9-2014

Graph Matching by Simplified Convex-Concave Relaxation Procedure

Zhiyong LIU
Chinese Academy of Sciences

Hong QIAO
Chinese Academy of Sciences

Xu YANG
Chinese Academy of Sciences

Steven C. H. HOI
Singapore Management University, chhoi@smu.edu.sg

Int J Comput Vis (2014) 109:169–186
DOI 10.1007/s11263-014-0707-7

Graph Matching by Simplified Convex-Concave Relaxation Procedure

Zhi-Yong Liu · Hong Qiao · Xu Yang · Steven C. H. Hoi

Received: 30 July 2013 / Accepted: 22 February 2014 / Published online: 22 March 2014

Abstract The convex and concave relaxation procedure (CCRP) was recently proposed and exhibited state-of-the-art performance on the graph matching problem. However, CCRP involves explicitly both convex and concave relaxations which typically are difficult to find, and thus greatly limit its practical applications. In this paper we propose a simplified CCRP scheme, which can be proved to realize exactly CCRP, but with a much simpler formulation without needing the concave relaxation in an explicit way, thus significantly simplifying the process of developing CCRP algorithms. The simplified CCRP can be generally applied to any optimizations over the partial permutation matrix, as long as the convex relaxation can be found. Based on two convex relaxations, we obtain two graph matching algorithms defined on adjacency matrix and affinity matrix, respectively. Extensive experimental results witness the simplicity as well as state-of-the-art performance of the two simplified CCRP graph matching algorithms.

Communicated by M. Hebert.

Z.-Y. Liu (✉) · H. Qiao · X. Yang
The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
e-mail: zhiyong.liu@ia.ac.cn

H. Qiao
e-mail: hong.qiao@ia.ac.cn

X. Yang
e-mail: xu.yang@ia.ac.cn

S. C. H. Hoi
School of Computer Engineering, Nanyang Technological University, Singapore, Singapore
e-mail: chhoi@ntu.edu.sg

Keywords Graph matching · Combinatorial optimization · Deterministic annealing · Graduated optimization · Feature correspondence

1 Introduction

Graph matching involves identifying correspondences between the vertices of two graphs in some optimal way. As a fundamental problem in theoretical computer science, graph matching is closely related to many problems in computer vision and pattern recognition, including for instance feature correspondence Maciel and Costeira (2003); Torresani et al. (2008), object recognition Demirci et al. (2006); Duchenne et al. (2011), and even MAP estimation of Markov random field where the constraint is a little bit different from graph matching Ravikumar and Lakerty (2006); Leordeanu et al. (2012). In this paper, we investigate the graph matching problem under one-to-one pairwise constraint. The problem is, however, in nature a NP-hard combinatorial optimization problem with a factorial complexity. Consequently, many approximate algorithms have been proposed in literature to make it computationally feasible.

Among the different types of approximations Conte et al. (2004), the continuous (relaxation) technique has been receiving extensive attentions in the past three decades, e.g., Fischler and Elschlager (1973); Gold and Rangarajan (1996); Leordeanu and Hebert (2005); Cour et al. (2007); Leordeanu et al. (2009); Zaslavskiy et al. (2009); Cho et al. (2010); Liu et al. (2012); Zhou and De la Torre (2012); Egozi et al. (2013); Cho et al. (2013); Zhou and De la Torre (2013). It involves relaxing the discrete combinatorial optimization problem to be a continuous one. The key point lies in the fact that a continuous optimization problem is usually more flexible to be approximated than its discrete counterpart. In particular,

the objective functions for graph matching can be roughly categorized into two groups, with the first one defined on the adjacency matrices Umeyama (1988); Zaslavskiy et al. (2009); Liu et al. (2012), and the other one on an affinity matrix which measures pairwise/unary similarity between graphs Gold and Rangarajan (1996); Leordeanu and Hebert (2005); Cour et al. (2007); Leordeanu et al. (2009); Suh et al. (2012); Cho and Lee (2012); Tian et al. (2012). Usually the first one can be taken as a special case of the latter one, which in general enjoys a more flexibility to define the node or pairwise similarity. Though many relaxation techniques have been proposed on both categories, such as linear, quadratic and spectral relaxations Leordeanu and Hebert (2005); Zaslavskiy et al. (2009), a common problem encountered by the relaxation techniques is the back projection Gold and Rangarajan (1996); Leordeanu et al. (2009); Zaslavskiy et al. (2009), which needs to project the relaxed continuous solution back to the original discrete one, and is essential for the relaxation techniques.

An intuitive but frequently used technique is the maximal linear assignment (MLA) criterion, which can be efficiently solved by the Hungarian algorithm but may introduce a significant additional error in the final result Zaslavskiy et al. (2009). Another commonly used schema is the graduated assignment Gold and Rangarajan (1996), which introduced a soft-version of iterative conditional modes (ICM) by giving each matching a soft probability by a softmax controlled by a parameter. As the parameter is increased to be large enough, the graduated assignment becomes exactly the maximal linear assignment. Recently, by combining both convex and concave relaxations, the convex-concave relaxation procedure (CCRP) was proposed to tackle the back-projection problem Zaslavskiy et al. (2009); Liu et al. (2012); Liu and Qiao (2012); Zhou and De la Torre (2012). The objective function of CCRP transfers gradually from a convex relaxation to a concave relaxation, whose minima locate exactly in the original discrete domain. Or equivalently, starting from the convex relaxation, CCRP can be regarded as a gradual nonconvexity Blake and Zisserman (1987) like algorithm to minimize the concave relaxation. It was shown that such a deterministic annealing technique outperforms significantly the MLA and some other gradual projection strategies, such as the graduated assignment Gold and Rangarajan (1996), and achieved the state-of-the-art performance on matching accuracy.

However, the convex or concave relaxation can be difficult to construct. Typical examples include the three types of concave relaxations proposed in Zaslavskiy et al. (2009); Liu et al. (2012); Zhou and De la Torre (2012). Here we consider a more ambitious problem: is it possible to find a general concave relaxation that is applicable for any types of problems/graphs? In this paper we will show that, along with a convex relaxation, a very simple quadratic concave function can be equivalently used as a concave relaxation in CCRP, which thus leads to a simplified CCRP. Compared with the conventional CCRP, the simplified CCRP is much simpler to construct. Moreover, although this paper is concentrated on graph matching tasks, the proposed simplified CCRP idea can be generally applied to solve any similar optimizations over partial permutation matrix for which the convex relaxation can be found.

Specifically, the key contributions of this paper are twofold. First, given a convex relaxation, the simplified CCRP offers a much simpler and problem-free way to construct a CCRP algorithm. By contrast, the concave relaxation in CCRP is often quite difficult to find Zaslavskiy et al. (2009); Liu et al. (2012); Liu and Qiao (2012); Zhou and De la Torre (2012). Second, based on the simplified CCRP two types of graph matching algorithms are proposed, defined on adjacency matrix and affinity matrix respectively.

The remainder of this paper is organized as follows. Sect. 2 gives some explanations on the problem formulation and CCRP, and Sect. 3 presents the simplified CCRP technique, followed by the resulting graph matching algorithms in Sect. 4. We conduct extensive experiments on both synthetic data and feature correspondences in Sect. 5, give a brief review of related works in literature in Sect. 6, and finally conclude the paper in Sect. 7.

2 Problem Formulation and CCRP

2.1 Problem Formulation

We consider two types of objective function for the graph matching problem, with the first one defined on the adjacency matrices and the second one on affinity matrix. While the first objective function considers only equal-sized graph matching, the second one can be used on partial matching with outliers.

Given two equal-sized graphs G_M (Model graph) and G_D (Data graph) to be matched, the first objective function is defined on the adjacency matrix as follows,

$$\min . F^a(\mathbf{P}) = \| \mathbf{A}_D - \mathbf{P}\mathbf{A}_M\mathbf{P}^\top \|_F^2, \mathbf{P} \in \mathcal{P} \tag{1}$$

where $\mathbf{A}_D$ and $\mathbf{A}_M$ respectively denote the adjacency matrices of G_D and G_M, $\| \cdot \|_F$ the Frobenius matrix norm, and $\mathcal{P}$ the set of $(N \times N)$ permutation matrices.

The second objective function is defined on a large affinity matrix $\mathbf{K}$ as follows,

$$\max . F^k(\mathbf{X}) = \mathrm{vec}(\mathbf{X})^\top \mathbf{K} \mathrm{vec}(\mathbf{X}), \mathbf{X} \in \Pi \tag{2}$$

where $\mathrm{vec}(\mathbf{X})$ creates a column vector by stacking the column vectors of $\mathbf{X}$, $\mathbf{K}_{ia,jb}$ measures some similarity between the pairwise feature f_{ij} (or edge attribute) of the first graph and f_{ab} of the second graph, and Π denotes the set of $(N \times$

$M, N \leq M$) partial permutation matrix, which is defined as

$$\Pi := \{\mathbf{X} | \mathbf{X}_{ij} = \{0, 1\}, \sum_{j=1}^{M} \mathbf{X}_{ij} = 1, \sum_{i=1}^{N} \mathbf{X}_{ij} \leq 1, \forall i, j\},$$

taking $\mathcal{P}$ as a special case when $N = M$. It is interesting to note that, by setting

$$\mathbf{K}_{ia,jb} = -(\mathbf{A}_{Dij} - \mathbf{A}_{Mab})^2, \tag{3}$$

the two objective functions becomes exactly equal to each other, as described by the following *Corollary*.

