

A Convex-Concave Relaxation Procedure Based Subgraph Matching Algorithm

Zhi-Yong Liu ZHIYONG.LIU@IA.AC.CN and **Hong Qiao** HONG.QIAO@IA.AC.CN
*State Key Laboratory of Management and Control for Complex Systems, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China*

Editor: Steven C. H. Hoi and Wray Buntine

Abstract

Based on the convex-concave relaxation procedure (CCRP), the (extended) path following algorithms were recently proposed to approximately solve the equal-sized graph matching problem, and exhibited a state-of-the-art performance (Zaslavskiy et al., 2009; Liu et al., 2012). However, they cannot be used for subgraph matching since either their convex or concave relaxation becomes no longer applicable. In this paper we extend the CCRP to tackle subgraph matching, by proposing a convex as well as a concave relaxation of the problem. Since in the context of CCRP, the convex relaxation can be viewed as an initialization of a concave programming, we introduce two other initializations for comparison. Meanwhile, the graduated assignment algorithm is also introduced in the experimental comparisons, which witness the validity of the proposed algorithm.

Keywords: subgraph matching, convex relaxation, concave relaxation, feature correspondence

1. Introduction

Subgraph matching as a fundamental problem in theoretical computer science finds wide applications in many fields such as computer vision, machine learning and bioinformatics. It is also sometimes referred to as subgraph isomorphism, but here we consider the inexact subgraph matching or subgraph homomorphism problem, which provides a basis for many real applications, such as pattern/object matching, feature correspondence, and weighted subgraph matching. Since the problem has been proven to be NP-complete (Abdulkader, 1998), in general we have to resort to some approximate approaches to get a trade-off between the complexity and matching accuracy.

Though there existed some quite successful algorithms (Ullman, 1976; Haralick and Elliot, 1980) targeting at subgraph isomorphism, they can hardly be extended to solve weighted subgraph matching problems. On the other hand, an important family of approximate subgraph matching algorithms arises from the relaxation technique, which usually involves relaxing the discrete matching problem to be a continuous one. The point lies in the fact that a continuous problem is usually more convenient to be approximated than its discrete counterpart. Typical relaxation based subgraph matching algorithms include, for instance, probabilistic relaxation (Christmas et al., 1995) and convex relaxation (Schellewald and Schnörr, 2005).

Actually, relaxation technique based graph matching algorithms have been receiving extensive attentions in the past three decades (Fischler and Elschlager, 1973; Luo and Hancock, 2001; Umeyama, 1988; Gold and Rangarajan, 1996; Zaslavskiy et al., 2009; Liu et al., 2012). A common problem faced by the relaxation techniques is the back projection, which involves projecting the relaxed continuous solution back to a discrete one. A frequently used approach is based on the maximal linear assignment principle, closely related to the winner-take-all schema, which may however introduce significant additional errors. Some more successful algorithms such as the graduated assignment (Gold and Rangarajan, 1996) and (extended) path following algorithms (Zaslavskiy et al., 2009; Liu et al., 2012) gradually approach a discrete solution, via a deterministic annealing or convex-concave relaxation procedure (CCRP). However, it is not trivial to extend a graph matching algorithm (with equal size) to be a subgraph matching one, by, for instance, adding some dummy nodes into the smaller graph. There are two issues to be considered. Firstly, such an addition expands the search space of the problem ($N_S!/(N_S - N_M)! \rightarrow N_S!$), and thus increases the chance of getting trapped into poor local minima, especially as $N_S \gg N_M$, where N_S and N_M denote the size of the bigger and smaller graphs. Secondly, and maybe more importantly, it is easy to check that adding dummy nodes makes the matching problem, in general, not equivalent to the original one, or in other words, a good solution of the expanded problem may result in a poor solution for the original problem, depending on the objective functions selected.

In this paper we will propose a CCRP based algorithm directly targeting at subgraph matching problems. The proposed method can be taken as an extension of the (extended) path following algorithms (Zaslavskiy et al., 2009; Liu et al., 2012), both of which are based on CCRP, but can be used on only equal-sized graph matching problems. The next section gives a brief description of the subgraph matching problem and the convex-concave relaxation procedure, and the proposed method is detailed in section 3. After giving some experimental comparisons in section 4, section 5 concludes the paper.

2. Problem Description and Convex-Concave Relaxations Procedure

2.1. Problem Description

The objective function of subgraph matching can be defined from different perspectives. For instance, three types of generalization from equal-sized matching problem to subgraph matching problem were discussed in (Zaslavskiy et al., 2009), including the expansion scheme discussed previously. Actually, what we are interested in this paper is the first case in (Zaslavskiy et al., 2009), which matches the smaller graph directly to a subgraph in the bigger one without needing to modify the adjacency matrices. However, the (extended) path following algorithm in (Zaslavskiy et al., 2009; Liu et al., 2012) cannot tackle such a subgraph problem because as will be shown below the transform matrix P is no longer a permutation matrix. Specifically, given the two graph models G_M and G_S to be matched, the objective function for the subgraph matching problem is defined by

$$F(P) = \| A_M - P A_S P^T \|_F^2, P \in \Omega, \tag{1}$$

where the problem domain Ω is defined as

$$\Omega = \{P | P_{ij} = \{0, 1\}, \sum_j P_{ij} = 1, \sum_i P_{ij} \leq 1, \forall i, j\}. \tag{2}$$

In (1), $A_M \in \mathbb{R}^{N_M \times N_M}$ and $A_S \in \mathbb{R}^{N_S \times N_S}$ denote the adjacency matrices, and without loss of generality, it is here and hereafter assumed that $N_M \leq N_S$, which implies that P is a partial permutation matrix since $PP^T = I$, but $P^T P \neq I$ in general. Actually, (1) was also adopted as the objective function for the equal-sized graph matching problem in (Zaslavskiy et al., 2009; Liu et al., 2012) where P is constrained as a permutation matrix. However, generalization of P from permutation to partial permutation matrix makes the subgraph matching problem greatly different from the equal-sized problem, especially in the context of CCRP, as to be discussed in 2.2.

To globally solve the above integer programming, an exhaustive search involves $C_{N_S}^{N_M}$ times of enumerations, which is a typical NP-hard problem.

