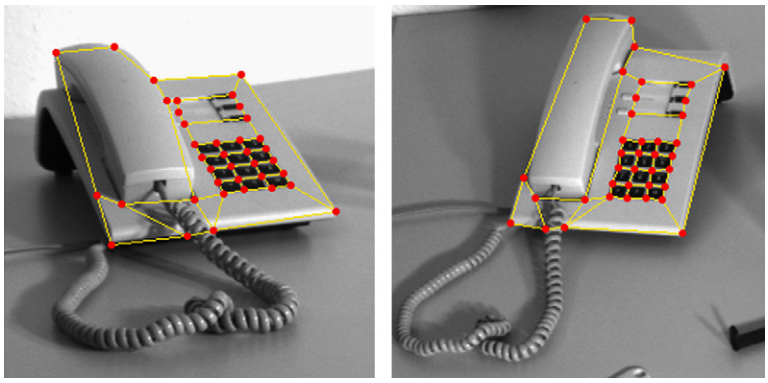

Fig. 1. Two graphs representing the same object. The graphs are created based on features obtained with the FEX-system (cf. [7,8]).

The task of developing good matching algorithms amounts to make a compromise between two contradictory requirements: first, a high-quality solution close to the (unknown) global optimum should be efficiently computable in polynomial time. Second, since the matching algorithm is only a particular module of a computer vision system, there should be no need for manual interaction (selection of a starting point or tuning parameters) in order to obtain good performance. It seems to us that this latter algorithmic aspect has received less attention in the computer vision literature than the former one. This motivated us to look for an approach satisfying *both* requirements.

1.2. Related work

Our aim is to find desired matchings between relational structures that occur in computer vision applications. In order to do this we utilize a reformulation of a weighted graph matching problem into the quadratic assignment problem. The quadratic assignment problem is a classical problem in the field of combinatorial optimization which is very general in the sense that several well-known problems, for example the traveling salesman problem, are special cases of it. For a comprehensive review of the quadratic assignment problem we refer to [9].

Classical approaches to the quadratic assignment problem include

- *linearizations* [10,11] which can be efficiently solved in polynomial time at the cost of a considerably larger number of variables,
- the use of *exact search algorithms*, like the cutting plane method [12] or branch-and-bound [13], which are based on lower bounds [14,10] of the objective function but can only be applied to small problem instances, or
- *heuristic search algorithms* like tabu search, modified simulated annealing, or genetic algorithms [15–17]. In [18] the authors present a ant colony optimization method that is combined with a local search algorithm. The authors in [19] proposed a chaos driven tabu search neural network hardware system to solve the QAP.

A separate class of approaches is formed by *relaxations* of the quadratic assignment problem. Typically, these approaches aim at computing a lower bound of the objective function based on eigenvalue problems [20,21,9]. Recently, an convex relaxation approach has been proposed [4] which can be used to compute both a lower bound and the corresponding approximate minimizer. A favorable property due to the convexity of this approach is that the relaxed solution can numerically be computed without the need to select good starting points and tuning parameter values. This will be studied in more detail below.

Another important approach to the quadratic assignment problem is based on the *deterministic annealing* strategy. This approach has been extensively investigated in the neural-network literature [22–29]. In the context of image segmentation, piecewise-smooth restoration and clustering, deterministic annealing is well-known in the computer vision literature as well [30–35]. A favorable feature of deterministic annealing is that this strategy can be derived in a theoretically sound way using the maximum entropy principle (e.g., [34]). On the other hand, an obvious disadvantage from the viewpoint of algorithm design is the complex bifurcation phenomena encountered when lowering the annealing parameter [36]. To the best of our knowledge, no control strategy is known which guarantees to reach a "good" local minimum. While excellent experimental results have been reported using deterministic annealing, the dependence of these results on various tuning parameters apparently has not received much attention in the literature. In view of this important aspect we present a thorough comparison of a deterministic annealing approach tailored to the quadratic assignment problem [27,29] with a convex programming approach [4].

Further important work on the graph-matching problem includes alternative algorithmic approaches like probabilistic relaxation [37], genetic search [38], error-correcting matching [39] or two-step iterative approaches [40], and also specialized work like, for example, simultaneous estimation of transformation geometry [41], or matching trees in terms of the maximum clique of the association graph [42]. A comparison of this variety of approaches is beyond the scope of

this paper. We also do not discuss variations of the optimization criterion (see, e.g., [37,43]), nor do we investigate the task to determine the weight function w in some optimal way. Rather, by focusing on the optimization problem, we wish to emphasize the advantages of parameter-free convex relaxation in the context of graph-matching and to reveal its performance in comparison to recent approaches which are based on the well-known deterministic annealing framework.

As relaxation has a context-dependent meaning, the word relaxation refers in this paper to the strategy of making a combinatorial problem computationally tractable by keeping exactly the objective criterion with respect to feasible combinatorial solutions but weakening the combinatorial constraints. Solving (approximately) the combinatorial weighted graph matching problem by relaxation has several advantages. First, it is *mathematically* comprehensible where and how approximations are introduced in order to solve the optimization problem. By contrast, many other approaches are based on *algorithmic* modifications, making a theoretical understanding and comparison more difficult. Second, solving a relaxed problem formulation results in a lower bound of the objective function, because weakening the constraints gives more degrees of freedom for minimization. Third, relaxations can be used either to directly compute approximate solutions to the original combinatorial problem, or as subroutines in exact branch-and-bound search algorithms.

Due to these advantages, this paper focuses on relaxations of a weighted graph matching problem formulation.

1.3. Contribution

Our work contributes under the aspects *relaxation, algorithm design* and *performance evaluation*.

1.3.1. Relaxation

A relaxation of a combinatorial weighted graph matching problem formulation is used to directly compute approximate solutions to the original graph matching problems. For this, the weighted graph matching problem formulation is reformulated into a quadratic assignment problem formulation. Different problem relaxations can be *ranked* based on the lower bound they compute and which determines the overall performance: the larger the lower bound, the better the relaxation.

1.3.2. Algorithm design

A common problem of many approaches concerns the selection of tuning parameters in order to obtain good performance. A representative example – a deterministic annealing strategy – will be examined in more detail below. In this context, *convex* optimization approaches provide an attractive alternative, because the *global* optimum exists under mild conditions and can be computed by established numerical algorithms in polynomial time [44] without *any* additional parameters.

1.3.3. Performance evaluation

A widely-used collection of difficult, real-life benchmark problems exists in the field of combinatorial optimization [45], which has become a standard during the last years. Apparently, this database has not been used in the computer vision literature so far in order to evaluate approaches to the weighted graph matching problem. Besides extensive random ground-truth experiments that are created to simulate computer vision graph matching problems, our performance evaluation was carried out for problem instances of this database. In addition to this also a real world graph matching problem will be shown.

1.4. Organisation of the paper

After stating the problem formally in Section 2, we present a hierarchy of relaxations in Section 3, the strongest one being a convex relaxation. For comparison, we sketch two alternative approaches from the literature in Section 4, a simple but fast approach based on eigenvalue decomposition, and a more sophisticated deterministic annealing strategy. In order to stress the difference to these non-convex approaches, various aspects of the convex relaxation approach are visualized for a toy example in Section 5. In Section 6, the results of numerous experiments for both real-life benchmarks from the field of combinatorial optimization and for ground-truth experiments based on computer-generated graphs are summarized. Also a real world example is shown. We conclude and indicate further work in Section 7.

2. Problem statement and definitions

2.1. Notation

We will use the following notation:

$X^\top$	transpose of the matrix X
I_n	$n \times n$ unit matrix
$\mathcal{O}$	set of orthogonal matrices X, i.e., $X^\top X = I_n$
$\mathcal{E}$	set of matrices with unit row and column sums
$\mathcal{N}$	set of non-negative matrices
Π	set of permutation matrices $\Pi \in \mathcal{O} \cap \mathcal{E} \cap \mathcal{N}$
e	vector of all ones: $e_i = 1$, $i = 1, \ldots, n$
$\mathrm{Tr}[X]$	trace of the matrix X
$\cdot$	scalar product of two matrices X, Y: $X \cdot Y = \mathrm{Tr}[X^\top Y]$
$\|X\|$	Frobenius norm of the matrix X: $\|X\| = \mathrm{Tr}[X^\top X]^{1/2}$
$\mathrm{vec}[X]$	vector obtained by stacking the columns of the matrix X
$\otimes$	Kronecker product
$\lambda(X)$	vector of the eigenvalues of the matrix X
δ_{ij}	Kronecker delta: $\delta_{ij} = 1$ if $i = j$, and 0 otherwise

2.2. Problem statement

In this paper, we consider undirected graphs $G = (V, E, w)$ with nodes $V = \{1, \ldots, n\}$ and edges $E \subset V \times V$. The weight function $w : E \to \mathbb{R}_0^+$ typically encodes a similarity measure with respect to pairs of features (i,j). This measure along with the structure of the graph is represented by the adjacency matrix A_G: $(A_G)_{ij} = w_{ij}$, $i, j = 1, \ldots, n$. Since $w_{ij} = w_{ji}$, adjacency matrices are symmetric: $A_G^\top = A_G$.