Corollary 1 *By constructing* $\mathbf{K}$ *following (3), the two optimizations respectively given by (1) and (2) are equal to each other.*

Proof See Appendix.

Thus, based on *Corollary 1*, the objective function (1) is a special case of (2) when the similarity measure takes the form of square error. Though (2) is much more flexible to define the pairwise/unary similarity than (1), (1) has its own advantage on complexity: (1) enjoys a smaller storage complexity ($\mathcal{O}(N^2)$ versus $\mathcal{O}(N^4)$) and also generally a smaller computational complexity ($\mathcal{O}(N^3)$ versus $\mathcal{O}(N^4)$, see Fig. 6 later).

In practice, however, a more commonly used similarity measure takes the following exponential form Gold and Rangarajan (1996); Zhou and De la Torre (2012),

$$\mathbf{K}_{ia,jb} = \begin{cases} 0 \text{ if either } \mathbf{A}_{Dij} \text{ or } \mathbf{A}_{Mab} = 0 \\ e^{-(\mathbf{A}_{Dij} - \mathbf{A}_{Mab})^2} \text{ otherwise} \end{cases} \tag{4}$$

We then consider the convex relaxations of the two objective functions. First, the convex hull of Π is given by the set of $(N \times M)$ doubly sub-stochastic matrices denoted by $\mathcal{D}$ as follows Maciel and Costeira (2003),

$$\mathcal{D} := \{\mathbf{X} | \mathbf{X}_{ij} \geq 0, \sum_{j=1}^{M} \mathbf{X}_{ij} = 1, \sum_{i=1}^{N} \mathbf{X}_{ij} \leq 1, \forall i, j\}.$$

Similarly, $\mathcal{D}$ takes the set of $(N \times N)$ doubly stochastic matrices (the convex hull of $\mathcal{P}$) as a special case.

For the first objective function $F^a(\mathbf{P})$, as $\mathbf{P} \in \mathcal{P}$, it can be derived as

$$\begin{aligned} \min . F_0^a(\mathbf{P}) &= \| \mathbf{A}_D - \mathbf{P}\mathbf{A}_M\mathbf{P}^\top \|_F^2 = \| \mathbf{A}_D\mathbf{P} - \mathbf{P}\mathbf{A}_M \|_F^2 \\ &= \text{vec}(\mathbf{P})^\top \mathbf{Q} \text{vec}(\mathbf{P}), \quad \mathbf{P} \in \mathcal{P}, \end{aligned} \tag{5}$$

where

$$\mathbf{Q} := (\mathbf{I} \otimes \mathbf{A}_D - \mathbf{A}_M^\top \otimes \mathbf{I})^\top (\mathbf{I} \otimes \mathbf{A}_D - \mathbf{A}_M^\top \otimes \mathbf{I}) \tag{6}$$

is a symmetric positive definite matrix. Therefore, $F_0^a(\mathbf{P})$ in (5) is a convex relaxation of (1) when relaxing $\mathbf{P} \in \mathcal{D}$.

In Zhou and De la Torre (2012), as $\mathbf{K}$ takes the form (4) and by factorizing the $\mathbf{K}$, a convex relaxation of (2) is obtained as follows [1],

$$F_0^k(\mathbf{X}) = F^k(\mathbf{X}) - \frac{1}{2}\text{tr}\left(\mathbf{A}_i^1\mathbf{A}_i^1\mathbf{X}\mathbf{X}^\top + \mathbf{A}_i^2\mathbf{A}_i^2\mathbf{X}^\top\mathbf{X}\right) \tag{7}$$

where $\mathbf{A}_i^1 = \mathbf{H}_1\text{diag}(\mathbf{u}_i)\mathbf{H}_1^\top$ and $\mathbf{A}_i^2 = \mathbf{H}_2\text{diag}(\mathbf{v}_i)\mathbf{H}_2^\top$, with $\mathbf{M} = \mathbf{U}\mathbf{V}^\top = \sum_{i=1}^{c} \mathbf{u}_i\mathbf{v}_i^\top$ being a SVD factorization, and $\mathbf{M}$ and $\mathbf{H}$ are obtained by factorizing $\mathbf{K}$ in the following way

$$\mathbf{K} = (\mathbf{H}_2 \otimes \mathbf{H}_1)\text{diag}(\text{vec}(\mathbf{M}))(\mathbf{H}_2 \otimes \mathbf{H}_1)^T. \tag{8}$$

Readers are referred to Zhou and De la Torre (2012) for detailed formulations of $\mathbf{H}$ and $\mathbf{M}$.

2.2 Convex-Concave Relaxation Procedure

The global optimum denoted by $\mathbf{P}_\mathcal{D}$ of the two convex relaxations given above can be found in polynomial time, however, it in general belongs to $\mathcal{D}$ instead of $\mathcal{P}$ or Π. Intuitively, the following maximum linear assignment (MLA) can be used to get a discrete solution,

$$\mathbf{P}_\mathcal{P} = \arg\max_{\mathbf{P} \in \mathcal{D}} \text{tr}\mathbf{P}_\mathcal{D}^\top\mathbf{P}, \tag{9}$$

by inherently assuming that the true discrete solution is close to $\mathbf{P}_\mathcal{D}$. However, such an assumption can be quite wrong in practice, and it may thus introduce a significant additional error in the final result. In Zaslavskiy et al. (2009), in addition to the convex relaxation, a concave relaxation was further introduced to construct the convex-concave relaxation procedure (CCRP) Liu et al. (2012); Liu and Qiao (2012); Zhou and De la Torre (2012) as follows,

$$F_\gamma(\mathbf{P}) = (1 - \gamma)F_0(\mathbf{P}) + \gamma F_1(\mathbf{P}), \quad \mathbf{P} \in \mathcal{D}, \tag{10}$$

where $F_0(\cdot)$ and $F_1(\cdot)$ denote the convex and concave relaxations respectively. In implementation, by gradually increasing γ from 0 to 1, the objective function transfers from a convex relaxation to a concave relaxation, and therefore pushes $\mathbf{P}$ from a doubly (sub-)stochastic matrix gradually to a (partial) permutation matrix. It was shown that such a deterministic annealing outperforms significantly the maximal linear assignment criterion and some other gradual projection strategies, such as the graduated assignment Gold and Rangarajan (1996), and achieved the state-of-the-art performance on matching accuracy.

[1] The convex relaxation (7) is derived by adding some dummy nodes into the smaller graph to obtain an equal-sized matching problem, such that $\mathbf{X}$ is constrained as a permutation instead of a partial permutation matrix. Such an expansion is appropriate in case $\mathbf{K}$ is constructed following (4), because it is straightforward to check that adding dummy nodes will not change the problem. However, if we define the partial matching based on objective function (1) by similarly adding some dummy nodes such that the convex relaxation (5) still holds, it can be shown that it in general changes the objective function.

The convex relaxations corresponding to the two types of objective functions in (1) and (2) are given by (5) and (7), respectively. A convex (concave) relaxation (on graph matching problem) denotes a strictly convex (concave) function $F_0(\mathbf{P})$ ($F_1(\mathbf{P})$) that holds the same minima as the original problem over $\mathcal{P}$ or Π Maciel and Costeira (2003); Zaslavskiy et al. (2009), that is

$$\arg\min_{\mathbf{P}\in\Pi} F_1(\mathbf{P}) = \arg\min_{\mathbf{P}\in\Pi} F_0(\mathbf{P}) = \arg\min_{\mathbf{P}\in\Pi} F(\mathbf{P}). \tag{11}$$

Minimization of a strictly concave function with $\mathbf{P} \in \mathcal{D}$ will result in $\mathbf{P} \in \mathcal{P}$ or Π, which implies that by such a definition a concave relaxation is equivalent to the original problem, since minimization the concave relaxation can directly get a graph matching result, without needing a further step of back projection.

A concave relaxation of $F^a(\mathbf{P})$ in (1) on the undirected graphs without self-loops was figured out as follows Zaslavskiy et al. (2009):

$$F_1^a(\mathbf{P}) = -\mathrm{tr}(\triangle\mathbf{P}) - 2\mathrm{vec}(\mathbf{P})^\top(\mathbf{L}_M^\top \otimes \mathbf{L}_D^\top)\mathrm{vec}(\mathbf{P}), \tag{12}$$

where $\triangle_{ij} := (\mathbf{D}_M(i,i) - \mathbf{D}_D(j,j))^2$, and for a given adjacency matrix $\mathbf{A}$, the diagonal degree matrix $\mathbf{D}_{ii} := \sum_{j=1}^N \mathbf{A}_{ij}$ and Laplacian matrix $\mathbf{L} := \mathbf{D} - \mathbf{A}$. Recently, based on (12), a concave relaxation was proposed for the directed graphs without self-loops as follows Liu et al. (2012),

$$F_2^a(\mathbf{P}) = \frac{1}{|l|}\Big[-\mathrm{vec}(\mathbf{P})^\top(\mathbf{L}_M^\top \otimes \mathbf{L}_D^\top + \mathbf{L}_M \otimes \mathbf{L}_D - l\mathbf{I}) \mathrm{vec}(\mathbf{P}) - \mathrm{tr}(\triangle\mathbf{P})\Big] \tag{13}$$

where l denotes a lower bound of the smallest eigenvalue of the matrix $\mathbf{L}_M^\top \otimes \mathbf{L}_D^\top + \mathbf{L}_M \otimes \mathbf{L}_D$.