2.2. Convex-Concave Relaxations Procedure

To utilize the CCRP to approximately solve an (integer) optimization problem like (1), we need first to relax the feasible domain the problem from a discrete set to a (convex) continuous one. For instance, the set of $(N_M \times N_S)$ partial permutation matrices in (2) can be relaxed to the set of $(N_M \times N_S)$ doubly substochastic matrices as follows,

$$\Pi = \{P | P_{ij} \geq 0, \sum_j P_{ij} = 1, \sum_i P_{ij} \leq 1, \forall i, j\}. \quad (3)$$

It is easy to check that Π is the convex hull of Ω (Maciel and Costeira, 2003). Then, we need to figure out both of its convex and concave relaxations. A convex relaxation (underestimates) $F_v(P)$ of $F(P)$ is a convex function that satisfies

$$F_v(P) = F(P), P \in \Omega, \text{ and } F_v(P) \leq F(P), P \in \Pi, \quad (4)$$

and on the other hand, a concave relaxation (overestimates) $F_c(P)$ of $F(P)$ is a concave function that satisfies

$$F_v(P) = F(P), P \in \Omega, \text{ and } F_c(P) \geq F(P), P \in \Pi. \quad (5)$$

Finally, based on $F_v(P)$ and $F_c(P)$, the objective function of a CCRP is constructed as follows (Zaslavskiy et al., 2009),

$$F_\gamma(P) = (1 - \gamma)F_v(P) + \gamma F_c(P), 0 \leq \gamma \leq 1, P \in \Pi. \quad (6)$$

In implementation, γ is increased gradually from 0 to 1, making $F_\gamma(P)$ becomes gradually from $F_v(x)$ to $F_c(X)$. The global minimum of $F_v(P)$ can be found, which provides the initialization of CCRP, and finally, minima of $F_c(P)$ are located exactly in Ω , the feasible domain of the original integer optimization problem. Such a procedure is in spirit similar to the graduated non-convexity algorithm (Blake and Zisserman, 1987) in that both methods approximate the global minimum of a non-convex function by first solving a simple convex program, and then gradually transferring to the original problem.

The CCRP was firstly proposed to solve the equal-sized graph matching problem, where P is constrained as a permutation matrix, i.e., both $PP^T = I$ and $P^T P = I$, which makes the convex relaxation of (1) directly obtainable as (Zaslavskiy et al., 2009; Liu et al., 2012),

$$F_v(P) = \|A_M - PA_S P^T\|_F^2 = \|A_M P - PA_S\|_F^2 = \text{vec}(P)^T Q \text{vec}(P), \quad (7)$$

where $Q = (I \otimes A_M - A_S^T \otimes I)^T (I \otimes A_M - A_S^T \otimes I)$ is a positive definite matrix. Meanwhile, though involving a complicated mathematical derivation, the concave relaxation is also obtainable by taking advantage of P being a permutation matrix (Zaslavskiy et al., 2009; Liu et al., 2012). However, as P is generalized to be a partial permutation matrix, the convex or concave relaxation found on equal-sized problem is no longer applicable. Below we will propose the convex and concave relaxations, and consequently the CCRP algorithm for the subgraph matching problem.

3. Proposed Method

A concave as well as a convex relaxation is firstly proposed to construct the CCRP, then a frank-wolfe based optimization algorithm is proposed to minimize the objective function, and finally some discussions will be given on the convex relaxation.

3.1. Concave and Convex Relaxations

Since P is not a permutation matrix, (7) is no longer satisfied. Instead, by removing a constant with respect to P , the objective function $F(P)$ in (1) can be equivalently written as,

$$F_0(P) = \text{tr}(P^T P A_S^T P^T P A_S) - 2\text{tr}(A_M^T P A_S P^T). \quad (8)$$

Then, we first discuss undirected graphs which involves a symmetric adjacency matrix, and then extend the results to directed graphs.

3.1.1. UNDIRECTED GRAPHS

As $P \in \Omega$, by introducing two constants c_1 and c_2 the objective function $F_0(P)$ above is derived as follows,

$$\begin{aligned} F_u(P; c_1, c_2) &= \text{tr}(P^T P A_S^T P^T P A_S) - 2\text{tr}(A_M^T P A_S P^T) \\ &= \text{tr}(P^T P A_S^T P^T P A_S) - 2\text{tr}(A_M^T P A_S P^T) - (c_1 + c_2) \underbrace{\text{tr} J P}_{N_M} \\ &\quad + c_1 \underbrace{\text{vec}(P^T P)^T \text{vec}(P^T P)}_{N_M} + c_2 \underbrace{\text{vec}(P)^T \text{vec}(P)}_{N_M} \end{aligned} \quad (9)$$

where $J := \mathbf{1}_{N_M \times N_M}$ denotes a unit matrix consisting of all 1s. Because A_S and A_M are both symmetric since both graphs are undirected, implying further that both $(A_S \otimes A_S)$ and $(A_S \otimes A_M)$ are also symmetric, the first two terms of $F_u(P; c_1, c_2)$ can be written in an eigen-decomposed form as,

$$\begin{aligned} \text{tr}(P^T P A_S^T P^T P A_S) &= \text{vec}(P^T P)^T (A_S \otimes A_S) \text{vec}(P^T P) = \text{vec}(P^T P)^T (U_1^T \Lambda_1 U_1) \text{vec}(P^T P), \\ \text{tr}(A_M^T P A_S P^T) &= \text{vec}(P)^T (A_S \otimes A_M) \text{vec}(P) = \text{vec}(P)^T (U_2^T \Lambda_2 U_2) \text{vec}(P), \end{aligned}$$

where $A_S \otimes A_S = U_1^T \Lambda_1 U_1$ and $A_S \otimes A_M = U_2^T \Lambda_2 U_2$ are eigen-decompositions. Finally, $F_u(P; c_1, c_2)$ can further written as

$$\begin{aligned} F_u(P; c_1, c_2) &= \text{vec}(P^T P)^T U_1^T (\Lambda_1 + c_1 I) U_1 \text{vec}(P^T P) + \text{vec}(P)^T U_2^T (c_2 I - 2\Lambda_2) U_2 \text{vec}(P) \\ &\quad - (c_1 + c_2) \text{tr} J P. \end{aligned} \quad (10)$$