Let $G = (V_G, E_G, w_G)$ and $H = (V_H, E_H, w_H)$ denote two given graphs. In order to match these two graphs, we want to compute a permutation Φ: $V_G \mapsto V_G$ of the nodes of G such that the following distance measure is minimized:

$$\sum_{i,j=1}^{n} \left(w_{G,\Phi(i)\Phi(j)} - w_{H,ij}\right)^2. \tag{1}$$

Representing the permutation Φ by a permutation matrix $X \in \Pi$, this cost function takes the following form [46] in terms of the adjacency matrices of G and H:

$$f(X) = \|XA_GX^\top - A_H\|^2. \tag{2}$$

For isomorphic graphs exists a permutation matrix such that the minimum value $f(X) = 0$ of the objective function is attained. For features V_G, V_H supplied by an image preprocessing stage, it is unlikely that G and H are isomorphic. In this case we define as the best match the permutation matrix X^* which minimizes f over Π. Thus, the graph matching problem formally reads

$$f(X^*) = \min_{X \in \Pi} \|XA_GX^\top - A_H\|^2. \tag{3}$$

The minimization problem (3) has a close relationship to the quadratic assignment problem (QAP) in combinatorial mathematics (for a survey, see [9]):

$$\min_{X \in \Pi} \mathrm{Tr}[AXBX^\top + CX^\top]. \tag{4}$$

Provided that the graphs have the same number of nodes, this relationship can be seen by reformulating the graph matching objective function as follows:

$$\begin{aligned} f(X) &= \|XA_GX^\top - A_H\|^2 \\ &= C_G + C_H - 2\mathrm{Tr}[A_HXA_G^\top X^\top]. \end{aligned} \tag{5}$$

Dropping the constant terms $C_G = \mathrm{Tr}[A_GA_G^\top]$ and $C_H = \mathrm{Tr}[A_HA_H^\top]$, we recognize the graph matching problem (3) as a special case of the quadratic assignment problem (4) with $A = A_H$, $B = -A_G$ and $C = 0$. Throughout the remainder of this paper, we can therefore consider the following optimization problem:

$$\text{(QAP)} \qquad \min_{X \in \Pi} \mathrm{Tr}[AXB^\top X^\top]. \tag{6}$$

We note that (1) corresponds to (6) only if $|V_G| = |V_H| = n$. In this paper, we make this simplifying assumption (as did Umeyama [46], for instance) in order to assess the techniques which have been developed for the quadratic assignment problem for the weighted graph matching problem in computer vision. The issue of extending the techniques to *sub*graph matching will be taken up in Section 7.

3. Relaxations and lower bounds

In this section, we consider various relaxations of problem (6). We will see that a ranking of these approaches can be obtained by virtue of the corresponding lower bounds.

3.1. Orthogonal relaxation

Relaxing the set Π to $\mathcal{O} \supset \Pi$, Finke et al. [20] suggested the so-called eigenvalue bound (EVB) which gives a lower bound for the minimization problem (6):

$$\text{(EVB)} \qquad \min_{X \in \mathcal{O}} \mathrm{Tr}[AXB^\top X^\top] = \langle \lambda(A), \lambda(B) \rangle. \tag{7}$$

Here, $\langle \lambda(A), \lambda(B) \rangle$ denotes the so-called *minimal scalar product*. This is the scalar product of the vectors $\lambda(A)$ and $\lambda(B)$ containing the eigenvalues of the adjacency matrices A and B ordered as follows: $\lambda_1(A) \leqslant \lambda_2(A) \leqslant \cdots \leqslant \lambda_n(A)$ and $\lambda_1(B) \geqslant \lambda_2(B) \geqslant \cdots \geqslant \lambda_n(B)$. The matrix X for which the bound (EVB) is attained can be calculated as well. If $U, V \in \mathcal{O}$ diagonalize the adjacency matrices A and B, respectively, and this columns are arranged according to the order of the eigenvalues mentioned above, then $X = UV^\top$. It turned out that in many cases this relaxation yields a bound for the minimization problem (6) which is too weak to be useful in practice.

3.2. Projected eigenvalue bound

Hadley et al. [21] improved the lower bound (7) by taking into account the constraint set $\mathcal{E}$ in addition to $\mathcal{O}$. To this end, they parameterized matrices $X \in \mathcal{O} \cap \mathcal{E}$ based on $(n-1) \times (n-1)$ orthogonal matrices $\hat{X} \in \mathcal{O}$ and the relationship

$$X = V\hat{X}V^\top + \frac{1}{n}E,$$

where $E = ee^\top$ and the $n-1$ columns of the $n \times (n-1)$ matrix V form a basis of the subspace orthogonal to the vector e. Conversely, for any $(n-1) \times (n-1)$ matrix $\hat{X} \in \mathcal{O}$ we have $X = V\hat{X}V^\top + \frac{1}{n}E \in \mathcal{O} \cap \mathcal{E}$. The $n \times (n-1)$ matrix V can be calculated using the following scheme:

$$V = \begin{pmatrix} a & a & \cdots & a & a \\ 1+b & b & \cdots & b & b \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ b & b & \cdots & b & 1+b \end{pmatrix}$$

with

$$a = -\frac{1}{\sqrt{n}} \quad \text{and} \quad b = -\frac{1}{n+\sqrt{n}}.$$

Using this parameterization, the objective function can be rearranged as follows:

$$\mathrm{Tr}[AXB^\top X^\top] = \mathrm{Tr}[\hat{A}\hat{X}\hat{B}^\top\hat{X}^\top] + \mathrm{Tr}[DX] - C_1, \tag{8}$$

where $\hat{A} = V^\top AV$, $\hat{B} = V^\top BV$, $D = \frac{2}{n}r(A)r(B)^\top$ and $C_1 = \frac{1}{n^2}s(A)s(B)$. The vector $r(A) = Ae$ denotes the vector of row sums of the matrix A, and the scalar $s(A) = e^\top Ae$ is the sum of all elements in A. The authors of [21] suggested to optimize the first two terms on the right hand side of (8) separately, the first one over $\hat{X} \in \mathcal{O}(n-1)$, and the second one over $X \in \Pi$. The latter problem amounts to solve the linear assignment problem

$$\mathrm{LAP}(D) = \min_{X\in\Pi} \mathrm{Tr}[DX], \tag{9}$$

which can be solved using any linear programming solver. As a result, the projected eigenvalue bound as a lower bound for the minimization problem (6) is obtained

$$\text{(PEVB)} \qquad \langle\lambda(\hat{A}), \lambda(\hat{B})\rangle_- + \mathrm{LAP}(D) - C_1. \tag{10}$$

However, a major drawback of this bound is that due to separately minimizing the two terms in (8), a corresponding minimizing matrix X cannot be computed in general. The next subsection shows how one can overcome this drawback by convex relaxation and even achieve a better lower bound than (10).