On the other hand, a concave relaxation of $F^k(\mathbf{X})$ in (2) was figured out in Zhou and De la Torre (2012) as follows,

$$F_1^k(\mathbf{X}) = \mathrm{tr}\left(\mathbf{K}_q^\top(\mathbf{G}_1^\top\mathbf{X}\mathbf{G}_2 \circ \mathbf{G}_1^\top\mathbf{X}\mathbf{G}_2)\right) - \mathrm{tr}\left((\mathbf{G}_1\mathbf{K}_q\mathbf{G}_2^\top)^\top\mathbf{X}\right) + \mathrm{tr}\left(\mathbf{K}_p^\top\mathbf{X}\right), \tag{14}$$

in which $\mathbf{G} \in \{0,1\}$ indicates a node-edge incidence matrix, and the two affinity matrices $\mathbf{K}_q$ and $\mathbf{K}_p$ measure the similarity of each node and edge pair respectively, for which the readers are referred to Zhou and De la Torre (2012) for a detailed explanation.

Then, together with the convex relaxations given by (5) or (7), CCRP, i.e., the (extended) Path following Zaslavskiy et al. (2009); Liu et al. (2012) and factorized graph matching Zhou and De la Torre (2012) algorithms, can be constructed to approximate the global minimum/maximum of the concave relaxations given above, which remains still a NP-hard problem. However, the above three concave relaxations $F_1^a(\mathbf{P})$, $F_2^a(\mathbf{P})$ and $F_1^k(\mathbf{X})$ were figured out by rather complicated mathematical derivations Zaslavskiy et al. (2009); Liu et al. (2012); Zhou and De la Torre (2012), which can hardly be generalized to other related problems. Below, we prove that, along with a convex relaxation, a very simple concave function can be used equivalently as a general concave relaxation in CCRP and thus propose a simplified CCRP.

3 Simplified CCRP

3.1 Formulations

Below along with a convex relaxation a general concave relaxation $F_3^*(\mathbf{P})$ over $\mathcal{D}$ is firstly constructed. Then, we prove that, in CCRP $F_3^*(\mathbf{P})$ can be equivalently realized by a very simple quadratic concave function ($F_3(\mathbf{P})$), and finally we propose the simplified CCRP.

Given **any** convex relaxation $F_0(\mathbf{P})$ over Π (of size $N \times M$), and by introducing a constant $\eta \in [0,1)$, $F_0(\mathbf{P})$ can be equivalently derived as follows [2],

$$\begin{aligned}\arg\min_{\mathbf{P}\in\Pi} F_0(\mathbf{P}) &= \arg\min_{\mathbf{P}\in\Pi}\left[(1-\eta)F_0(\mathbf{P}) - \eta N\right] \\ &= \arg\min_{\mathbf{P}\in\Pi}\Big[(1-\eta)F_0(\mathbf{P}) - \eta\mathrm{vec}(\mathbf{P})^\top\mathrm{vec}(\mathbf{P})\Big].\end{aligned} \tag{15}$$

We then define

$$\begin{aligned}F_3(\mathbf{P}) &:= -\mathrm{vec}(\mathbf{P})^\top\mathrm{vec}(\mathbf{P}), \\ F_3^*(\mathbf{P}) &:= (1-\eta)F_0(\mathbf{P}) + \eta F_3(\mathbf{P}).\end{aligned} \tag{16}$$

The Hessian matrix $H_3^*(\mathbf{P})$ of $F_3^*(\mathbf{P})$ is then found as

$$H_3^*(\mathbf{P}) = (1-\eta)H_0(\mathbf{P}) - \eta\mathbf{I}$$

where $H_0(\mathbf{P})$ denotes the Hessian matrix of $F_0(\mathbf{P})$. It is then straightforward to show that, to make $F_3^*(\mathbf{P})$ strictly concave, the constant η should satisfy

$$\eta > \frac{\lambda_{max}}{1+\lambda_{max}}, \tag{17}$$

where λ_{max} denotes the maximal eigenvalue of $H_0(\mathbf{P})$. For instance, for $F_0^a(\mathbf{P})$, $H_0(\mathbf{P}) = \mathbf{Q}$ in (6).

Thus, given a convex relaxation $F_0(\mathbf{P})$, we propose a general concave relaxation as follows,

$$\begin{aligned}F_3^*(\mathbf{P}) &= (1-\eta)F_0(\mathbf{P}) + \eta F_3(\mathbf{P}), \\ &\mathbf{P}\in\mathcal{D},\ 1 > \eta > \frac{\lambda_{max}}{1+\lambda_{max}}.\end{aligned} \tag{18}$$

[2] For convenience sake and without loss of generality, we consider *minimization* problem here, since the maximization problem such as (2) can be transferred to be a minimization one by setting $\mathbf{K} \leftarrow -\mathbf{K}$.

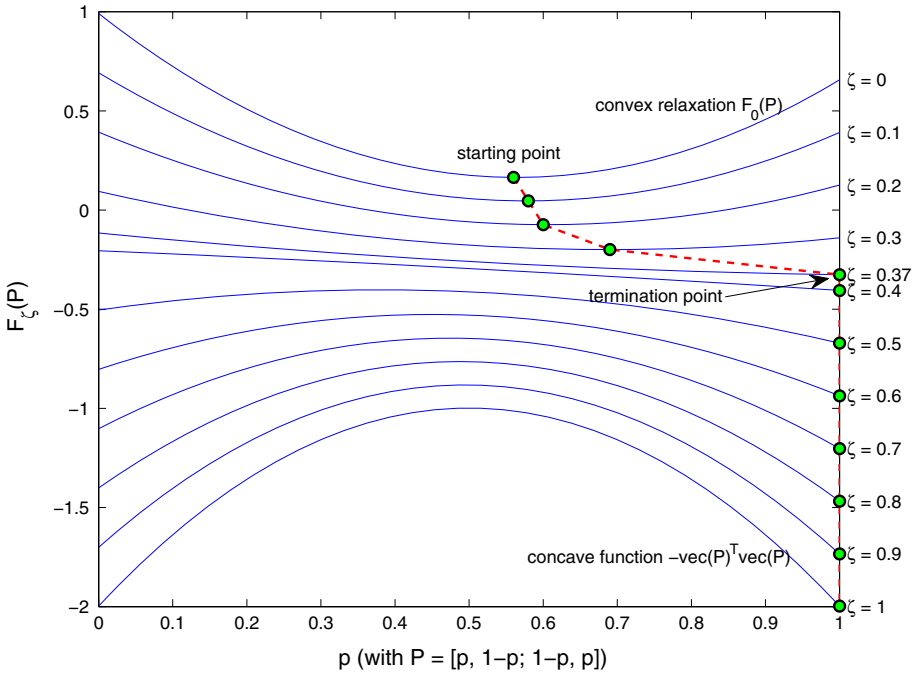

Fig. 1 Illustration of the convergence of the Simplified CCRP, where the *dotted line* denotes the convergence path (Color figure online)

Then, based on the convex relaxation $F_0(\mathbf{P})$ and concave relaxation $F_3^*(\mathbf{P})$, we can construct a CCRP as follows:

$$\begin{aligned} F_\gamma(\mathbf{P}) &= (1-\gamma)F_0(\mathbf{P}) + \gamma F_3^*(\mathbf{P}) \\ &= (1-\gamma\eta)F_0(\mathbf{P}) + \gamma\eta F_3(\mathbf{P}) \\ &= (1-\zeta)F_0(\mathbf{P}) + \zeta F_3(\mathbf{P}), \end{aligned} \tag{19}$$

where we define $\zeta := \gamma\eta$, and its value range is $[0, \eta]$. Below we show that the value range of ζ can be equivalently expanded from $[0, \eta]$ to $[0, 1]$, thus avoiding the trouble of finding λ_{max}, the largest eigenvalue of $H_0(\mathbf{P})$ which is a huge matrix of size $NM \times NM$.

To justify the equivalence between $[0, \eta]$ and $[0, 1]$ for ζ, we need just to show that the resulted $\mathbf{P}$ by minimizing (19) when $\zeta = \eta$ will remain unchanged by gradually increasing ζ from η to 1. We justify this point from two different perspectives. First, to make $F_3^*(\mathbf{P})$ a concave relaxation, η can be chosen to be arbitrarily close to 1. From the viewpoint of implementation in practice, it can be simply set to be 1 without deteriorating the final result. Second, as ζ reaches η, the objective function $F_\gamma(\mathbf{P})$ becomes $F_3^*(\mathbf{P})$, which results in $\mathbf{P} \in \Pi$. According to the simplified CCRP algorithm (as given in Algorithm 4.1 later), $\mathbf{P} \in \Pi$ will terminate the algorithm. Thus, though we expand the range of ζ from $[0, \eta]$ to $[0, 1]$ to avoid finding η explicitly, the algorithm will still stop as ζ reaches η, but in an implicit way[3].

[3] Actually, if ζ is further increased from η to be 1, the resulted $\mathbf{P} \in \Pi$ will retain since it remains to be a local minimum of the concave function $F_\zeta(\mathbf{P})$.