Relaxing the feasible domain from Ω to Π , $F_u(P; c_1, c_2)$ becomes a concave relaxation of $F_u(P)$ by setting c_1 and c_2 as

$$c_1 = \lambda_u^{c1} := -\max\{\Lambda_1\}, c_2 = \lambda_u^{c2} := 2\min\{\Lambda_2\},$$

where $\max\{\Lambda\}$ and $\min\{\Lambda\}$ denotes finding the maximal and minimal elements of Λ , respectively. It is here and hereafter assumed that $\max\{\Lambda_1\} > 0$ and $\min\{\Lambda_2\} < 0$, or otherwise, c_1 or c_2 can be just set as 0. Since the matrices $U_1^T(\Lambda_1 + \lambda_u^{c1}I)U_1$ and $U_2^T(\lambda_u^{c2}I - 2\Lambda_2)U_2$ are both symmetric and negative definite, and the linear term $\text{tr}JP$ does not affect the convexity or concavity, $F_u(P; \lambda_u^{c1}, \lambda_u^{c2})$ is a concave function. To verify that $F_u(P; \lambda_u^{c1}, \lambda_u^{c2})$ is meanwhile a concave relaxation, we need further to validate whether $F_u(P; \lambda_u^{c1}, \lambda_u^{c2})$ satisfies the two equations in (4). The first equation is obviously satisfied due to (9). On the other hand, it is easy to check that, as $P \in \Pi$, both $\text{tr}JP = N_M \geq \text{vec}(P^T P)^T \text{vec}(P^T P)$ and $\text{tr}JP = N_M \geq \text{vec}(P)^T \text{vec}(P)$, and meanwhile, both $\lambda_u^{c1} < 0$ and $\lambda_u^{c2} < 0$. Thus, $F_u(P; \lambda_u^{c1}, \lambda_u^{c2}) \geq F_u(P)$. Consequently, it is verified that $F_u(P; \lambda_u^{c1}, \lambda_u^{c2})$ is a concave relaxation of $F_u(P)$.

Moreover, since the term $\text{tr}JP$ remains a constant (N_M) for $P \in \Pi$, $F_u(P; c_1, c_2)$ above can be equivalently simplified as follows,

$$F_u(P; c_1, c_2) = \text{vec}(P^T P)^T U_1^T (\Lambda_1 + c_1 I) U_1 \text{vec}(P^T P) + \text{vec}(P)^T U_2^T (c_2 I - 2\Lambda_2) U_2 \text{vec}(P). \quad (11)$$

Similarly, we can get a convex relaxation $F_u(P; \lambda_u^{v1}, \lambda_u^{v2})$ based on $F_u(P; c_1, c_2)$ by setting

$$c_1 = \lambda_u^{v1} := -\min\{\Lambda_1\}, c_2 = \lambda_u^{v2} := 2\max\{\Lambda_2\}.$$

$\lambda_u^{c1}, \lambda_u^{c2}, \lambda_u^{v1}$ and λ_u^{v2} are straightforward to gotten, due to the fact that any eigenvalue of $A \otimes B$ is the product of an eigenvalue of A ($\lambda_i(A)$) and B ($\lambda_j(B)$) (Golub and Van Loan, 1996). For instance, denoting by λ_{min}^S and λ_{max}^S the minimal and maximal eigenvalue of A_S , and assuming that $\lambda_{max}^S > 0$ and $\lambda_{min}^S < 0$, λ_u^{c1} and λ_u^{v1} can be calculated as $\lambda_u^{c1} = -\max\{\lambda_{min}^S, \lambda_{max}^S\}$, and $\lambda_u^{v1} = -\lambda_{min}^S \lambda_{max}^S$.

3.1.2. DIRECTED GRAPHS

We then discuss the directed graphs, where asymmetric $A_S \otimes A_S$ and/or $A_S \otimes A_M$ make the convexity and concavity of the relaxed function difficult to check. Thus, the function is first equivalently rewritten taking a symmetric form as follows,

$$\begin{aligned} F_d(P) &= \text{vec}(P^T P)^T (A_S \otimes A_S) \text{vec}(P^T P) - 2\text{vec}(P)^T (A_S \otimes A_M) \text{vec}(P) \\ &= \frac{1}{2} \text{vec}(P^T P)^T (A_S \otimes A_S + A_S^T \otimes A_S^T) \text{vec}(P^T P) \\ &\quad - \text{vec}(P)^T (A_S \otimes A_M + A_S^T \otimes A_M^T) \text{vec}(P) \end{aligned} \quad (12)$$

Now, $(A_S \otimes A_S + A_S^T \otimes A_S^T)$ and $(A_S \otimes A_M + A_S^T \otimes A_M^T)$ become symmetric, and a similar derivation as that on undirected graphs can be proceeded to get

$$\begin{aligned} F_d(P; c_1, c_2) &= \frac{1}{2} \text{vec}(P^T P)^T (A_S \otimes A_S + A_S^T \otimes A_S^T) \text{vec}(P^T P) \\ &\quad - \text{vec}(P)^T (A_S \otimes A_M + A_S^T \otimes A_M^T) \text{vec}(P) \\ &\quad + c_1 \text{vec}(P^T P)^T \text{vec}(P^T P) + c_2 \text{vec}(P)^T \text{vec}(P). \end{aligned} \quad (13)$$

Then, a concave relaxation $F_d(P; \lambda_d^{c1}, \lambda_d^{c2})$ and convex relaxation $F_d(P; \lambda_d^{v1}, \lambda_d^{v2})$ can be obtained by setting

$$\lambda_d^{c1} := -\frac{1}{2}\max\{\Lambda_1\}, \lambda_d^{c2} := \min\{\Lambda_2\}, \lambda_d^{v1} := -\frac{1}{2}\min\{\Lambda_1\}, \lambda_d^{v2} := \max\{\Lambda_2\},$$

where Λ_1 and Λ_2 are gotten by the eigen-decompositions $(A_S \otimes A_S + A_S^T \otimes A_S^T) = U_1^T \Lambda_1 U_1$ and $(A_S \otimes A_M + A_S^T \otimes A_M^T) = U_2^T \Lambda_2 U_2$, respectively.