3.3. Convex relaxation

Following the work of Anstreicher, Brixius and Wolkowicz [3,4], we focus on a convex relaxation of the minimization problem (6) in this section. Besides the general arguments discussed in Section 1.1, the main motivation for this approach is its ability to compute *both* a tight lower bound and the corresponding matrix X where this bound is attained. In general this is not possible for the bound (10). As a starting point reconsider the minimization of the first term of the right hand side of Eq. (8) over the set $\mathcal{O}(n-1)$

$$\begin{aligned} \min_{\hat{X}} \quad & \mathrm{Tr}[\hat{A}\hat{X}\hat{B}^\top\hat{X}^\top] \\ \text{s.t.} \quad & \hat{X}\hat{X}^\top = I \\ & \hat{X}^\top\hat{X} = I. \end{aligned} \tag{11}$$

The Lagrangian dual of this problem reads [3]:

$$\begin{aligned} \max_{\hat{S},\hat{T}} \quad & \mathrm{Tr}[\hat{S} + \hat{T}] \\ \text{s.t.} \quad & \hat{Q} \succeq 0, \end{aligned} \tag{12}$$

where

$$\hat{Q} := (\hat{B}\otimes\hat{A}) - (I\otimes\hat{S}) - (\hat{T}\otimes I), \quad \hat{S} = \hat{S}^\top, \quad \hat{T} = \hat{T}^\top.$$

Here $\hat{Q} \succeq 0$ means that $\hat{Q}$ has to be positive semidefinite. The optimal solution for (11), according to (7), is

$$\min_{\hat{X}\in\mathcal{O}(n-1)} \mathrm{Tr}[\hat{A}\hat{X}\hat{B}^\top\hat{X}^\top] = \langle\lambda(\hat{A}), \lambda(\hat{B})\rangle. \tag{13}$$

The duality gap between the optimal solutions of (11) and (12) is zero since interior points exist for both problems (see, e.g., [44]). Hence, the optimal values are the same:

$$\max_{\hat{S},\hat{T}} \mathrm{Tr}[\hat{S} + \hat{T}] = \langle\lambda(\hat{A}), \lambda(\hat{B})\rangle. \tag{14}$$

The objective function in (11) can be reformulated as follows:

$$\begin{aligned} \mathrm{Tr}[\hat{A}\hat{X}\hat{B}^\top\hat{X}^\top] &= \mathrm{vec}(\hat{X})^\top(\hat{B}\otimes\hat{A})\mathrm{vec}(\hat{X}) \\ &= \mathrm{Tr}[(\hat{B}\otimes\hat{A})\mathrm{vec}(\hat{X})\mathrm{vec}(\hat{X})^\top] \\ &= (\hat{B}\otimes\hat{A})\cdot Y, \end{aligned} \tag{15}$$

where

$$Y = \mathrm{vec}(\hat{X})\mathrm{vec}(\hat{X})^\top.$$

For arbitrary matrices $\hat{S}$ and $\hat{T}$ and $\hat{X} \in \mathcal{O}$ the following equations hold:

$$\begin{aligned} \mathrm{Tr}[\hat{S}] &= \mathrm{Tr}[\hat{S}I] = \mathrm{Tr}[\hat{S}\hat{X}^\top\hat{X}] \\ &= \mathrm{Tr}[\hat{X}\hat{S}\hat{X}^\top] = \mathrm{Tr}[I\hat{X}\hat{S}\hat{X}^\top] \\ &= (\hat{S}\otimes I)\cdot Y, \\ \mathrm{Tr}[\hat{T}] &= \mathrm{Tr}[\hat{T}I] = \mathrm{Tr}[\hat{T}\hat{X}I\hat{X}^\top] \\ &= (I\otimes\hat{T})\cdot Y. \end{aligned}$$

Using this, a positive semidefinite form containing $\hat{Q}$ from (12) can be introduced into the objective function $\mathrm{Tr}[\hat{A}\hat{X}\hat{B}^\top\hat{X}^\top]$, if we assume that $\hat{S}$ and $\hat{T}$ are a feasible solution for the dual problem (12):

$$\begin{aligned} &\mathrm{Tr}[\hat{A}\hat{X}\hat{B}^\top\hat{X}^\top] \\ &\quad= (\hat{B}\otimes\hat{A})\cdot Y \\ &\quad= (\hat{B}\otimes\hat{A})\cdot Y + \mathrm{Tr}[\hat{S}] - (\hat{S}\otimes I)\cdot Y + \mathrm{Tr}[\hat{T}] - (I\otimes\hat{T})\cdot Y \\ &\quad= \mathrm{Tr}[\hat{S}+\hat{T}] + [(\hat{B}\otimes\hat{A}) - (I\otimes\hat{S}) - (\hat{T}\otimes I)]\cdot Y \\ &\quad= \mathrm{Tr}[\hat{S}+\hat{T}] + \hat{Q}\cdot Y \\ &\quad= \mathrm{Tr}[\hat{S}+\hat{T}] + \mathrm{vec}(\hat{X})^\top\hat{Q}\mathrm{vec}(\hat{X}). \end{aligned}$$

Choosing $\hat{S}$ and $\hat{T}$ as the optimal solution to (12) we obtain with (14):

$$\mathrm{Tr}[\hat{A}\hat{X}\hat{B}^\top\hat{X}^\top] = \langle\lambda(\hat{A}), \lambda(\hat{B})\rangle + \mathrm{vec}(\hat{X})^\top\hat{Q}\mathrm{vec}(\hat{X}). \tag{16}$$

Finally, substituting this expression as well as all the non-projected variables $\hat{X} = V^\top XV$, etc., into (8), we obtain after an elementary but tedious calculation the quadratic programming bound

$$\begin{aligned} \text{(QPB)} \qquad \mathrm{Tr}[AXB^\top X^\top] &= \langle\lambda(\hat{A}), \lambda(\hat{B})\rangle \\ &\quad + \mathrm{vec}(X)^\top Q\mathrm{vec}(X). \end{aligned} \tag{17}$$

A comparison with (8) shows that now we have just a single term on the right hand side comprising the unknown matrix X and (17) allows the computation of both a lower bound and the corresponding minimizing matrix X. For the *linear* term in (8), minimizing over the set Π (cf. (9)) is equivalent to minimizing over $\mathscr{E}\cap\mathscr{N}$. Accordingly, Anstreicher and Brixius [4] suggest to minimize the quadratic form in (17) over $\mathscr{E}\cap\mathscr{N}$, i.e., to solve the convex quadratic problem:

$$\begin{aligned} \min \quad & \mathrm{vec}(X)^\top Q \mathrm{vec}(X) \\ \text{s.t.} \quad & Xe = X^\top e = e \\ & X \geqslant 0. \end{aligned} \tag{18}$$

Here $X \geqslant 0$ means that all entries of the matrix X have to be non-negative. The following relationship between the bounds (7), (10) and (17) holds [4]:

$$(\mathrm{EVB}) \leqslant (\mathrm{PEVB}) \leqslant (\mathrm{QPB}) \leqslant (\mathrm{QAP}). \tag{19}$$

Consequently the bound (17) computed by convex programming cannot perform worse than the other bounds. The quality of the corresponding solution X in comparison to other approaches (see next section) will be assessed in Section 6.

3.4. Computing a combinatorial solution

To obtain a permutation matrix $P \in \Pi$ from the non-integer solution $X \in \mathscr{E} \cap \mathscr{N}$ to (18), a good permutation matrix close to X has to be found. A simple way of doing this is to solve the following linear programming problem:

$$P_0 = \arg\max_{P \in \Pi} \mathrm{Tr}[X^\top P]. \tag{20}$$

In this paper, we use a slightly different idea which takes into account that in most cases a linear approximation of the original problem leads to an improvement of the obtained objective function. To this end, we add an unknown matrix Δ to the relaxed solution X so as to give a permutation matrix:

$$P = (X + \Delta) \in \Pi.$$

Next, we expand the objective function $\mathrm{Tr}[APB^\top P^\top]$ around X up to linear terms with respect to Δ:

$$\begin{aligned} \mathrm{Tr}[APB^\top P^\top] =& \mathrm{Tr}[A(X+\Delta)B^\top (X+\Delta)^\top] \\ =& \mathrm{Tr}[AXB^\top X^\top] + \mathrm{Tr}[AXB^\top \Delta^\top] + \mathrm{Tr}[A\Delta B^\top X^\top] + \mathrm{Tr}[A\Delta B^\top \Delta^\top] \\ \approx& \mathrm{Tr}[AXB^\top X^\top] + \mathrm{Tr}[AXB^\top \Delta^\top] + \mathrm{Tr}[A\Delta B^\top X^\top] \\ =& -\mathrm{Tr}[AXB^\top X^\top] + \mathrm{Tr}[AXB^\top P^\top] + \mathrm{Tr}[B^\top X^\top AP] \\ =& -\mathrm{Tr}[AXB^\top X^\top] + 2\mathrm{Tr}[B^\top X^\top AP]. \end{aligned}$$

As a result, we have to minimize the term $\mathrm{Tr}[B^\top X^\top AP]$ to obtain the combinatorial solution P from the relaxed solution X. This problem can again be solved by linear programming:

$$P_1 = \arg\min_{P \in \Pi} \mathrm{Tr}[B^\top X^\top AP].$$

To see the difference to (20), we put $M = -B^\top X^\top A$ and finally have

$$P_1 = \arg\max_{P \in \Pi} \mathrm{Tr}[MP]. \tag{21}$$

3.5. The 2opt post-processing heuristics

A simple heuristics called 2opt was proposed in [29] in order to further improve combinatorial solutions computed by more expensive methods. This greedy strategy iteratively exchanges pairs of assignments in the permutation until no further improvement is possible.