Finally, we have the following CCRP objective function, though $F_3(\mathbf{P})$ itself is not a concave relaxation:

$$F_\zeta(\mathbf{P}) = (1-\zeta)F_0(\mathbf{P}) + \zeta F_3(\mathbf{P}), \quad \mathbf{P} \in \mathcal{D}, \zeta \in [0, 1]. \tag{20}$$

Thus, given a convex relaxation $F_0(\mathbf{P})$, we can easily construct a CCPR as (20), without resorting a concave relaxation in an explicit way. We thus name it the *simplified* CCRP.

3.2 Interpretation and Discussions

It is worth noting that the above derivation is applicable on any convex relaxation over Π, that is, the simplified CCRP can be generally applied to tackle those optimization problems over Π whose convex relaxation is available. It thus greatly simplifies the process of developing a CCRP algorithm to solve similar problems in a new application. By contrast, the concave relaxation in the original CCRP Zaslavskiy et al. (2009); Liu et al. (2012); Zhou and De la Torre (2012) is problem dependent and should be constructed specifically, which is typically a difficult task and consequently makes it difficult to generalize to other related problems. It is needed to point out that the concave relaxation, i.e., $F_3^*(\mathbf{P})$, of simplified CCRP is also inherently problem dependent, but it can be equivalently realized by a problem independent concave term, i.e., $F_3(\mathbf{P})$ in the simplified CCRP. Actually, as indicated by (19), the simplified CCRP will always converge before ζ reaches 1.

Based on the objective function (1), a simple illustration of the convergence process of the simplified CCRP (20) is given in Fig. 1, which shows how $F_\zeta(\mathbf{P})$ changes gradually from the convex relaxation to a concave relax-

ation as ζ increases, and thus pushes $\mathbf{P}$ from a doubly stochastic matrix to a permutation one. It is also worth noting that when $\zeta > 0.37 \approx \frac{\lambda_{max}}{1+\lambda_{max}}$, the objective function $F_\zeta(\mathbf{P})$ becomes concave; it consequently makes the result obtained by minimizing $F_\zeta(\mathbf{P})$ a discrete one ($p = 1$ with $\mathbf{P} := [p, 1-p; 1-p, p]$ in this example), implying that the simplified CCRP is terminated when ζ reaches 0.37.

The simplified CCRP also provides an explanation for the results shown in Liu et al. (2012), where it was found that different estimations (even a quite loose one) of the lower bound l in the concave relaxation $F_2^a(\mathbf{P})$ given by (13) has little effect on the final results. This is because even in the extreme case $l \to -\infty$, $F_2^a(\mathbf{P}) \to F_3(\mathbf{P})$ makes the extended PATH following algorithm become a simplified CCRP.

4 Graph Matching Algorithms Based on Simplified CCRP

Below we first give the algorithmic framework of the simplified CCRP algorithm, and then give two graph matching algorithms targeting at the two objective functions, i.e., (1) and (2).

Algorithm 4.1 presents the details of the proposed simplified CCRP algorithmic framework, where for each fixed ζ, we adopt the Frank-Wolfe algorithm Frank and Wolfe (1956) to minimize the objective function based on the result obtained from the previous step as the starting point. In the algorithm, the linear program, i.e.,

$$\mathbf{Y} = \arg\min_{\mathbf{Y}} \operatorname{tr}(\nabla F_\zeta(\mathbf{P})^\top \mathbf{Y}), \quad \text{s.t. } \mathbf{Y} \in \mathcal{D} \tag{21}$$

can be solved by applying the well-known Hungarian algorithm Kuhn (1955), and α can be found using a backtracking algorithm Boyd and Vandenberghe (2004) or in some cases by a closed form. For instance, the α in the two SCCRP algorithms below can be obtained analytically.

Algorithm 4.1: SIMPLIFIED_CCRP()

$\zeta \leftarrow 0, \mathbf{P} \leftarrow \mathbf{1}_{N\times M}/M$
repeat
  repeat
    $\mathbf{Y} = \arg\min_{\mathbf{X}} \operatorname{tr}(\nabla F_\zeta(\mathbf{P})^\top \mathbf{Y}), s.t.\ \mathbf{Y} \in \mathcal{D}$
    $\alpha = \arg\min_\alpha F_\zeta(\mathbf{P} + \alpha(\mathbf{Y} - \mathbf{P})), s.t.\ 0 \le \alpha \le 1$
    $\mathbf{P} \leftarrow \mathbf{P} + \alpha(\mathbf{Y} - \mathbf{P})$
  until $\mathbf{P}$ converged
  $\zeta \leftarrow \zeta + d\zeta$
until $\zeta \ge 1 \vee \mathbf{P} \in \Pi$
return $(\mathbf{P})$

The gradient $\nabla F_\zeta(\mathbf{P})$ takes the form

$$\nabla F_\zeta(\mathbf{P}) = (1-\zeta)\nabla F_0(\mathbf{P}) - 2\zeta\mathbf{P}, \tag{22}$$

where $\nabla F_0(\mathbf{P})$ denotes the gradient of the convex relaxation.

Specifically, when taking (1) as the objective function and (5) as its convex relaxation, $\nabla F_0^a(\mathbf{P})$ can be computed as follows,

$$\nabla F_0^a(\mathbf{P}) = 2\Big[\mathbf{A}_D^\top\mathbf{A}_D\mathbf{P} - \mathbf{A}_D^\top\mathbf{P}\mathbf{A}_M - \mathbf{A}_D\mathbf{P}\mathbf{A}_M^\top + \mathbf{P}\mathbf{A}_M\mathbf{A}_M^\top\Big] \tag{23}$$

The graph matching algorithm is denoted latter as SCCRP_A.

To use the simplified CCRP Algorithm 4.1 to maximize the objective function (2), we first set $\mathbf{K} \leftarrow -\mathbf{K}$ to transfer it to be a minimization problem. Then, when using (7) as the convex relaxation, the simplified CCRP algorithm can be directly used by finding $\nabla F_0^k(\mathbf{X})$ as follows Zhou and De la Torre (2012),

$$\nabla F_0^k(\mathbf{X}) = 2\mathbf{H}_1(\mathbf{H}_1^\top\mathbf{X}\mathbf{H}_2 \circ \mathbf{M})\mathbf{H}_2^\top - \mathbf{H}_1(\mathbf{H}_1^\top\mathbf{H}_1 \circ \mathbf{U}\mathbf{U}^\top) \times\mathbf{H}_1^\top\mathbf{X} - \mathbf{X}\mathbf{H}_2(\mathbf{H}_2^\top\mathbf{H}_2 \circ \mathbf{V}\mathbf{V}^\top)\mathbf{H}_2^\top. \tag{24}$$

The algorithm is denoted as SCCRP_K.

For the SCCRP, if the matching problem involves also the similarity between nodes, which is measured by a node affinity/similary matrix $\mathbf{C} \in \mathbf{R}^{N\times M}$, it can be directly incorporated into the objective function (1) or (2) by just adding one linear term $\operatorname{tr}(\mathbf{C}^\top\mathbf{X})$. Consequently, the SCCRP algorithms including SCCRP_A and SCCRP_K need only to be modified by adding the constant $\mathbf{C}$ into $\nabla F_0(\mathbf{P})$.

Without considering sparsity (of the adjacency matrices), the computational complexity of the SCCRP_A is roughly $\mathcal{O}(N^3)$, and the space complexity is $\mathcal{O}(N^2)$. It is possible to further improve the time and space complexity by exploiting the properties (e.g., sparsity) of the adjacency matrices in a real-world application, which is beyond the scope of our focus in this paper. The complexity of GNCCP_K is roughly $\mathcal{O}(|E|^3)$, dominated by the number of edges E, and generally bigger than that of SCCRP_A.

5 Experimental Evaluations

5.1 Overview

We have conducted two sets of experiments to evaluate the efficacy of the proposed simplified CCRP algorithms for graph matching. The first experiment is on synthetic data and the second one is on feature correspondence in computer vision and pattern recognition. We compare the two proposed simplified CCRP algorithms with a number of existing popular and state-of-the-art algorithms for graph matching, including

– the algorithm based on convex relaxation (5) along with a maximal linear assignment (9) (denoted as "QCV" for short);
– the graduated assignment algorithm (denoted as "GA" for short) Gold and Rangarajan (1996);
– the spectral graph matching algorithm Leordeanu and Hebert (2005) (denoted as "SM" for short);
– the reweighted random walks matching algorithm Cho et al. (2010) (denoted as "RRWM" for short);
– the integer projected fixed point matching algorithm Leordeanu et al. (2009) (denoted as "IPFP" for short);
– the probabilistic spectral graph matching algorithm Egozi et al. (2013) (denoted as "PGM" for short);
– the PATH algorithm Zaslavskiy et al. (2009) which can be applied only for undirected graphs;
– the extended PATH (denoted as "EPATH" for short) for directed graph only Liu et al. (2012),
– the factorized graph matching algorithm Zhou and De la Torre (2012) (denoted as "FGM" for short);
– the simplified CCRP algorithm minimizing (1) (denoted as "SCCRP_A");
– the simplified CCRP algorithm maximizing (2) with $\mathbf{K}$ given by (4)(denoted as "SCCRP_K").

Some of the above algorithms were conducted on adjacency matrices of the graphs, including QCV, (E)PATH, and SCCRP_A, while the remainder seven attempt to maximize the objective function (2).