Another issue on the directed graphs is how to figure out the minimal and maximal element of Λ_1 and Λ_2 , the eigenvalues of the two matrices with large sizes $N_S^2 \times N_S^2$ and $N_S N_M \times N_S N_M$ respectively. It is usually computationally prohibited to directly calculate them. In practice we may resort to some iterative technique such as the power method (Golub and Van Loan, 1996), but we notice that an exact maximal or minimal eigenvalue is unnecessary but we need only to estimate an upper/lower bound of them, a similar problem faced by the extended path following algorithm (Liu et al., 2012). Therefore, we adopt the same procedure in (Liu et al., 2012) to calculate the upper/lower bound. Specifically, letting a and b denote the minimal and maximal elements of matrix M (with size $n \times n$), a looser lower bound λ_l of its smallest eigenvalue is gotten by (Zhan, 2006):

$$\lambda_{min}(M) \geq \begin{cases} \begin{cases} n(a-b)/2 & \text{if } n \text{ is even,} \\ (na - \sqrt{a^2 + (n^2 - 1)b^2})/2 & \text{otherwise,} \end{cases} & \text{if } |a| < b, \\ na & \text{otherwise,} \end{cases} \quad (14)$$

Then, a positive definite matrix Φ is constructed as $\Phi = M - \lambda_l I$. Writing Φ_n in a partition form as $\Phi_n = \begin{bmatrix} \Phi_{n-1} & \mathbf{d} \\ \mathbf{d}^T & c \end{bmatrix}$ a tight lower bound for the smallest eigenvalue of Φ can be iteratively gotten by (Dembo, 1988),

$$l_n = \frac{1}{2} \left(c + l_{n-1} - \sqrt{(c - l_{n-1})^2 + 4\mathbf{d}^T \mathbf{d}} \right) \leq \lambda_{min}(\Phi_n), \quad (15)$$

where l_{n-1} denotes the lower bound of Φ_{n-1} . It is not difficult to check that all of $\Phi_{n-1} \dots \Phi_1$ are symmetric positive definite matrices, thanks to the rather general lower bound given by (14). Thus, we can iteratively find l_n starting from l_1 , and consequently, a tight estimation of the smallest eigenvalue can be obtained by $\lambda = l_n + \lambda_l$. The upper bound of the biggest eigenvalue can also be gotten in a similar way.

3.2. CCRP and Algorithm

Based on the convex and concave relaxations found above, the objective function of CCRP can be then constructed as,

$$\begin{aligned} F_\gamma(P) &= (1 - \gamma)F_*(P; \lambda_*^{v1}, \lambda_*^{v2}) + \gamma F_*(P; \lambda_*^{c1}, \lambda_*^{c2}), 0 \leq \gamma \leq 1, P \in \Pi, \\ &= \text{tr}(P^T P A_S^T P^T P A_S) - 2\text{tr}(A_M^T P A_S P^T) + ((1 - \gamma)\lambda_*^{v1} + \gamma\lambda_*^{c1})\text{tr}P^T P P^T P \\ &\quad + ((1 - \gamma)\lambda_*^{v2} + \gamma\lambda_*^{c2})\text{tr}P^T P, \end{aligned} \quad (16)$$

where $*$ = $\{u, d\}$ denotes the undirected or directed graphs.

Algorithm 1 CCRP Subgraph Matching Algorithm

Input: A_M, A_S
Output: P
Initialize: $\gamma \leftarrow 0, P \leftarrow 1_{N_M \times N_S} / N_S$
repeat
 repeat
 $X = \arg \min \text{tr} X^T \nabla F_\gamma(P)$, s.t. $X \in \Pi$
 $\alpha = \arg \min F_\gamma(P + \alpha(X - P))$, s.t. $0 \leq \alpha \leq 1$
 $P \leftarrow P + \alpha(X - P)$
 until P converges
 $\gamma \leftarrow \gamma + d\gamma$
until $\gamma \geq 1 \vee P \in \Omega$

Then, similar to the (extended) path following algorithms, for each fixed γ a frank-wolfe algorithm is employed to minimize the objective function. The algorithm is summarized in Algorithm 1. In the algorithm, $X = \arg \min \text{tr} X^T \nabla F_\gamma(P)$ can be solved by the Hungarian algorithm (Kuhn, 1955), and α can be found by a backtracking algorithm (Boyd and Vandenberghe, 2004).

$\nabla F_\gamma(P)$ in the algorithm is given by:

$$\begin{aligned} \nabla F_\gamma(P) = & 2P(A_S^T P^T P A_S + A_S P^T P A_S^T) - 2(A_M P A_S^T + A_M^T P A_S) + \\ & 4((1 - \gamma)\lambda_*^{v1} + \gamma\lambda_*^{c1})PP^T P + 2((1 - \gamma)\lambda_*^{v2} + \gamma\lambda_*^{c2})P. \end{aligned} \quad (17)$$

For each fixed γ , the convergence of P can be confirmed by checking whether

$$\text{tr} \nabla F_\gamma(P)^T (P - X) < \varepsilon |F_\gamma(P) + \text{tr} \nabla F_\gamma(P)^T (X - P)|. \quad (18)$$

The computational complexity of the proposed algorithm is roughly $O(N_S^3)$ due to matrix multiplication, and the storage is $O(N_S^2)$. However, on directed graphs, both computational and storage complexities become roughly $O(N_S^4)$ due to (15).

3.3. on Convex Relaxation

Because all of the minima of a concave function locate exactly in Ω , a concave relaxation gotten by taking advantage of the properties of $P \in \Omega$ such as $\text{tr} P^T P = N_M$ will retain the same minima as the original discrete matching problem. However, there is a quite different situation for convex relaxation, whose minima are in general located in Π but not Ω . In other words, the minima of the convex relaxation may deviate greatly from the originally relaxed problem. From another viewpoint, the CCRP can be in some sense regarded as a concave programming to approach the global minimum of the concave relaxation, i.e., the original problem, and the convex relaxation provides a reasonable initialization (as well as some constraints during the procedure). Therefore, to evaluate the proposed convex relaxation, it is worth comparing it with some other types of initializations (may even not be a convex relaxation). Below we introduce two other initializations.

First, the smaller graph G_M is expanded to the size N_S by adding some dummy (null) nodes. The new \hat{P} becomes exactly a permutation matrix which implies that $\hat{P}^T \hat{P} = I$.