4. Non-convex approaches

In this section, we briefly sketch two approaches that we used for comparison with the convex relaxation approach of Section 3.3. The first one was proposed by Umeyama [46] and resembles the spectral relaxation approach of [20]. Furthermore, we consider the deterministic annealing approaches [27] and [29] for which excellent performances are reported in the literature.

4.1. The approach by Umeyama

Based on the *Eigenvalue Bound* (7), Umeyama [46] proposed the following estimate for the solution of (6):

$$X_{\mathrm{Ume}} = \arg\max_{X \in \Pi} \mathrm{Tr}(X^\top |U||V|^\top). \tag{22}$$

Here, U and V diagonalize the adjacency matrices A and B, respectively, with the eigenvalues sorted according to (EVB), and $|\cdot|$ denotes the matrix consisting of the absolute values taken for each element. (22) is a linear assignment problem which can be efficiently solved by using standard methods like linear programming.

4.2. Graduated assignment

Gold and Rangarajan [27] and Ishii and Sato [29] independently developed a technique commonly referred to as *graduated assignment* or *soft assign* algorithm. The set of permutation matrices Π is replaced by the convex set $\mathscr{D} = \mathscr{E} \cap \mathscr{N}$ of positive matrices with unit row and column sums (doubly stochastic matrices). In contrast to previous mean-field annealing approaches, the graduated assignment algorithm enforces hard constraints on row *and* column sums, making it usually superior to other deterministic annealing approaches.

The core of the algorithm is an iteration scheme, which computes an approximative solution matrix X at each step of the decreasing annealing schedule. In our description $\beta > 0$ denotes the current annealing parameter; γ is a fixed "self-amplification" parameter, which enforces that the minimum on the set $\mathscr{D}$ is also in Π. Denoting the iteration time step by the superscript, the matrix $X^{(r+1)}$ is calculated as follows (for β fixed):

$$X_{ij}^{(r+1)} = g_i h_j y_{ij}^{(r)}, \tag{23}$$

with

$$y_{ij}^{(r)} = \exp\left(-\beta \sum_{k,l} \left(A_{ik}B_{jl} + \delta_{ik}\delta_{jl}\gamma\right) X_{kl}^{(r)}\right).$$

The scaling coefficients g_i,h_j are computed so that $X^{(r+1)}$ is projected on the set $\mathscr{D}$ using Sinkhorn's algorithm [27] as inner loop

$$y_{ij}^{(r,2s)} = \frac{y_{ij}^{(r,2s-1)}}{\sum_k y_{ik}^{(r,2s-1)}},$$

$$y_{ij}^{(r,2s+1)} = \frac{y_{ij}^{(r,2s)}}{\sum_k y_{kj}^{(r,2s)}}.$$

Stopping criteria based on convergence bounds or the number of iterations have to be established for the inner projection loop and the iteration scheme. For more details, we refer the reader to [27,29].

Rangarajan et al. [28] showed that this scheme locally converges under mild assumptions. Several studies revealed excellent experimental results. In our experiments, we improved the obtained results with the local 2opt heuristics.

A drawback of the graduated assignment algorithm is that the selection of several "tuning"-parameters is necessary to obtain optimal performance. An annealing schedule has to be set up, which is usually described by three parameters: an initial temperature, the annealing rate, and a final temperature or other stopping criterion [27]. There are theoretically motivated methods that give a lower bound for reasonable initial temperatures based on an analysis of the bifurcation structure of the problem [29]. Nevertheless, careful selection of the parameter greater than this bound can improve the results. The self-amplification parameter also has a lower bound that guarantees the above property that the minimizer of the objective function is in Π. An exhaustive parameter search for the annealing schedule, even below the theoretical bound, may increase the performance. Finally, the stopping criteria also influence the quality of the results. All parameters have in common that their optimal values vary for different problem instances (cf. [29]).

5. Convex relaxation: an illustrative numerical example

For the purpose of illustration, we apply the convex relaxation approach to a small graph matching problem in this section.

5.1. A small graph matching problem

In order to graphically visualize the convex relaxation approach, we consider the two small weighted graphs G and H shown in Fig. 2. Obviously, the best match corresponds to exchanging vertices 2 and 3 in either graph. The adjacency matrices of the graphs G and H are:

$$A_G = \begin{pmatrix} 0 & 0.56 & 0.92 \\ 0.56 & 0 & 0.12 \\ 0.92 & 0.12 & 0 \end{pmatrix},$$

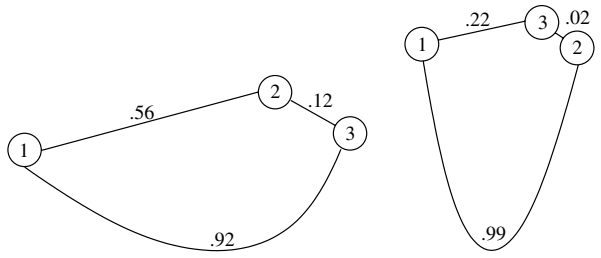

Fig. 2. Two small sample graphs G and H to be matched.

$$A_H = \begin{pmatrix} 0 & 0.99 & 0.22 \\ 0.99 & 0 & 0.02 \\ 0.22 & 0.02 & 0 \end{pmatrix}.$$

In this example the objective function of the graph matching problem (3) attains the following values for each of the six possible permutations:

$$1.370, 3.077, 2.01, 3.365, 0.613, 0.261.$$

Thus, the optimum of this graph matching problem is

$$\mathrm{OPT} = C_G + C_H + 2 \min_{X \in \Pi} \mathrm{Tr}[AXB^\top X^\top] \approx 0.261$$

with $A = A_H$, $B = -A_G$, $C_G = \mathrm{Tr}[A_G A_G^\top] \approx 2.349$ and $C_H = \mathrm{Tr}[A_H A_H^\top] \approx 2.058$. In the following, we visualize permutation matrices X by representing entries $X_{ij} = 1$ graphically by black squares and $X_{ij} = 0$ by white squares. Accordingly, the permutation matrices which lead to the objective function values given above (in the same order) are depicted here

The last permutation matrix represents an exchange of vertices 2 and 3 and thus corresponds to the global optimum of this graph matching problem.

5.2. Relaxations and bounds

We calculate the bounds described in Section 3 for the problem depicted in Fig. 2.

5.2.1. Orthogonal relaxation

The eigenvalue bound leads to the following lower bound:

$$\mathrm{EVB} = C_G + C_H + 2\langle \lambda(A), \lambda(B) \rangle_- \approx 0.023$$

with $C_G = \mathrm{Tr}[A_G A_G^\top] \approx 2.349$, $C_H = \mathrm{Tr}[A_H A_H^\top] \approx 2.058$ and $\langle \lambda(A), \lambda(B) \rangle_- \approx -2.192$. This bound is attained for

$$X \approx \begin{pmatrix} 0.999 & 0.041 & 0.002 \\ -0.014 & 0.316 & 0.948 \\ -0.038 & 0.948 & -0.317 \end{pmatrix}.$$

Due to the small size of our problem, this solution already indicates the optimum despite the obvious weakness of the EVB-bound in general.

5.2.2. Projected eigenvalue bound

Using the projected eigenvalue bound, we obtain the following lower bound for our small graph matching problem:

$$\begin{aligned}\mathrm{PEVB} &= C_G + C_H + 2[\langle\lambda(\hat{A}),\lambda(\hat{B})\rangle_- + \mathrm{LAP}(D) - C_1] \\ &\approx 0.181,\end{aligned}$$

where $\langle\lambda(\hat{A}),\lambda(\hat{B})\rangle \approx -0.985$, $\mathrm{LAP}(D) \approx -2.003$, $C_1 \approx -0.875$. Note that this bound is much stronger than the EVB-bound. On the other hand, as mentioned in Section 3.2, this approach does not allow to compute a corresponding matrix X for which the PEVB-bound is attained.