Some algorithms targeting at maximizing (2) requires a symmetric affinity matrix $\mathbf{K}$. Thus, in case of asymmetric affinity matrix such as on directed graph, we rewrite the objective function as follows,

$$F(\mathbf{x}) = \mathbf{x}^\top \mathbf{K}\mathbf{x} = \frac{1}{2}\mathbf{x}^\top(\mathbf{K}+\mathbf{K}^\top)\mathbf{x}, \tag{25}$$

and then simply take the symmetric matrix $\frac{1}{2}(\mathbf{K}+\mathbf{K}^\top)$ as the new affinity matrix. As shown in Fig. 2, such a simple operation can significantly enhance the performance of some algorithms, including SM, RRWM, IPFP and PGM, when used on an asymmetric $\mathbf{K}$. Thus, in all of the following experiments involving an asymmetric $\mathbf{K}$ (on directed graphs), we utilized (25) instead of (2).

The algorithms were evaluated on both objective functions, i.e., (1) and (2). Specifically, QCV, (E)PATH and SCCRP_A were directly implement to minimize (1), while by setting $\mathbf{K}$ following (3) and according to *Corollary* 1, SM, RRWM, IPFP and PGM were implemented equivalently to minimize (1). On the other hand, GA, SM, RRWM, IPFP, PGM, FGM, and SCCRP_K were used to maximize (2) by setting $\mathbf{K}$ following (4).

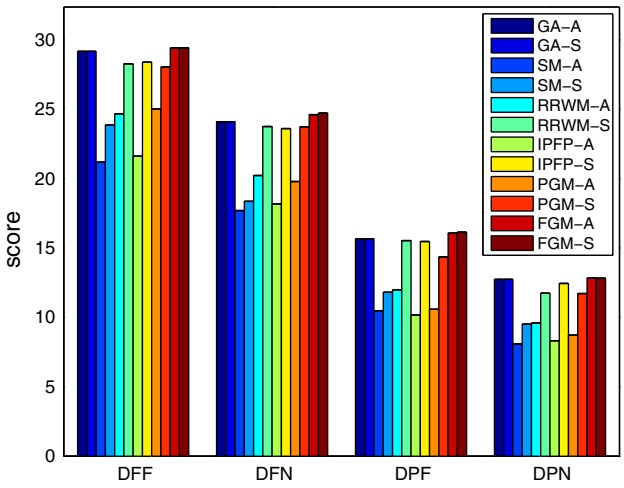

Fig. 2 The comparative matching scores on directed graphs by using asymmetric and symmetric affinity matrix, where "-A" denotes using asymmetric matrix, "-S" denotes using symmetric matrix in (25), and the graph types are as described in Sect. 5.2. It is observed that, except for GA and FGM, "-S" outperforms "-A" significantly (Color figure online)

The four algorithms, SM, RRWM[4], FGM[5], and IPFP[6] were implemented by the public codes, and the remainder seven algorithms were implemented by us[7]. SM, RRWM and FGM were largely implemented by C codes, and the remainder eight algorithms by Matlab. All of the algorithms adopted the the same initial $\mathbf{P}$, i.e., $\mathbf{P}^0 = \mathbf{1}_{N\times M}/M$ ($M = N$ in case of equal-sized matching), and the $d\zeta$ in SCCRP and CCRP was updated following the schema proposed by PATH Zaslavskiy et al. (2009), that is, $d\zeta$ is chosen to make $|F_{\zeta+d\zeta}(\mathbf{P}) - F_\zeta(\mathbf{P})|$ just below a preset constant.

5.2 Experiments on Synthetic Data

The synthetic graphs were generated according to three options:

– undirected (U) or directed (D);
– the degree distribution is uniform (F) or power-law (P);
– the weight distribution is uniform (F) or absolute normal (N).

Thus, there are a total of 8 types of graphs used in the experiments and below we will use a sequential three-character representation of the graph types. For instance, "UFF" denotes the undirected uniform graph with a uniform weight distribution. A uniform graph model is generated as follows: Given

[4] Codes of SM and RRWM are available at http://cv.snu.ac.kr/research/~RRWM/.

[5] http://www.f-zhou.com/gm.html.

[6] http://109.101.234.42/code.php.

[7] All of the codes of the ten algorithms are available at http://www.escience.cn/people/zyliu/SCCRP.html.

a density $s \in [0, 1]$, for each entry $\mathbf{A}_{ij}$ generate a random number r uniformly distributed within $[0, 1]$; if $r < s$, assign $\mathbf{A}_{ij}$ a random weight, or otherwise $\mathbf{A}_{ij} = 0$. The power-law degree distribution of the scale-free graph takes the form $p(k) \propto k^{-\alpha}$ and the interval for the uniform weight distribution is $[0, 1]$.

On equal-sized matching we have conducted four different experiments to evaluate the performance of the ten algorithms in varied settings: (i) evaluation on randomly generated graphs, (ii) evaluation of their noise resistances, (iii) evaluation on the graphs with different density levels, and (iv) evaluation of their scalability with respect to different graph sizes. On partial graph matching, we evaluated the seven algorithms on graph pairs with the smaller graph generated by adding some noises on the bigger one.

In the first experiment, 20 pairs of the 8 types of graphs with size $N = 20$ are randomly generated. The density s for the uniform graphs was set as 0.5 and $\alpha = 1.5$ for the scale-free graphs. The experimental results on the two objective functions are listed in Tables 1 and 2, respectively. It is observed that on both objective functions, the SCCRP algorithms, i.e., SCCRP_A and SCCRP_K, exhibited quite promising performance. Specifically, on both objective functions SCCRP_A and SCCRP_K achieved four best results out of the eight graph types. Meanwhile, on the first objective function RRWM, PGM and (E)PATH respectively achieved one, two, and one best, while on the second one, FGM achieved also four best results. In the tables the criterion "ADB" denotes the average deviation from the best result, reflecting the average performance of the algorithm. Specifically, denoting by $r^{\{i\}}$ the best result, i.e., the minimal for the first objective and maximal for the second, obtained by different algorithms on graph type i, the "ADB" (Average Deviation from Best) of algorithm j is calculated by $\text{ADB}_j = \frac{1}{m}\sum_{i=1}^{m}(r_j^{\{i\}} - r^{\{i\}})/r^{\{i\}}$ (on the first objective as in Table 1) or $\text{ADB}_j = \frac{1}{m}\sum_{i=1}^{m}(r^{\{i\}} - r_j\{i\})/r^{\{i\}}$ (on the second objective as in Table 2) where $r_j^{\{i\}}$ denotes the result of algorithm j on graph type i. We can observe that in terms of ADB, on the first objective function the SCCRP_A achieved the best result, and (E)PATH, PGM, and IPFP exhibited slightly worse but still quite promising performances; on the second objective function, the SCCRP_K exhibited a comparable performance to FGM, and both of them outperformed the five competitors.

The second experiment is to test the noise resistance of the algorithms. For each graph pair, the second graph was generated based on the first one by adding some edges controlled

Table 1 Comparative experimental results of the seven algorithms on minimizing (1) by using random synthetic graph

Graph type	QCV	SM	RRWM	IPFP	PGM	(E)PATH	SCCRP_A
UFF	59.8138	67.2794	**34.4401**	39.2331	36.5020	35.2917	35.0733
UFN	179.1208	232.7750	195.9828	118.5279	112.8901	109.7163	**108.282**
UPF	29.3888	37.8726	26.1485	21.0325	19.5174	**19.1531**	19.2473
UPN	85.3349	102.2174	92.8245	53.7769	**47.6693**	49.4849	51.3808
DFF	63.0127	72.2175	60.4988	47.5529	47.2501	46.0678	**45.9765**
DFN	214.4420	245.4640	229.8141	162.2196	163.1717	155.8804	**152.8135**
DPF	33.1539	33.1508	29.0047	22.0202	**20.8692**	22.5778	21.1685
DPN	96.1069	109.0729	103.2193	68.8688	69.5422	71.0895	**65.5589**
ADB	0.5680	0.8318	0.4883	0.0827	0.0347	0.0330	**0.0144**

The best result is shown in bold face

Table 2 Comparative experimental results of the seven algorithms on maximizing (7) with **K** given by (4) by using random synthetic graph

Graph type	GA	SM	RRWM	IPFP	PGM	FGM	SCCRP_K
UFF	123.5550	92.0693	123.9663	125.9298	123.3539	129.9033	**129.9155**
UFN	112.5168	80.4336	106.4767	113.5049	110.3012	**116.4575**	115.9953
UPF	58.0686	47.1001	63.5702	63.6977	53.7354	**65.3023**	65.2606
UPN	42.9694	31.7521	44.1167	45.0439	37.7354	47.2658	**47.2659**
DFF	112.2337	88.7486	115.7447	112.4110	111.4211	**116.5019**	116.2281
DFN	96.2857	71.4531	97.7637	96.1769	94.7641	99.8361	**99.9680**
DPF	36.8796	25.8628	38.7678	38.3836	32.1910	39.5037	**39.6536**
DPN	31.6412	24.4481	34.2227	33.6833	29.2977	**35.7917**	35.0095
ADB	0.0680	0.2995	0.0399	0.0364	0.1184	**0.0006**	0.0036