Then, similar to the path following algorithm (Zaslavskiy et al., 2009), a convex relaxation is gotten as follows,

$$\hat{F}_v^2(\hat{P}) = \text{tr} \left[(\hat{A}_M \hat{P} - \hat{P} A_S)^T (\hat{A}_M \hat{P} - \hat{P} A_S) \right] = \text{vec}(\hat{P})^T \hat{Q} \text{vec}(\hat{P}), \quad (19)$$

where $\hat{Q} = (I \otimes \hat{A}_M - A_S^T \otimes I)^T (I \otimes \hat{A}_M - A_S^T \otimes I)$ is a positive definite matrix. Then, we approximately use the function

$$F_v^2(P) = \text{vec}(P)^T Q \text{vec}(P), \quad (20)$$

as the convex relaxation, where $Q = (I \otimes A_M - A_S^T \otimes I)^T (I \otimes A_M - A_S^T \otimes I)$ is also positive definite. Thus, the objective function becomes

$$F_\gamma^2(P) = (1 - \gamma) F_v^2(P) + \gamma F_*(P; \lambda_*^{c1}, \lambda_*^{c2}), 0 \leq \gamma \leq 1, P \in \Pi, \quad (21)$$

and its gradient $F_\gamma^2(P)$ is given as follows,

$$\begin{aligned} \nabla F_\gamma^2(P) = & 2(1 - \gamma)(A_M^T A_M P + P A_S A_S^T) - 2A_M P A_S^T - 2A_M^T P A_S + \\ & 2\gamma [P(A_S^T P^T P A_S + A_S P^T P A_S^T) + 2\lambda_*^{c1} P P^T P + \lambda_*^{c2} P] \end{aligned} \quad (22)$$

Second, since as discussed above the minima of a convex relaxation may greatly deviate from those of the originally relaxed problem, we directly use the relaxed function as the initialization, though itself is in general no longer a convex function. The objective function is now

$$F_\gamma^3(P) = (1 - \gamma) F_0(P) + \gamma F_*(P; \lambda_*^{c1}, \lambda_*^{c2}), 0 \leq \gamma \leq 1, P \in \Pi, \quad (23)$$

and the gradient becomes

$$\nabla F_\gamma^3(P) = 2(A_M^T A_M P - A_M^T P A_S - A_M P A_S^T + P A_S A_S^T) + 4\gamma \lambda_*^{c1} P P^T P + 2\gamma \lambda_*^{c2} P. \quad (24)$$

4. Experiments and Discussions

In this section we extensively compare the proposed CCRP based subgraph matching algorithm, denoted by CCRP1 below, with some other typical algorithms. Specifically, the other two types of CCRP algorithm with their convex relaxations given by (21) and (23) are denoted by CCRP2 and CCRP3, respectively. Meanwhile, we will also introduce as below a graduated assignment algorithm abbreviated as GA_S to approximately minimize (8). Three other algorithms include the path following (PATH) (Zaslavskiy et al., 2009) and extended path following (EPATH) algorithms (Liu et al., 2012), for undirected graphs and directed graphs respectively, and the original graduated assignment algorithm (GA) (Gold and Rangarajan, 1996). All of the (E)PATH and GA algorithms are implemented by expanding G_M with the same size as G_S by adding some dummy nodes, and finally their matching errors are calculated via (8) by truncating the estimated permutation matrix P from the size $N_S \times N_S$ to a partial permutation matrix with the size $N_M \times N_S$. The parameters in GA and GA_S are set the same as those in (Gold and Rangarajan, 1996), for the CCRP algorithms, ε is set as 0.001, and the initialization of all of the algorithms except for U is set as $P^0 = 1_{N_M \times N_S} / N_S$.

Algorithm 2 Graduated Assignment Subgraph Matching Algorithm

Input: A_M, A_S
Output: P
Initialize: $\beta \leftarrow \beta_0, P \leftarrow 1_{N_M \times N_S} / N_S$
repeat
 repeat
 $Q \leftarrow -\nabla F_0(P)$
 $P^0 \leftarrow \exp(\beta Q)$
 repeat
 $P_{ai}^1 \leftarrow P_{ai}^0 / \sum_i P_{ai}^0$
 $P_{ai}^0 \leftarrow P_{ai}^1 / \sum_a P_{ai}^1$
 until P converges
 $P \leftarrow N_M P / N_S$
 until P converges
 $\beta \leftarrow \beta_r \beta$
until $\beta > \beta_f$

4.1. a graduated assignment subgraph matching algorithm

Firstly, to get a comparative evaluation on the proposed CCRP based subgraph matching algorithm, based on the graduated assignment schema we propose another algorithm to approximately minimize (8). Specifically, given an initialization P^0 , $F_0(P)$ can be approximated around P^0 via a Taylor series as:

$$F_0(P) \approx F_0(P^0) + \text{tr}(P - P^0)^T \nabla F_0(P), \quad (25)$$

where $\nabla F_0(P)$ is given by

$$\nabla F_0(P) = 2P(A_S^T P^T P A_S + A_S P^T P A_S^T) - 2(A_M P A_S^T + A_M^T P A_S). \quad (26)$$

Therefore, minimization of the Taylor series of $F_0(P)$ is approximated by maximizing $-\text{tr} P^T \nabla F_0(P)$. Similar to the graduated assignment algorithm (Gold and Rangarajan, 1996) and based on (26), the graduated assignment based subgraph matching algorithm is proposed as listed in Algorithm 2. It is noted that, to make the double normalization process applicable on the non-square matrix to guarantee $P \in \Pi$, in the algorithm we multiply a factor N_M / N_S on the normalization results.

4.2. on artificial data

Four types of artificial graph data are generated for the experimental comparison: undirected/directed uniform graph and undirected/directed scale-free graph. The uniform graph is generated by setting its degree distribution as a uniform distribution. Specifically, a uniform graph model is generated by the following steps: Given a sparsity $s \in [0, 1]$, for each entry A_{ij} of A generate a random number r which is uniformly distributed within $[0, 1]$; if $r > s$, associate A_{ij} a random weight, or otherwise $A_{ij} = 0$. The degree distribution of a scale-free graph follows a power law $p(k) \propto k^{-\alpha}$ (in the following experiments α is fixed

at 1.5). The edge weights of the generated graphs follow an absolute normal distribution ($p(w) = \frac{2}{\sqrt{2\pi}}e^{-\frac{w^2}{2}}, w \geq 0$).

The first experiment compared the seven algorithms by randomly generating 100 graph pairs on each graph type, with fixed $N_S = 16$ and $N_M = 8$. For uniform graphs, the sparsity $s = 0.5$. The experiment results are listed in Table 1, from which we can notice the following three points:

- The first three algorithms which realize the subgraph matching by adding some dummy nodes in the smaller graph got much worse performance than the other four algorithms based directly on (1), echoed by the previous discussions.
- All of the three CCRP algorithms outperforms GA_S. This witnesses the superiority of the CCRP framework.
- The proposed CCRP1 outperforms both CCRP2 and CCRP3.