5.2.3. Quadratic programming bound

The quadratic programming bound gives:

$$\begin{aligned}\mathrm{QPB} &= C_G + C_H + 2[\langle\lambda(\hat{A}),\lambda(\hat{B})\rangle \\ &\quad + \min_{X\in\mathscr{E}\cap\mathscr{N}} \mathrm{vec}(X)^\top Q\mathrm{vec}(X)] \\ &\approx 0.215.\end{aligned}$$

Here the minimization of the quadratic term results in $\min_{X\in\mathscr{E}\cap\mathscr{N}}\mathrm{vec}(X)^\top Q\mathrm{vec}(X) \approx -1.111$ and the bound is attained for

$$X \approx \begin{pmatrix} 0.747 & 0.000 & 0.253 \\ 0.253 & 0.000 & 0.747 \\ 0.000 & 1.000 & 0.000 \end{pmatrix}.$$

As predicted, this bound is superior to the PEVB-bound. So summarizing, for the numerical example considered here the ranking (19) of these bounds reads:

$$\begin{aligned}\mathrm{EVB} \approx 0.023 \leqslant \mathrm{PEVB} \approx 0.181 \leqslant \mathrm{QPB} \approx 0.215 \\ \leqslant \mathrm{OPT} \approx 0.261.\end{aligned} \tag{24}$$

5.3. Visualization

To illustrate how the convex relaxation approximates the original combinatorial problem, we inspect graphically the original objective function

$$f_{\mathrm{orig}}(X) = C_G + C_H + 2\mathrm{Tr}[AXB^\top X^\top]$$

along with its convex relaxation

$$f_{\mathrm{convex}}(X) = C_G + C_H + 2[\langle\lambda(\hat{A}),\lambda(\hat{B})\rangle_- + \mathrm{vec}(X)^\top Q\mathrm{vec}(X)]$$

for a few one-dimensional paths $X(\alpha)$ through the relaxed solution set defined by $X \in \mathscr{E}\cap\mathscr{N}$. It is well-known (Birkhoff–von Neumann theorem) that this set is just the convex hull of the original feasible set, i.e., the permutation matrices $X \in \Pi$. Hence, all paths

$$X(\alpha) = \alpha X_2 + (1-\alpha)X_1, \quad \alpha \in [0,1]$$

between extreme points $X_1, X_2 \in \Pi$ go through the interior relaxed solution set, and we can graphically explore the two cost functions above by plotting their graphs over various paths.

Figs. 3 and 4 show several paths and illustrate the following facts:

- At the end-points of all paths, the two cost functions coincide because the relaxed approach does not change the original objective function at the original feasible set (cf. Section 1.3).
- The original objective function is non-convex in the relaxed solution set and thus exhibits local minima. This is not the case for the objective function of the convex relaxation.

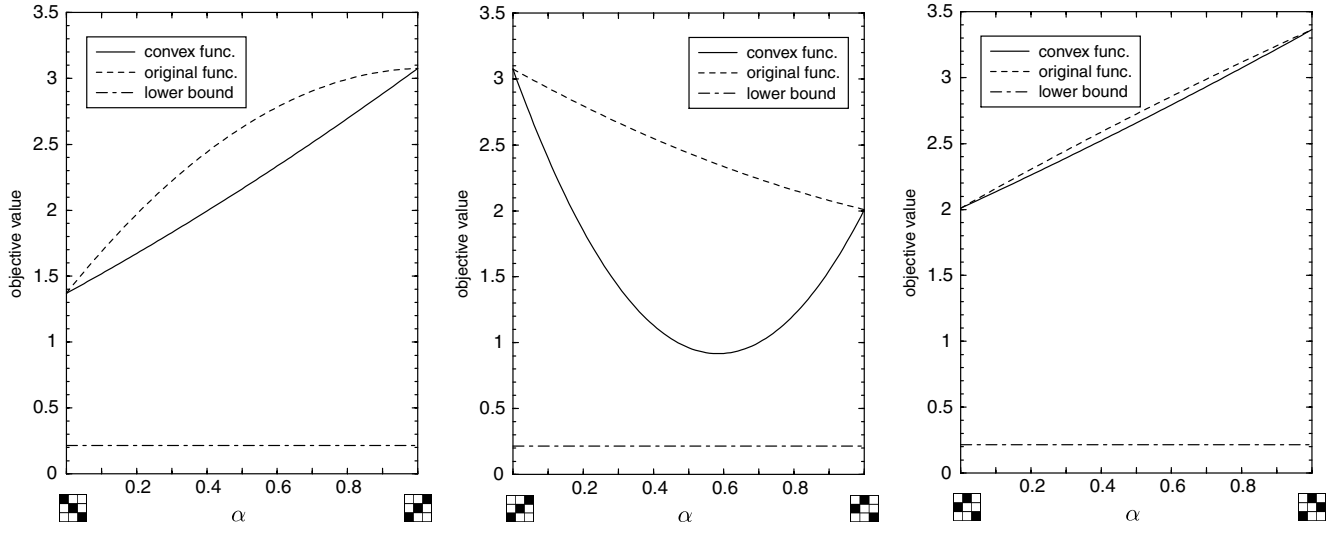

Fig. 3. The original objective function and its convex relaxation along paths through the relaxed solution set.

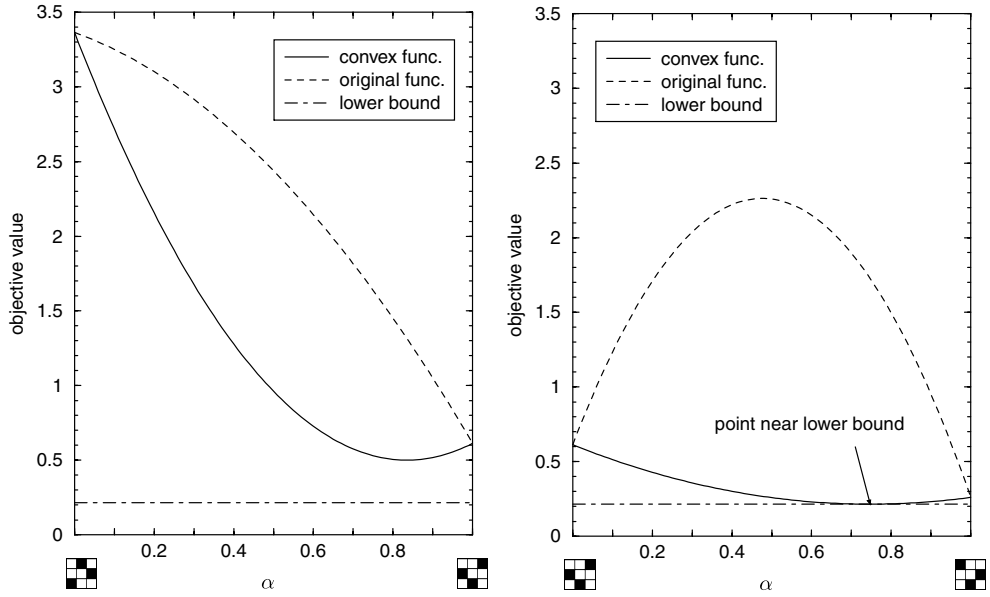

Fig. 4. The original objective function and its convex relaxation along paths through the relaxed solution set. On the right, the plot illustrates how the lower bound is attained at a point close to the global optimum (end-point on the right).

- The plot on the right hand side of Fig. 4 illustrates how the lower bound is attained by the convex relaxation approach. Furthermore, the point where this bound is attained is close to the global optimum (the end-point on the right) due to the tightness of the lower bound.

In summary, these Figures illustrate that the convex relaxation "feels" where the "good" minimum lies, and that a nearby point can be computed without any initialisation problem or parameter tuning!

6. Experiments and discussion

This section has three parts. In the first part we investigate the performance of the convex optimization relaxation. To this end, we compare the corresponding lower bounds with the combinatorial solutions of several benchmark problems from the QAPLIB-collection [45]. The QAPLIB is a public library of very difficult real-life quadratic assignment problems which can be used to evaluate and to compare the performance of any quadratic assignment approach. In the second part we present statistical results computed for a large set of randomly generated graphs (including ground-truth). Finally, in the third part a real world graph matching example is shown.