The best result is shown in bold face

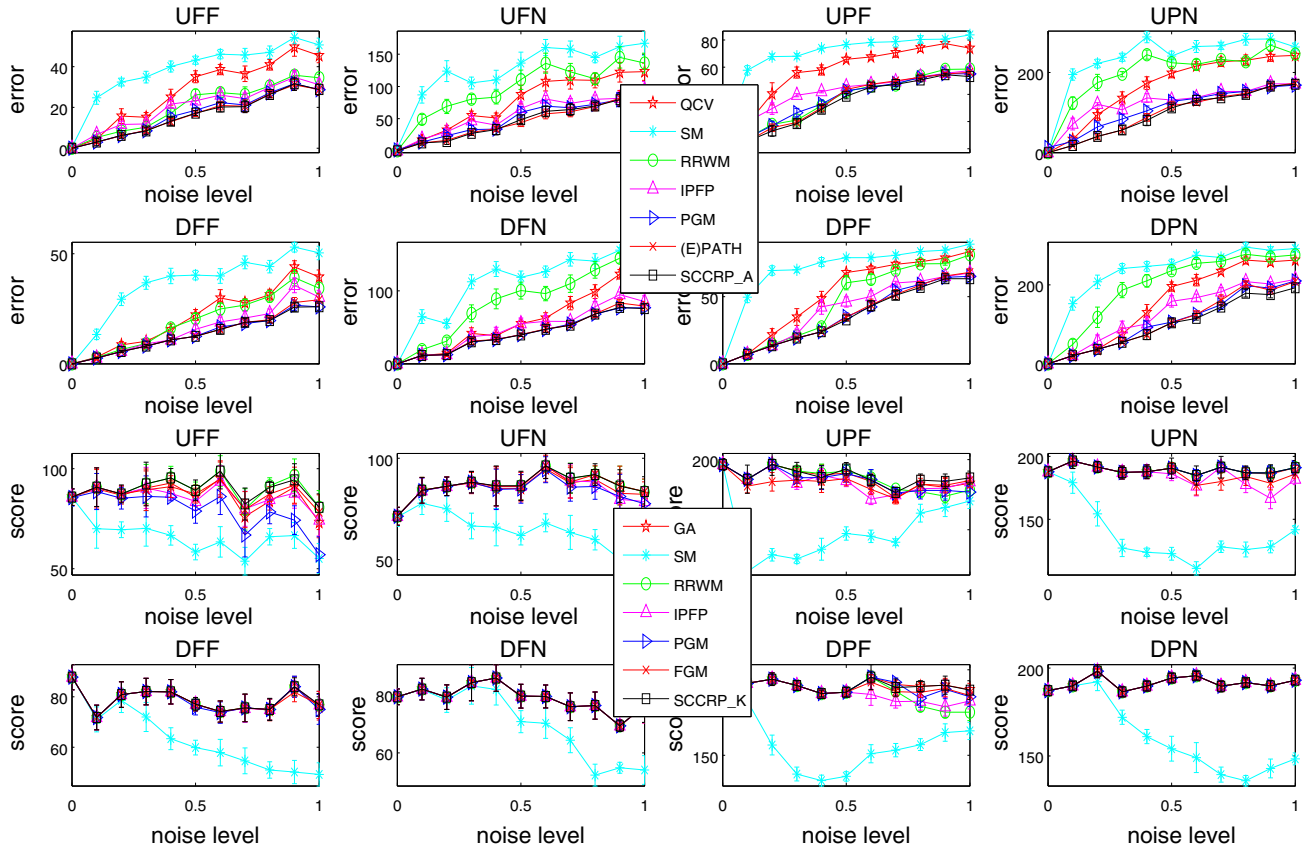

Fig. 3 Matching errors on the eight types of graphs with respect to noise levels. The results were obtained by averaging over 20 random runs on each noise level, and the *error bar* denotes its standard error (Color figure online)

by a noise level β. Specifically, denoting by $|E_D|$ the number of edges in G_D, the second graph G_M is then generated by randomly adding $\beta|E_D|$ edges into G_D. In the experiment, β increases from 0 to 1 by a step size 0.1, and on each noise level, 20 graph pairs with $N = 20$ were generated. The experimental results are shown in Fig. 3, and the number of best result achieved out of the 88 settings by different algorithms are listed in Tables 3 and 4.

The third experiment is to evaluate the algorithms on the graphs with different density levels. For the graphs with uniform degree distributions, the parameter s that control its density was increased from 0.1 to 1 by a step size 0.1, and for the scale-free graph, α was decreased from 5.5 to 1 by a step size 0.5. On each level of density, 20 graph pairs with $N = 20$ were randomly generated. Fig. 4 shows the experimental results and Tables 3 and 4 list the numbers of best result achieved by different algorithms.

The fourth experiment is to evaluate the scalability of the algorithms with respect to graph size. For each of the eight types of graphs, 5 groups of graph pairs were generated, with the sizes increasing from 12 to 60 by a step size 12, and for each group 20 graph pairs are randomly generated. The experimental results are plotted in Fig. 5, and also listed in Tables 3 and 4. It is noted that in this experiment, due to a heavy computational load, the PGM was tested only up to the size $N = 24$.

From the above four experiments on equal-sized graph matching, we can draw two main observations. First, on both objective functions, the performance of the proposed SCCRP, i.e., SCCRP_A and SCCRP_K are at lease comparable to the other state-of-the-art graph matching algorithms, including the CCRP algorithms with more complicated concave relaxations, i.e., PATH, EPATH, and FGM. More specifically, as shown in Tables 3 and 4, on objective function (1), SCCRP_A got 75 best results out of the total 208 experiments, higher than the 70 of (E)PATH, 54 of PGM, and 56 of IPFP, while on objective function (2) with $\mathbf{K}$ given by (4), SCCRP_K got 118 best results, comparable to the 121 of FGM, and higher than those of the PGM, IPFP, GA, and RRWM. Second, the CCRP algorithms including SCCRP, (E)PATH and FGM got in general better results than the other competitors. Third, some interesting points could also be noticed on the four algorithms, SM, RRWM, IPFP, and PGM when using different $\mathbf{K}$. For instance, PGM seemed to be more competitive by setting $\mathbf{K}$ following (3), while IPFP remained a quite stable performance on both objective functions. On the other hand,

Table 3 The numbers of best results achieved by the seven algorithms on minimizing (1)

Experimental settings	QCV	SM	RRWM	IPFP	PGM	(E)PATH	SCCRP_A
on 88 noise levels	12	4	10	14	34	37	39
on 80 density levels	0	0	3	24	15	20	19
on 40 scale levels	0	0	0	18	5	13	17
total number	12	4	13	56	54	70	75

Table 4 The numbers of best results achieved by the seven algorithms on maximizing (2) with **K** given by (4)

Experimental settings	GA	SM	RRWM	IPFP	PGM	FGM	SCCRP_K
on 88 noise levels	56	7	73	51	55	72	80
on 80 density levels	18	0	10	13	6	23	18
on 40 scale levels	4	0	6	4	2	26	20
total number	78	7	89	66	63	121	118

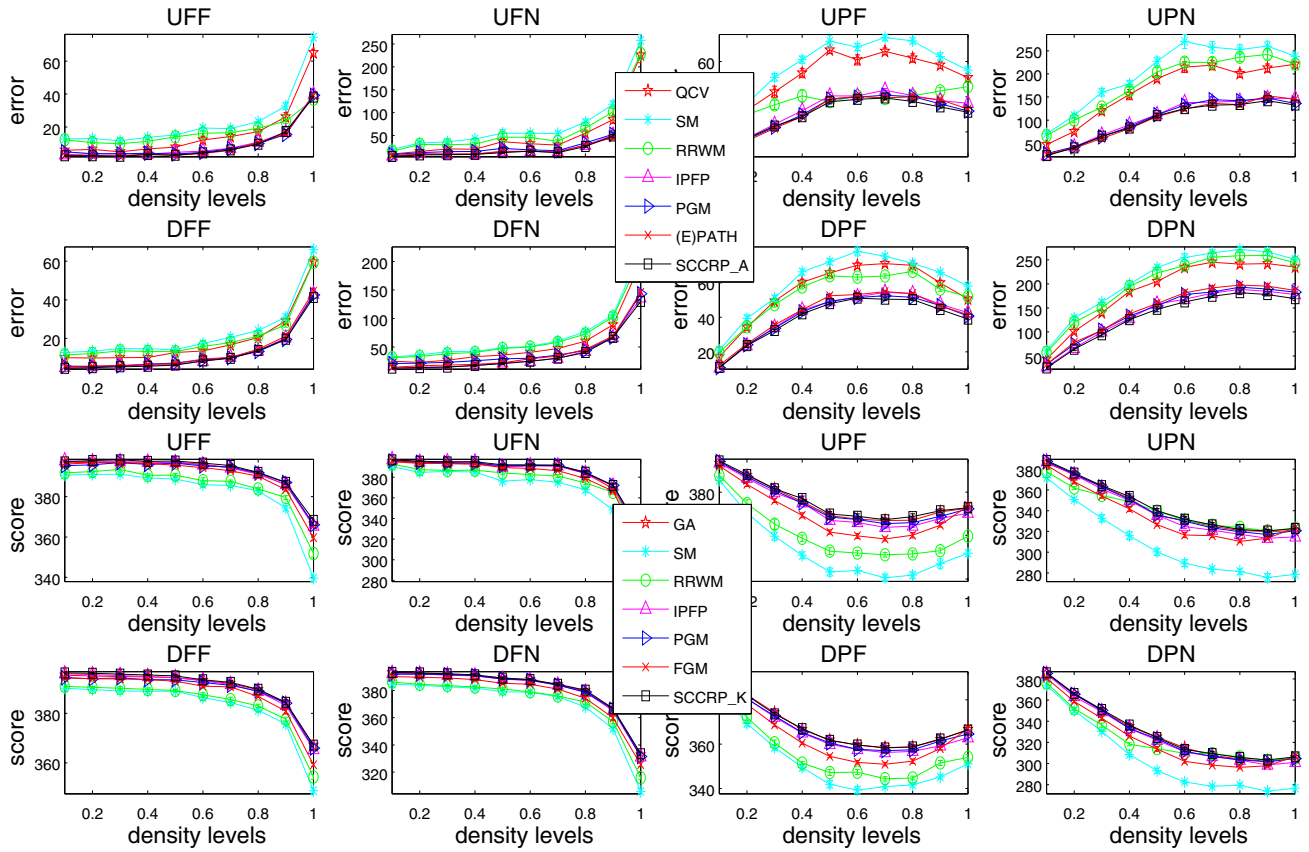

Fig. 4 Matching errors on the eight types of graphs with respect to density. The results were obtained by averaging over 20 random runs on each density level, and the *error bar* denotes its standard error (Color figure online)

though RRWM averagely got much better results by setting **K** as (4) than (3), by (3) it got the best result on UFF.