Table 1: Comparative experimental results on 100 graph pairs with $N_S = 16$ and $N_M = 6$

| | error | GA | PATH | EPATH | GA_S | CCRP1 | CCRP2 | CCRP3 |
|----|-------|-------|-------|-------|-------|-------|-------|-------|
| uu | ave | 24.34 | 22.82 | - | 18.55 | 7.272 | 8.775 | 8.773 |
| | std | 11.82 | 7.573 | - | 7.378 | 3.992 | 2.862 | 2.521 |
| du | ave | 38.67 | - | 28.47 | 23.52 | 11.20 | 17.67 | 11.92 |
| | std | 10.32 | - | 7.858 | 6.416 | 2.705 | 3.914 | 3.733 |
| us | ave | 22.75 | 14.98 | - | 13.11 | 6.229 | 6.305 | 6.976 |
| | std | 6.658 | 9.704 | - | 7.366 | 4.937 | 4.464 | 4.639 |
| ds | ave | 26.87 | - | 19.97 | 16.12 | 6.912 | 9.220 | 7.809 |
| | std | 10.22 | - | 8.617 | 5.853 | 2.009 | 3.995 | 2.702 |

ave: average matching error; std: standard deviation; uu: undirected uniform graphs; du: directed uniform graphs; us: undirected scale-free graphs; ds: directed scale-free graphs

The second experiment was to evaluate the algorithms with respect to the scale of the graph. First, the size of G_S is fixed at 40, and the size of G_M is increased from 4 to 40 by a step size 4, and then the size of G_M is fixed at 8, and the size of G_S is increased from 8 to 40 by a step size 4. In the experiment, the sparsity s of the uniform graphs is set as 0.5.

The experiment results are plotted in Figs. 1 and 2, respectively, where the error bar denotes the standard deviation. From the experimental results we can observed that, first, CCRP1 got the best matching results in almost all cases; second, on equal-sized graph matching problems, i.e., $G_M = 40$ in Fig. 1 and $G_S = 8$ in Fig. 2, the PATH or EPATH got comparable results as the CCRP1. However, their performance declines quickly as the problem becomes a subgraph matching problem.

The third experiment was to evaluate the noise resistant ability of the algorithms. For each graph pair, the second graph was generated based on the first one by adding some noises which are controlled by a noise level. Specifically, for uniform graphs, given a noise level $\beta \in [0, 1]$ and a randomly generated uniform graph G_S (with $N_S = 16$), the smaller

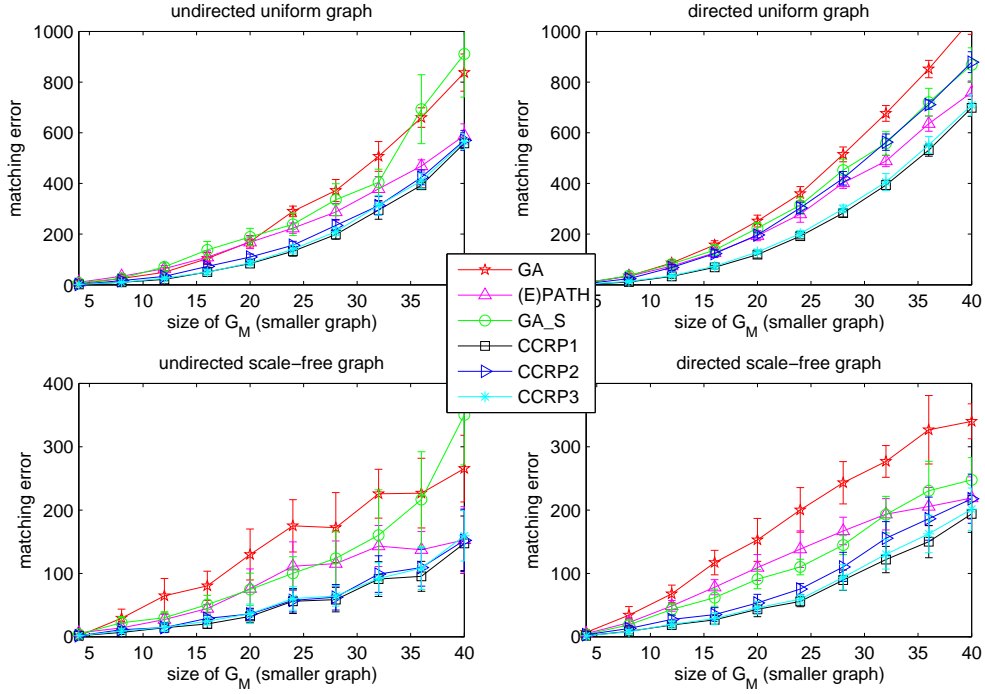


Figure 1: Comparative experimental results on the four types of graphs summarized from 50 random runs. Size of the bigger graph G_S is fixed at 40, and the size of the smaller one G_M is increased from 4 to 40 by a step size 4.

graph G_M with $N_M = 8$ is firstly gotten by randomly extracting out from G_S eight nodes as well as the edges belonging to them. Then, the noises are added by the following steps:

1. For each $(A_M)_{ij}, i \neq j$, generate two uniformly distributed random numbers r_1 and $r_2 \in [0, 1]$.
2. If $(A_M)_{ij} > 0$: if $r_1 < \beta$, $(A_M)_{ij} \leftarrow 0$; or otherwise, $(A_M)_{ij} \leftarrow (A_M)_{ij} + \beta r_2$.
3. If $(A_M)_{ij} = 0$: if $r_1 < \beta$, $(A_M)_{ij} \leftarrow r_2$.
4. Generate a random permutation matrix P , and set $A_M \leftarrow P A_M P^T$.

For the scale free graphs, the noise is added as follows. Generate $G_{\hat{M}}$ ($N_{\hat{M}} = 8$) from a scale-free G_S ($N_S = 16$) by randomly extracting out from G_S eight nodes as well as the edges belonging to them. Then, G_M is generated by randomly adding βN_E edges into $G_{\hat{M}}$, where we denote by N_E the number of edges of $G_{\hat{M}}$.

The experimental results on the four types of graphs are shown in Fig. 3, which witnesses once again that the proposed CCRP1 is the best one in almost all of the experiments.