Abbreviations: The following abbreviations are used within the tables of this section. f represents in all cases the value calculated by the objective function (cf.(6))

$$f(X) = \mathrm{Tr}[AXB^{\top}X^{\top}],$$

with an $X \in \Pi$. The subscript of f shows how and with which approach the $X \in \Pi$ was obtained.

f^*	value of the objective function. (6) at the global optimum $X^* \in \Pi$.
EVB	the eigenvalue bound (7).
PEVB	the projected eigenvalue bound (10).
QPB	the quadratic programming bound (17).
f_{QPB}	value of the objective function in (6) using the permutation matrix obtained with (20) from the QPB solution.
f^1_{QPB}	value of the objective function in (6) using the permutation matrix obtained with (21) from the QPB solution.
f_{GA}	value of the objective function in (6) using the permutation matrix obtained by the graduated assignment algorithm (23).
f_{Ume}	value of the objective function (6) using the permutation matrix obtained by the approach from Umeyama (22).

An additional "+"-sign (e.g., $f_{\mathrm{QPB}}+, f^1_{\mathrm{QPB}}+, f_{\mathrm{GA}}+, f_{\mathrm{Ume}}+$) indicates that the 2opt-heuristics was used as a post-processing step to further improve the permutation matrix found.

6.1. QAPLIB benchmark experiments

6.1.1. Quality of the relaxations

The quality of the various relaxation approaches, namely the eigenvalue bound (EVB), the projected eigenvalue bound (PEVB) and the quadratic programming bound (QPB), can be assessed by measuring how close these bounds are to the global optimum (see (19)).

Table 1 shows the results for problems drawn from the QAPLIB [45]. The first column comprises labels indicating

Table 1
Bounds computed for QAPLIB-problems

Problem	f^*	EVB	PEVB	QPB
chr12c	11,156	−127,514	−24,375	−22,648
chr15a	9896	−190,769	−52,468	−48,539
chr15c	9504	−186,403	−50,295	−47,409
chr20b	2298	−30,995	−8051	−7728
chr22b	6194	−66,432	−22,126	−20,995
esc16b	292	−230	250	250
rou12	235,528	−274,122	200,024	205,461
rou15	354,210	−424,419	296,705	303,487
rou20	725,522	−739,730	597,045	607,362
tai10a	135,028	−181,950	112,528	116,260
tai12a	224,416	−284,261	193,124	199,378
tai15a	388,214	−414,351	325,019	330,205
tai17a	491,812	−496,403	408,910	415,578
tai20a	703,482	−714,901	575,831	584,942
tai30a	1,818,146	−1,505,553	1,500,406	1,517,829
tai35a	2,422,002	−2,015,233	1,941,622	1,958,998
tai40a	3,139,370	−2,559,063	2,484,371	2,506,806

the problem and the number $|V|$ of vertices of a data set from the QAPLIB. The second columns shows the value of the objective function at the global optimum. The corresponding lower bounds computed by the relaxation approaches are listed in the remaining columns.

Since zero is a trivial lower bound, a negative sign indicates that the relaxation is not tight. This happens for most problems with the EV-bound which therefore can be considered not to be useful, and for some problems with the other bounds as well (hence these problems seem to be most difficult).

Furthermore, Table 1 confirms the relationship (19), and that the convex relaxation approach gives the best lower bound.

6.1.2. Comparison to spectral decomposition and graduated assignment

We compare the combinatorial solutions obtained with the convex relaxation approach with those computed with the graduated assignment approach [27,29] and the spectral decomposition approach by Umeyama [46].

Table 2 shows the results in the same way as Table 1, but now only combinatorial solutions are shown. A '+''-sign indicates that the 2opt-heuristics was used as a post-processing step to improve the solution. The difference between f_{QPB} and f^1_{QPB} is that linearization was used to ''round'' the relaxed convex programming solution to a combinatorial solution in the latter case (see Section 3.4).

The columns labeled with f_{GA} and f_{Ume} show the results obtained for the graduated assignment approach [27,29] and for the approach by Umeyama [46]. It should be noted that considerable care was taken to find out optimal parameter values for the graduated assignment approach for each data set [47].

The following conclusions can be drawn from the results shown in Table 2:

- The convex relaxation approach f_{QPB} and the soft-assign approach f_{GA} have similarly good performance, despite the fact that the latter approach is much more intricate from the optimization point-of-view and involves a couple of tuning parameters which have to be (and were) optimized by hand.
- The approach of Umeyama f_{Ume} based on spectral decomposition is not as competitive.
- Using the simple 2opt greedy-strategy as a post-processing step significantly improves the solution in most cases.

6.2. Random ground-truth experiments

In this subsection we discuss our results obtained for two different ground-truth experiments. In the first experiment we created many problem instances (6) by independently computing two different random graphs with the same number of vertices. In the second experiment we com-

Table 2
Results of the QAPLIB benchmark experiments (see text)

Problem	f^*	f_{QPB}	$f_{\mathrm{QPB}}+$	f^1_{QPB}	$f^1_{\mathrm{QPB}}+$	f_{GA}	$f_{\mathrm{GA}}+$	f_{Ume}	$f_{\mathrm{Ume}}+$
chr12c	11,156	20,306	15,860	27,912	13,088	19,014	11,186	40,370	11,798
chr15a	9896	26,132	14,454	20,640	13,540	30,370	11,062	60,986	17,390
chr15c	9504	29,862	17,342	19,436	12,754	23,686	13,342	76,318	13,338
chr20b	2298	6674	2858	7276	3832	6290	2650	10,022	3294
chr22b	6194	9942	6848	8958	6902	9658	6732	13,118	7418
esc16b	292	296	292	312	292	298	292	306	292
rou12	235,528	278,834	246,712	266,864	241,802	273,438	246,282	295,752	251,848
rou15	354,210	381,016	371,480	394,192	374,000	457,908	359,748	480,352	384,018
rou20	725,522	804,676	746,636	795,578	757,270	840,120	738,618	905,246	765,872
tai10a	135,028	165,364	143,260	154,282	139,524	168,096	135,828	189,852	147,838
tai12a	224,416	263,978	237,200	246,424	238,902	263,778	224,416	294,320	252,044
tai15a	388,214	455,778	399,732	432,610	390,782	451,164	400,328	483,596	405,442
tai17a	491,812	550,852	513,170	545,410	526,518	589,814	505,856	620,964	526,814
tai20a	703,482	799,790	740,696	752,896	726,038	871,480	724,188	915,144	775,456
tai30a	1,818,146	1,996,442	1,883,810	1,979,530	1,872,722	2,077,958	1,886,790	2,213,846	1,875,680
tai35a	2,422,002	2,720,986	2,527,684	2,677,688	2,511,800	2,803,456	2,496,524	2,925,390	2,544,536
tai40a	3,139,370	3,529,402	324,3018	341,1278	3,277,450	3,668,044	3,249,924	3,727,478	3,282,284

puted a large collection of random graphs along with slightly perturbed and randomly permutated "copies" of these graphs.

6.2.1. Random graphs

In this experiment we created many problem instances (6) by independently computing two different random graphs with the same number of vertices. The probability that an edge is present in the underlying complete graph was about 0.3. Fig. 5 shows an example in order to visualize the edge-density of such graphs. The global optimum for (6) was computed using an exact search algorithm. The global optimum was used to calculate the ratio of the suboptimal objective value to the best objective value for each problem instance. Table 3 summarizes our results based on this ratio. It shows the statistics (mean, worst case and the best case) for three experiments with different sizes of the graphs ($n = 9,11,15$). The number of problem instances for each experiment is shown in angular brackets. The number of correctly found matchings without/with the 2opt heuristics as post-processing step are shown in round brackets. The following conclusions can be drawn from the results shown in Table 3:

- The soft-assign approach performs somewhat better for these experiments than the convex relaxation approach but the latter needs no tuning parameters which have to be optimized by hand.
- With increasing problem size the performance decreases for all three approaches.
- The approach of Umeyama f_{Ume} based on spectral decomposition is not as competitive.
- Using the simple 2opt greedy-strategy as a post-processing step significantly improves the results.