Figure 6 illustrates the comparisons of average time costs of the ten algorithms with respect to different graph sizes. Empirically, the result reveals that the time complexities of the three algorithms defined on adjacency matrices, i.e., QCV, PATH, and EPATH are around $\mathcal{O}(N^{2.8\pm0.3})$, and those defined on affinity matrix, i.e., GA, SM, RRWM, IPFP, PGM, FGM, and SCCRP_K are around $\mathcal{O}(N^{4.2\pm0.5})$.

The final experiment on synthetic data was conducted to evaluate the seven algorithms defined on the affinity matrix on partial matching with outliers. The size of the smaller graph was fixed at 20, and the bigger one is increased from 25 to 40 by 5. For each graph type, 20 graph pairs were gen-

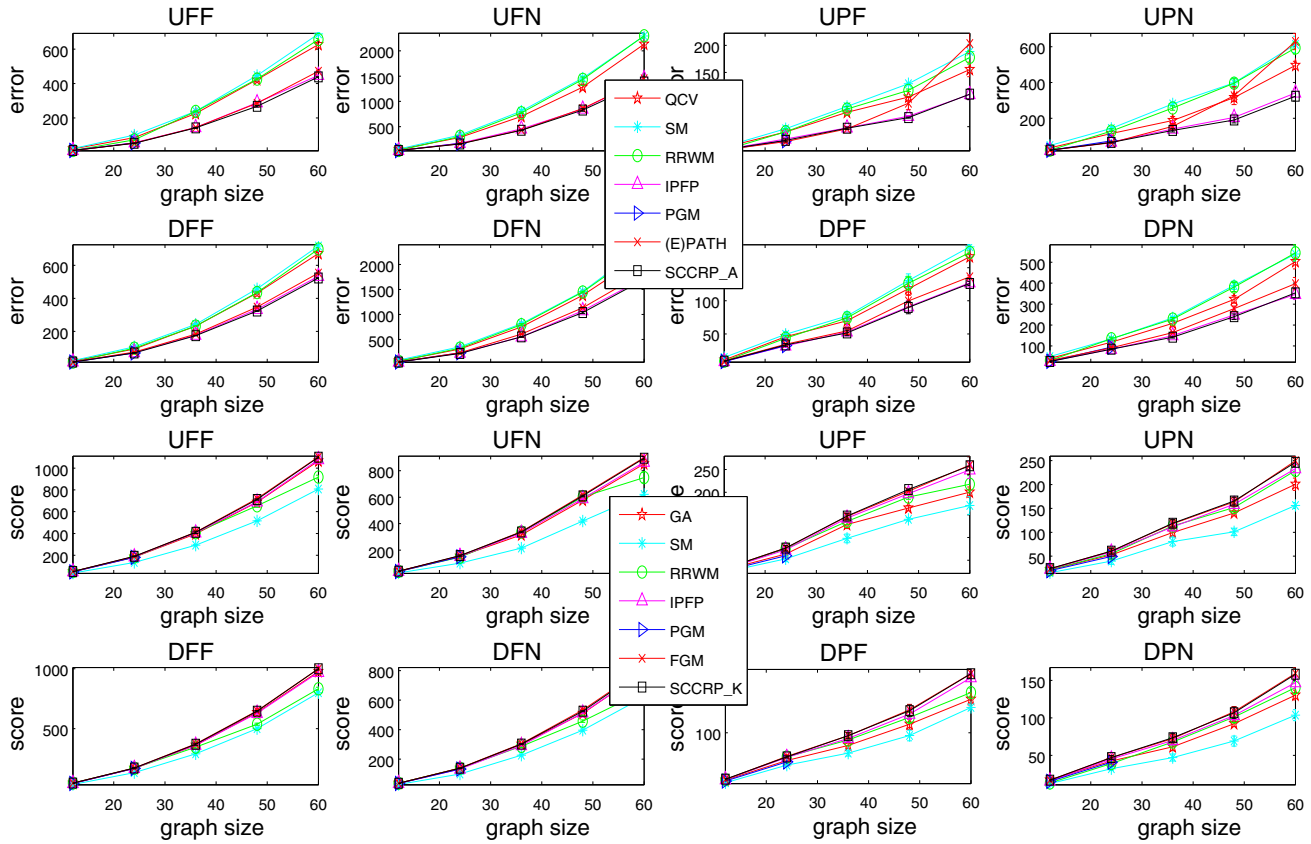

Fig. 5 Matching errors on the eight types of graphs with respect to graph sizes. The results were obtained by averaging over 20 random runs on each size, and the *error bar* denotes its standard error (Color figure online)

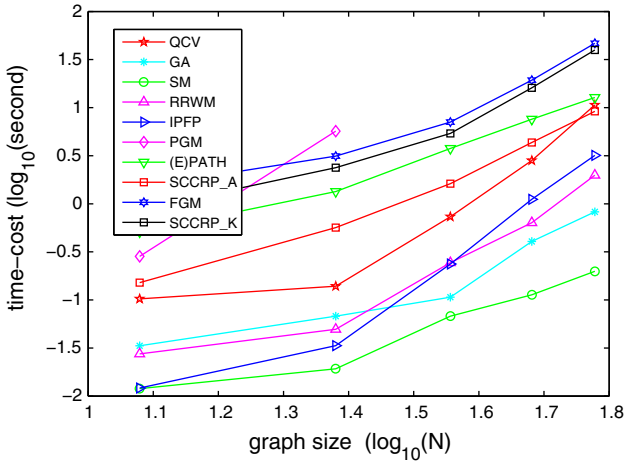

Fig. 6 Time cost of the ten algorithms with respect to graph sizes. The results were obtained by averaging over 20 random runs on each size (Color figure online)

erated in the following way. In each pair, the bigger one was randomly generated, and the smaller one was generated based on the bigger one by randomly extracting out 20 nodes and adding some noises following the second experiment with the noise level $\beta = 0.2$. The experimental results are plotted in Fig. 7, where the GA, SM, RRWM, IFPF, PGM, FGM, and SCCRP_K respectively achieved 12, 0, 14, 13, 11, 19, and 20 best results out of the total 32 settings (on 8 different graph types and 4 levels of number of outliers). Thus, the FGM and SCCRP_K still achieved slightly better results than the competitors, and meanwhile, SCCRP_K exhibited a comparable performance to FGM.

5.3 Experiments on Feature Correspondence

In the following, we first formulate the Feature Correspondence task in computer vision as a Graph Matching Problem, followed by presenting our experimental results on different datasets.

5.3.1 Overview

Feature correspondence (or point pattern matching) is a fundamental problem in computer vision and pattern recognition. Recently, feature correspondence involving pairwise constraint between the feature points has been becoming a popular topic in literature, e.g. Philbin et al. (2011), and consequently the task can be formulated as a graph matching problem Maciel and Costeira (2003); Torresani et al. (2008). Here we consider different types of pairwise constraints such