The last experiment was to evaluate the algorithm with respect to the graph sparsities (densities). The uniform graphs were used in the experiment, and the sizes of G_S and G_M

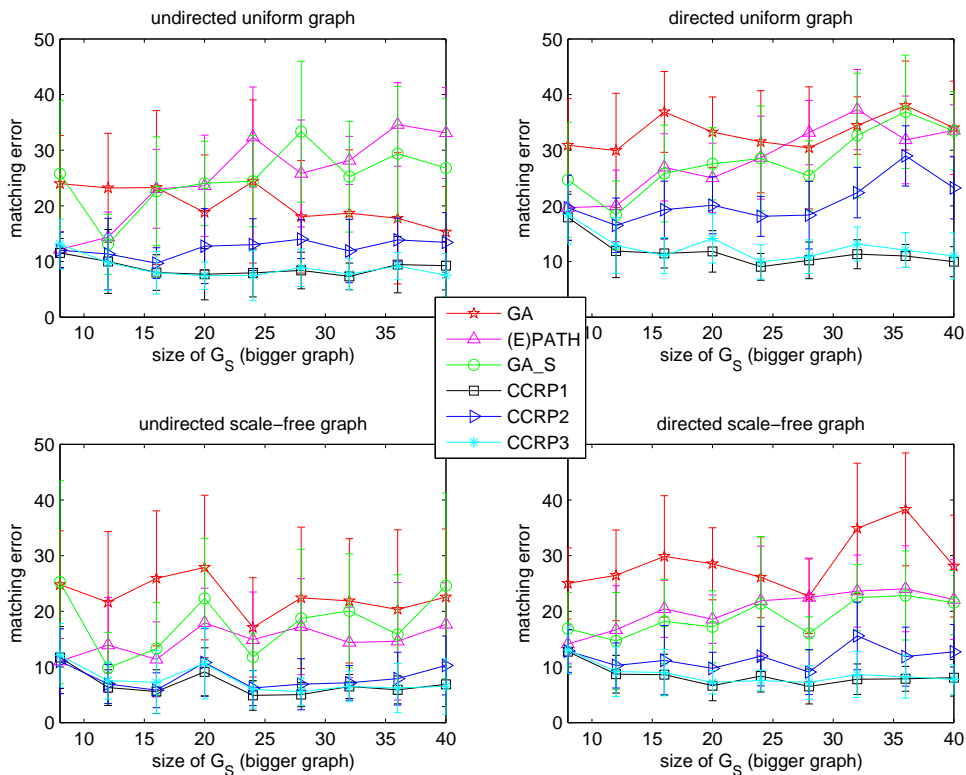


Figure 2: Comparative experimental results on the four types of graphs summarized from 50 random runs. Size of the smaller graph G_M is fixed at 8, and the size of the bigger one G_S is increased from 8 to 40 by a step size 4.

are fixed at 20 and 10 respectively. The sparsity s for both G_S and G_M is increased from 0.1 to 1 by a step size 0.1, and on each level of sparsity 50 graph pairs were randomly generated. The experimental results are shown in Fig. 4, where it is witnessed that CCRP1 outperforms except only for undirected uniform graphs with $s = 1$, on which GA showed a somehow surprisingly good performance.

4.3. on object matching

4.3.1. OBJECTIVE FUNCTION

Feature correspondence (or point pattern matching) is a fundamental problem in computer vision. The objective function of feature correspondence usually involves two terms, the unary term related to the appearance cues and the pairwise term related to the geometric relationship (Torresani et al., 2008; Philbin et al., 2011). Here we consider the problem of matching a smaller structured model to a larger scene structure, i.e., a subgraph matching problem. To formulate the problem under subgraph matching framework where the unary

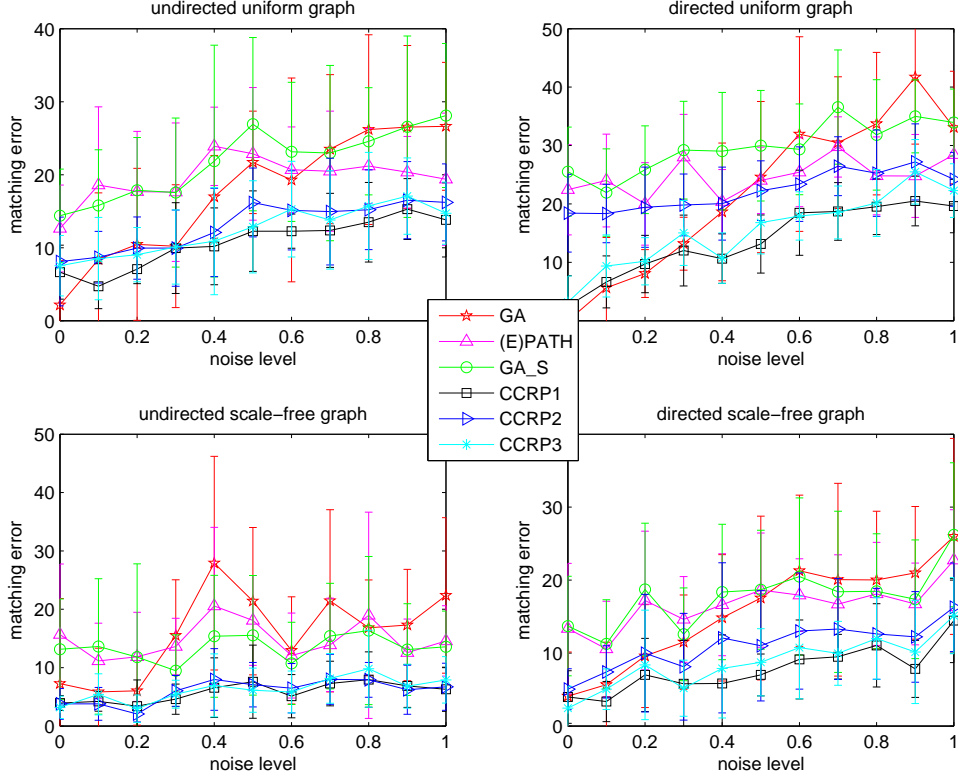


Figure 3: Comparative experimental results on the four types of graphs summarized from 50 random runs. The noise level is increased from 0 to 1 by a step size 0.1.