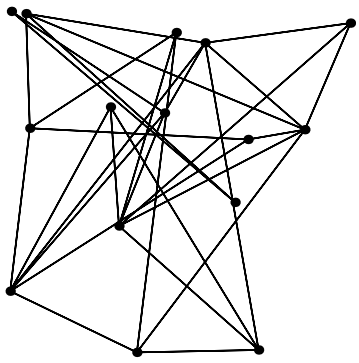

Fig. 5. A randomly generated graph with 15 vertices and a probability of about 0.3 for the presence of an edge.

6.2.2. Perturbed graphs

In the second series of experiments we computed a large collection of random graphs along with slightly perturbed and randomly permutated "copies" of these graphs. The weights of the second graph were perturbed by a normally distributed factor with standard deviation $\sigma = 0.1$ around 1. The results for this kind of experiments are shown in Table 4 which has the same structure as Table 3. For larger problems (more than 15 vertices) where computing the global optimum was too expensive, we assumed the optimal permutation to be the inverse of the random permutation matrix which was used to compute the second graph of each pair. In some cases this was not true and hence a different permutation with a lower objective value could be found by the algorithms. This explains why some of the quotients in Table 4 have a value greater than 1.

In summary, the statistics of our results shown in Table 4 reveal that in almost every case of these "low-level noise" experiments the optimal permutation was found by the quadratic programming approach.

6.3. Real world example

In this section, we show an example for a real world graph matching problem where the nodes of the object graphs are based on features that can be found by an appropriate feature extractor like, for example, the FEX-system (cf. [7,8]). The graphs we want to match are shown in Fig. 1. They have 38 nodes, which means that there is the tremendous number of approximately 10^{44} possible assignments. The result of the graph matching experiment is shown in Fig. 6. The convex relaxation was able to find the expected assignment which is very encouraging because the number of possible assignments is very huge. Furthermore, it should be mentioned that the results shown in this

Table 3
Statistics of the results of random ground-truth experiments (see text)

	f^1_{QPB}/f^*			f_{Ume}/f^*			f_{GA}/f^*		
	Mean	Worst case	Best case	Mean	Worst case	Best case	Mean	Worst case	Best case
$n = 9$ [128]	(22/55)			(7/29)			(31/55)		
	0.88765	0.43810	1	0.638244	0.065173	1	.948342	.7756129	1
2opt	0.97130	0.79256	1	0.928304	0.753007	1	.969914	.843046	1
$n = 11$ [42]	(3/10)			(0/7)			(7/10)		
	0.83043	0.56268	1	0.636159	0.295194	0.998591	.940740	.8338586	1
2opt	0.95760	0.85043	1	0.933206	0.811326	1	.958863	.8434407	1
$n = 15$ [99]	(0/2)			(0/1)			(4/11)		
	0.78726	0.52307	0.938917	0.225983	0.131333	0.863508	.916225	.105164	1
2opt	0.92195	0.77956	1	0.890131	0.74688	1	.95763	.820596	1

Table 4
Statistics of the results of perturbed graph experiments (see text)

	f^1_{QPB}/f^*			f_{Ume}/f^*			f_{GA}/f^*		
	Mean	Worst case	Best case	Mean	Worst case	Best case	Mean	Worst case	Best case
$n = 9$ [155]	(154/155)			(142/154)			(144/151)		
	0.999996	0.999382	1	0.986481	0.463282	1	.997206	.859380	1
2opt	1	1	1	0.999883	0.981862	1	.998154	.859380	1
$n = 15$ [183]	(183/183)			(163/175)			(176/181)		
	1	1	1	0.974078	0.379189	1	.998347	.833787	1
2opt	1	1	1	0.993836	0.718871	1	.998484	.833787	1
$n = 20$ [173]	(173/173)			(148/163)			(167/171)		
	1	1	1	0.977225	0.475711	1	.998205	.855257	1
2opt	1	1	1	0.991662	0.772512	1	.998338	.855257	1
$n = 25$ [169]	(169/169)			(64/123)			(126/143)		
	1.00001	1	1.00155	0.848105	0.216079	1	.966097	.491432	1.001550
2opt	1.00002	1	1.00155	0.960519	0.602629	1.00099	.9748815	.686842	1.001550

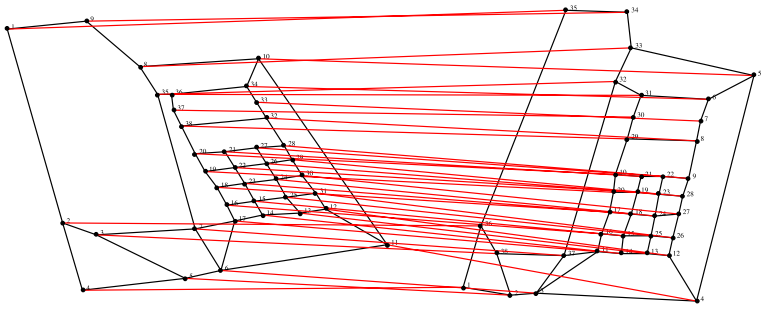

Fig. 6. For the shown object graphs the desired matching is obtained by the convex optimization approach.

section are not further optimized by the 2opt post-processing step and are therefore the results of the convex relaxation approach (17), together with the linear optimization (21) which is a convex problem too.

We added some small structure perturbations to this real world example and the result for the structural perturbed problem is shown in Fig. 7. Nearly the desired matching is found but for the sake of clarity only the two undesired mappings are shown. Note that the full desired matching is very likely obtained by the two-opt post processing step.

The objective function (4) favors matchings which map large edges of the first graph to large edges in the second graph. Therefore, strong perturbations which affect the large weights in the graphs are likely to lead to a combinatorial optimum which corresponds to an undesired matching. But from the viewpoint of computer vision large weights can be expected to involve reliable feature measurements. Therefore large weighted edges are likely to be present in both the object and scene graph.

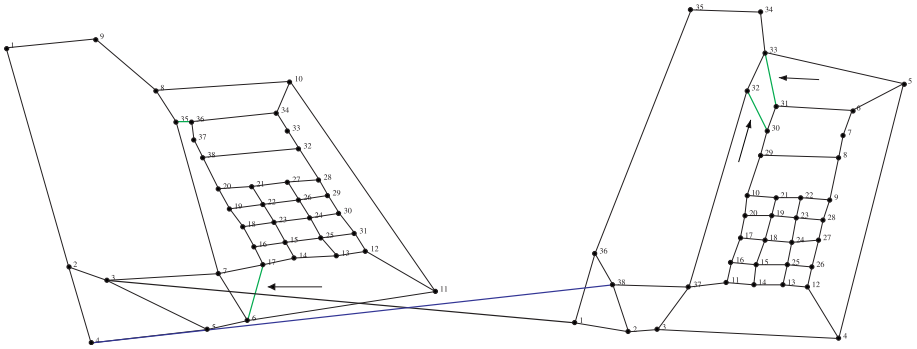

Fig. 7. The matching result obtained with the convex optimization approach. In the graphs additionally to the weights also the structure is perturbed. These perturbations are marked by arrows. Nearly the desired matching is found but for the sake of clarity only the two undesired mappings are shown (compare with Fig. 6). Note that the 2opt post processing leads to the full desired matching.

7. Conclusion and further work

We showed that the convex programming approach is a competitive approach for finding suboptimal solutions to the weighted graph-matching problem. We compared the convex approach with both a recent deterministic annealing approach and an approach based on the eigenvalue decomposition. The performance of the latter approach is worse whereas the deterministic annealing approach performs similarly or slightly better, but uses parameters values which were optimized by hand. The advantage of the convex approach is that no "tuning" parameters have to be determined at all. Furthermore, in contrast to the deterministic annealing approach, the convex approach provides a lower bound and thus can be used as a subroutine within an exact search strategy like branch-and-bound. Our results and the real world example show that it is an attractive direction of research for solving relational matching problems in the context of view-based object recognition.

Towards subgraph matching. Our further work will focus on the case of graphs with an unequal number of vertices: $|V_G| \neq |V_H|$. If this difference is small, our approach can be applied by filling up the smaller graph with "virtual nodes". In general, of course, this is not a satisfying way. The consequence of different numbers of vertices is that the unknown permutation matrix X becomes a matching matrix, and that either of the two constants C_G, C_H in the combinatorial objective function (5) changes to a term which depends on X, too. In our further work we will extend our approach to this more general case. For details we refer to the dissertation [48].

Acknowledgments

Comments on the manuscript by Jens Keuchel and Daniel Cremers are gratefully acknowledged.

References

[1] M. Herbert, J. Ponce, T. Boult, A. Gross, Object Representation in Computer VisionLecture Notes in Computer Science, Springer, New York, 1995.
[2] H. Bülthoff, S. Edelman, Psychophysical support for a two-dimensional view interpolation theory of object recognition, Proc. Nat. Acad. Sci. USA 92 (1992) 60–64.
[3] K. Anstreicher, H. Wolkowicz, On langrangian relaxation of quadratic matrix constraints, SIAM J. Matrix Anal. Appl. 22 (2000) 41–55.
[4] N. Brixius, K. Anstreicher, Solving quadratic assignment problems using convex quadratic programing relaxations, Optimization Methods and Software 16 (1–4) (2001) 49–68.
[5] S. Sahni, T. Gonzalez, P-complete approximation problems, J. Assoc. Comput. Mach. 23 (3) (1976) 555–565.
[6] M.R. Garey, D.S. Johnson, Computers and Intractability, a Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, CA, 1991.
[7] W. Förstner, A framework for low level feature extraction, in: J. Eklundh (Ed.), Computer Vision – ECCV '94, 801, Springer, New York, 1994, pp. 61–70.
[8] W. Förstner, Fex website, http://www.ipb.uni-bonn.de/ipb/projects/fex/fex.html, Institute of Photogrammetry, University of Bonn, Germany.
[9] R.E. Burkard, E. Çela, P.M. Pardalos, L.S. Pitsoulis, The quadratic assignment problem, in: P.M. Pardalos, D.-Z. Du (Eds.), Handbook of Combinatorial Optimization, Kluwer Academic Publishers, Dordrecht, 1998, pp. 241–338.
[10] E.L. Lawler, The quadratic assignment problem, Manag. Sci. 9 (1963) 586–599.
[11] H.A. Almohamad, S. Duffuaa, A linear programming approach for the weighted graph matching problem, IEEE Trans. Patt. Anal. Mach. Intell. 15 (5) (1993) 522–525.
[12] M.S. Bazaraa, H. Sherali, On the use of exact and heuristic cutting plane methods for the quadratic assignment problem, J. Operational Res. Soc. 33 (1982) 991–1003.
[13] P.M. Pardalos, K.G. Ramakrishnan, M.G.C. Resende, Y. Li, Implementation of a variable reduction based lower bound in a branch and bound algorithm fir the quadratic assignment problem, SIAM J. Optim. 7 (1997) 280–294.
[14] P. Gilmore, Optimal and suboptimal algorithms for the quadratic assignment problem, SIAM J. Applied Math. 10 (1962) 305–313.
[15] E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley & Sons, Chichester, 1997.
[16] A. Misevičius, A modified simulated annealing algorithm for the quadratic assignment problem, Informatica, Lith. Acad. Sci. 14 (4) (2003) 497–514.
[17] Z. Drezner, A new genetic algorithm for the quadratic assignment problem, INFORMS J. Comput. 15 (2003) 320–330.
[18] L.M. Gambardella, E.D. Taillard, M. Dorigo, Ant colonies for the quadratic assignment problem, J. Operational Res. Soc. 50 (2) (1999) 167–176.
[19] S. Matsui, Y. Kobayashi, K.Watanabe, Y. Horio, Exponential chaotic tabu search hardware for quadratic assignment problems using switched-current chaotic neuron IC, in: Proc. IJCNN, 2004, pp. 2221–2226.
[20] G. Finke, R. Burkard, F. Rendl, Quadratic assignment problems, Annal. Discrete Mathemat. 31 (1987) 61–82.
[21] S.W. Hadley, F. Rendl, H. Wolkowicz, A new lower bound via projection for the quadratic assignment problem, Math. Operations Res. 17 (1992) 727–739.
[22] R. Kree, A. Zippelius, Recognition of topological features of graphs and images in neural networks, J. Phys. A 21 (1988) L813–L818.
[23] C. Peterson, B. Söderberg, A new method for mapping optimization problems onto neural networks, Int. J. Neural Syst. 1 (1989) 3–22.
[24] A.L. Yuille, Generalized deformable models, statistical physics, and matching problems, Neural Comput. 2 (1990) 1–24.
[25] P. Simic, Constrained nets for graph matching and other quadratic assignment problems, Neural Comput. 3 (2) (1991) 268–281.
[26] J.J. Kosowsky, A.L. Yuille, The invisible hand algorithm: Solving the assignment problem with statistical pyhysics, Neural Networks 7 (3) (1994) 477–490.
[27] S. Gold, A. Rangarajan, A graduated assignment algorithm for graph matching, IEEE Trans. Patt. Anal. Mach. Intell. 18 (4) (1996) 377–388.
[28] A. Rangarajan, A. Yuille, E. Mjolsness, Convergence properties of the softassign quadratic assignment algorithm, Neural Comput. 11 (6) (1999) 1455–1474.
[29] S. Ishii, M. Sato, Doubly constrained network for combinatorial optimization, Neurocomputing 43 (2002) 239–257.
[30] A. Blake, A. Zisserman, Visual Reconstruction, MIT Press, Cambridge, MA, 1987.
[31] Y.G. Leclerc, Constructing simple stable descriptions for image partitioning, Int. J. Comput. Vision 3 (1) (1989) 73–102.
[32] D. Geiger, F. Girosi, Parallel and deterministic algorithms from mrf's: surface reconstruction, IEEE Trans. Patt. Anal. Mach. Intell. 13 (5) (1991) 401–412.
[33] D. Geiger, A. Yuille, A common framework for image segmentation, Int. J. Comput. Vision 6 (3) (1991) 227–243.

[34] T. Hofmann, J. Buhmann, Pairwise data clustering by deterministic annealing, IEEE Trans. Patt. Anal. Mach. Intell. 19 (1) (1997) 1–14.
[35] A.V. Rao, D.J. Miller, K. Rose, A. Gersho, A deterministic annealing approach for parsimonious design of piecewise regression models, IEEE Trans. Patt. Anal. Mach. Intell. 21 (2) (1999) 159–173.
[36] M. Sato, S. Ishii, Bifurcations in mean-field-theory annealing, Phys. Rev. E 53 (5) (1996) 5153–5168.
[37] W.J. Christmas, J. Kittler, M. Petrou, Structural matching in computer vision using probabilistic relaxation, IEEE Trans. Patt. Anal. Mach. Intell. 17 (8) (1995) 749–764.
[38] A.D.J. Cross, R.C. Wilson, E. Hancock, Inexact graph matching using genetic search, Pattern Recog. 30 (6) (1997) 953–970.
[39] B.T. Messmer, H. Bunke, A new algorithm for error-tolerant subgraph isomorphism detection, IEEE Trans. Patt. Anal. Mach. Intell. 20 (5) (1998) 493–504.
[40] B. Luo, E.R. Hancock, Structural graph matching using the em algorithm and singular value decomposition, IEEE Trans. Patt. Anal. Mach. Intell. 23 (10) (2001) 1120–1136.
[41] A.D.J. Cross, E.R. Hancock, Graph matching with a dual-step em algorithm, IEEE Trans. Patt. Anal. Mach. Intell. 20 (11) (1998) 1236–1253.
[42] M. Pelillo, K. Siddiqi, S.W. Zucker, Matching hierarchical structures using association graphs, IEEE Trans. Patt. Anal. Mach. Intell. 21 (11) (1999) 1105–1120.
[43] H. Bunke, Error correcting graph matching: on the influence of the underlying cost function, IEEE Trans. Patt. Anal. Mach. Intell. 21 (9) (1999) 917–922.
[44] Y. Nesterov, A. Nemirovski, Interior Point Polynomial Methods in Convex Programming, SIAM Series in Applied Mathematics, Philadelphia, 1994.
[45] R.E. Burkard, S. Karisch, F. Rendl, Qaplib – a quadratic assignment problem library, J. Global Optimization 10 (1997) 391–403.
[46] S. Umeyama, An eigendecomposition approach to weighted graph matching problems, IEEE Trans. Patt. Anal. Mach. Intell. 10 (5) (1988) 695–703.
[47] Stefan Roth, Analysis of a deterministic annealing method for graph matching and quadratic assignment problems in computer vision. Master's thesis, CVGPR-group, University of Mannheim, May 2001.
[48] C. Schellewald. Convex Mathematical Programs for Relational Matching of Object Views. Ph.D. thesis, Department of Mathematics and Computer Science, University of Mannheim, 2005.