and pairwise terms actually respectively correspond the vertices and structure dissimilarity in the context of graph matching, we first introduce the distance between the shape context descriptor (Belongie et al., 2002) as the unary term as follows,

$$F_u(P) = \text{tr}(C^T P), C_{ij} = \frac{1}{2} \sum_{k=1}^k \frac{[h_{D_i}(k) - h_{M_j}(k)]^2}{h_{D_i}(k) + h_{M_j}(k)}, \quad (27)$$

where h_{D_i} and h_{M_j} denote the histogram of the shape context of the i th feature of image D and j th feature of image M , respectively. Then, the pairwise term is formulated by the normalized square of the difference between the distances between two feature locations as follows,

$$F_p(P) = \|A_D - P A_M P^T\|_F^2, A_{ij} = \begin{cases} 0 & \text{if } i = j, \\ \|l_i - l_j\|_F / \max_{mn} A_{mn} & \text{otherwise,} \end{cases} \quad (28)$$

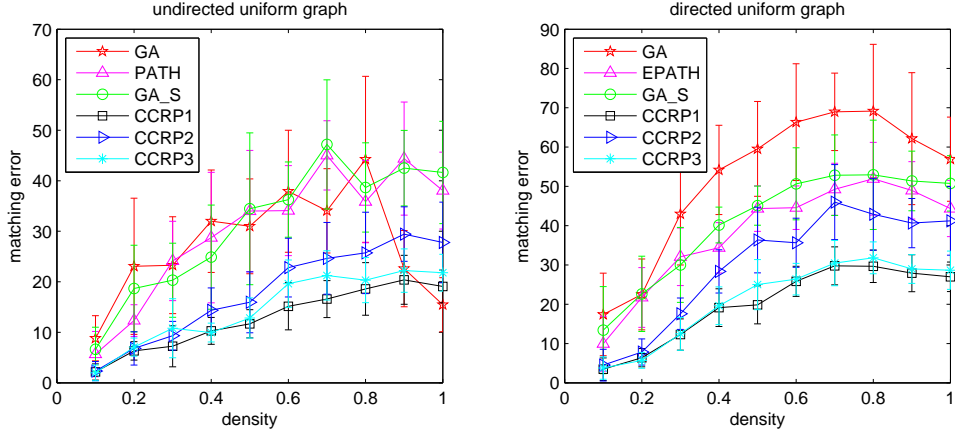


Figure 4: Comparative experimental results on the four types of graphs summarized from 50 random runs. The density level is increased from 0.1 to 1 by a step size 0.1.

where $l_i = (x_i, y_i)^T$ denotes the location of the feature i . The objective function for the feature correspondence is finally given by

$$F(P) = \alpha F_u(P) + (1 - \alpha) F_0(P), P \in \Omega, \quad (29)$$

where α is a constant, which we set 0.1 in our experiments, implying that the matching is dominated by the geometric cues. Due to the linearity of $F_u(P)$ which does not affect the convexity/concavity of $F(P)$, the previously proposed CCRP and GA_S algorithms can be directly used, by adding a linear term as,

$$F(P) = \alpha F_u(P) + (1 - \alpha) F_\gamma^*(P), P \in \Omega, \quad (30)$$

in which $F_\gamma^*(P)$ denotes the objective function of CCRP1, CCRP2, CCRP3 or GA_S. Thus, we need just modify the proposed algorithm by adding a constant term αC into the gradient given by (17), (22), (24) or (12).

4.3.2. RESULTS

In the experiment we adopt the CMU hotel sequence data for the illustration, from which we choose 6 frames, i.e., the frames 1, 21, 41, 61, 81 and 101 with a 20-frame interval. For each frame we manually marked the same 26 feature points (typically the corner points) as shown in Fig. 4.3.2 (the left-hand side one) and extracted their shape context descriptors. A sub-structure with 14 nodes is selected as the sub-structure as shown in Fig. 4.3.2 (the right-hand side one). There are totally 6 sub-structures (one for each frame), each of which was matched to reminder 5 frames, and thus there are totally 60 matchings, which involves totally 420 node pairs ($0.5 \times 60 \times 14$). We compare the GA_S, CCRP1, CCRP2, and CCRP3, and use the average matching error as well as the number of mismatched node pairs as the criterion, as summarized in Tab. 2, from which it is witnessed that the proposed CCRP1 got the best matching results. It is also noticed that CCRP3 whose initialization is provided by a non-convex function also got a promising performance.



Figure 5: The scene structure (left-hand) and sub-structure model (right-hand).

Table 2: Comparative experimental results on feature correspondence

| | GA_S | CCRP1 | CCRP2 | CCRP3 |
|---------------------------------|--------|--------|--------|--------|
| average matching error | 0.5850 | 0.0984 | 0.6106 | 0.1875 |
| number of mismatched node pairs | 38 | 12 | 46 | 16 |

5. Conclusions

In this paper we proposed a convex-concave relaxation procedure (CCRP) based subgraph matching algorithm, which can be used on any types of graph provided that it can be represented by an adjacency matrix. An extensive experimental comparison witness the validity of the proposed algorithm.

References

- A. M. Abdulkader. *Parallel Algorithms for Labelled Graph Matching*. PhD thesis, Colorado School of Mines, 1998.
- S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–521, 2002.
- A. Blake and A Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, USA, 1987.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764, 1995.
- A. Dembo. Bounds of the extreme eigenvalues of postive-definite toeplitz matrices. *IEEE Transactions on Information Theory*, 34(2):352–355, 1988.
- M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67–92, 1973.

- S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.
- G. H. Golub and C. F. Van Loan. *Matrix Computations (third edition)*. The Johns Hopkins University Press, Baltimore and London, 1996.
- R. M. Haralick and G. L. Elliot. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14:263–313, 1980.
- H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
- Z. Y. Liu, H. Qiao, and L. Xu. An extended path following algorithm for graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1451–1456, 2012.
- B. Luo and R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1120–1136, 2001.
- J. Maciel and J. P. Costeira. A global solution to sparse correspondence problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.
- G. Philbin, J. Sivic, and A. Zisserman. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *International Journal of Computer Vision*, 95(2):138–153, 2011.
- Christian Schellewald and Christoph Schnörr. Probabilistic subgraph matching based on convex relaxation. In *In Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 171–186, 2005.
- L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part II, LNCS 5303*, pages 596–609, 2008.
- J. R. Ullman. An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23(1):31–42, 1976.
- S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.
- M. Zaslavskiy, F. Bach, and J. P. Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, 2009.
- X. Z. Zhan. Extremal eigenvalues of real symmetric matrices with entries in an interval. *SIAM Journal on Matrix Analysis and Applications*, 27(3):851–860, 2006.