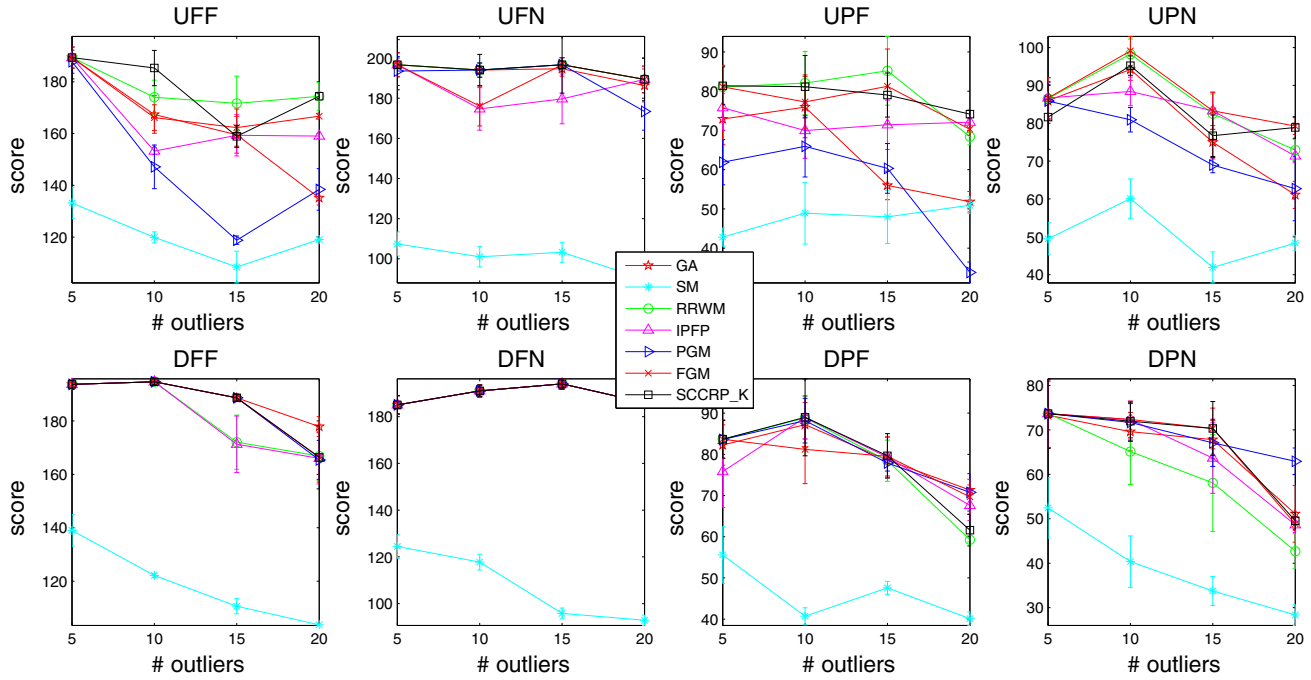

Fig. 7 Matching errors on the eight types of graphs with respect to graph sizes. The results were obtained by averaging over 20 random runs on each size, and the *error bar* denotes its standard error (Color figure online)

as *distance* and *direction* between two feature points. Thus, different from the graph matching discussed previously especially objective (1) where each graph is represented by one adjacency matrix, each feature set in the feature correspondence involves m adjacency matrices, where m denotes the number of constraints. Fortunately, the ten algorithms evaluated above can be straightforwardly generalized to tackle the problem. Meanwhile, because the pairwise features used in the experiment are all symmetric, which implies the problem is basically an undirected graph matching problem, the EPATH is excluded in the experiment.

Specifically, denoting by $f_{ij}^{\{1\}}, \ldots, f_{ij}^{\{m\}}$ the m pairwise features under consideration, the $\mathbf{K}$ in (3) and (4) become respectively

$$\mathbf{K}_{ia,jb} = -\sum_{l=1}^{m} \alpha_l (f_{D_{ij}}^{\{l\}} - f_{M_{ab}}^{\{l\}})^2, \tag{26}$$

$$\mathbf{K}_{ia,jb} = \begin{cases} 0 & \mathbf{A}_{Dij} \text{ or } \mathbf{A}_{Mab} = 0 \\ e^{-\sigma \sum_{l=1}^{m} \alpha_l (f_{D_{ij}}^{\{l\}} - f_{M_{ab}}^{\{l\}})^2} & \text{otherwise} \end{cases} \tag{27}$$

where α_l is a positive constant weight with the constraint $\sum_{l=1}^{m} \alpha_l = 1$, and σ is a scaling factor, which was set as $\sigma = \frac{1}{0.15}$, following some algorithms Cho et al. (2010); Zhou and De la Torre (2012). For those algorithms defined on adjacency matrices including QCV, PATH and SCCRP_A, the convex relaxation of the objective function becomes now

$$F_0(\mathbf{P}) = \sum_{l=1}^{m} \alpha_l \parallel \mathbf{A}_D^{\{l\}} \mathbf{P} - \mathbf{P} \mathbf{A}_M^{\{l\}} \parallel_F^2, \tag{28}$$

where the adjacency matrix $\mathbf{A}_{ij}^{\{l\}} := f_{ij}^{\{l\}}$. For the PATH, the concave relaxation can also be obtained similarly as

$$F_1(\mathbf{P}) = -\sum_{l=1}^{m} \alpha_l \Big[\mathrm{tr}(\triangle^{\{l\}} \mathbf{P}) + 2\mathrm{vec}(\mathbf{P})^\top (\mathbf{L}_M^{\{l\}\top} \otimes \mathbf{L}_D^{\{l\}\top}) \mathrm{vec}(\mathbf{P}) \Big] \tag{29}$$

The proposed SCCRP_A can be directly used on the convex relaxation (28) without needing any changes. Also it is straightforward to show that *Corollary 1* still holds for the two objective functions (2) (by setting $\mathbf{K}$ as (26)) and (28).

Below we evaluated the two proposed SCCRP algorithms based feature correspondence on three data-sets, the CMU house sequences[8], the Chinese Character dataset Liu et al. (2011), and the Pascal object class dataset[9]. In all of the three experiments, on both objective functions the algorithms are compared in two terms: the matching error/score and accuracy rate, i.e., the number of correct matchings.

5.3.2 Results on House Sequence

The CMU house sequence dataset was frequently used for testing the correspondence algorithms, see e.g., Cho et al. (2010); Zhou and De la Torre (2012); Egozi et al. (2013). We labeled manually the same 30 nodes for all of the 111 frames, and linked every pair of them to construct a complete graph,

[8] http://vasc.ri.cmu.edu/idb/html/motion/.

[9] http://109.101.234.42/code.php.

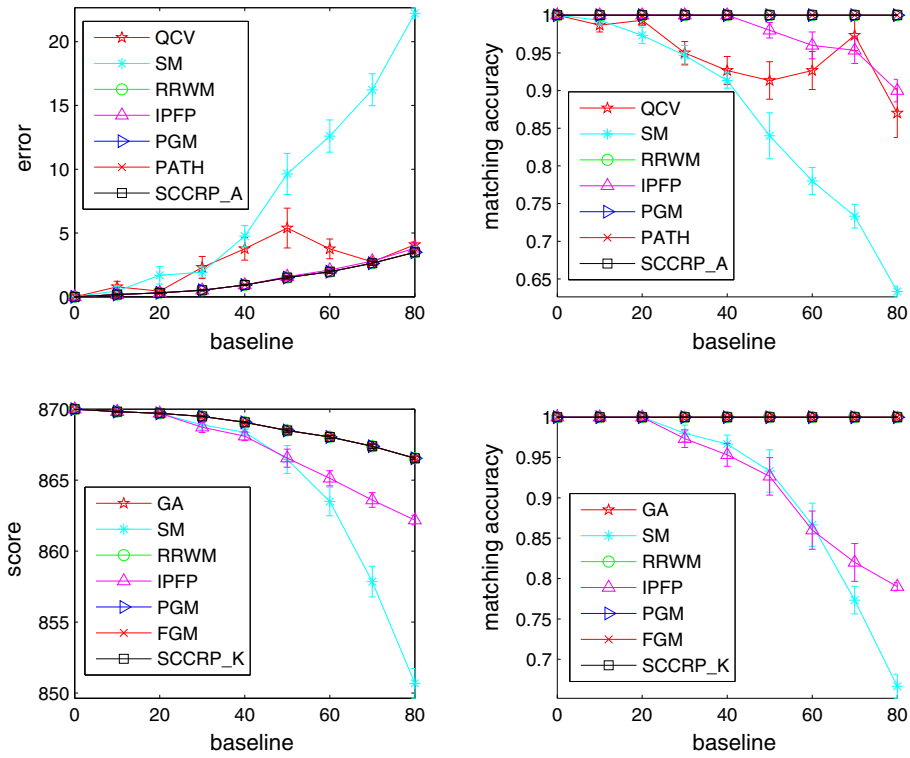

Fig. 8 The matching error of the ten algorithms on House sequence. The top two sub-figures show the results on minimizing (1), and the bottom two on maximizing (2) by setting **K** as (4) (Color figure online)

which suffers mainly a rotation distortion and also a slight perspective distortion. Thus, the correspondences between them is a graph matching problem which is close to graph isomorphism. Similar to the setting in Zhou and De la Torre (2012) we tested the algorithms on all of the possible frame pairs with the separation between them increasing from 0 to 80 by a step size 10. Two types of pairwise constraints with the same weights, i.e., $\alpha_1 = \alpha_2 = 0.5$, were used in the experiment, *distance* (normalized by the maximal one for each object) as well as the *angle* between the edge and the main axis of the structure, where the main axis is calculated by $\overline{d} = \sum_{\substack{i=1\cdots n \\ l_i \neq \bar{l}}} \frac{l_i - \bar{l}}{\|l_i - \bar{l}\|}$, where $\bar{l} = \frac{1}{n}\sum_{i=1\cdots n} l_i$ is taken as the center of the object, and $\|l_i - \bar{l}\|$ is the distance between point i and the center.

The experimental results are shown in Fig. 8, and one typical matching result is given by Fig. 9. It is observed that while the matching error on (1) increases or score on (2) decreases along with the number of separation, both SCCRP algorithms achieved totally correct matchings on all of the 9 baselines. On the other hand, RRWM, PGM, GA, PATH, and FGM also exhibited quite promising performance.

5.3.3 Results on Chinese Characters

In this experiment we conducted the feature correspondence on four hand-written Chinese characters shown in Fig. 10, where each character consists of 10 samples Liu et al. (2011).

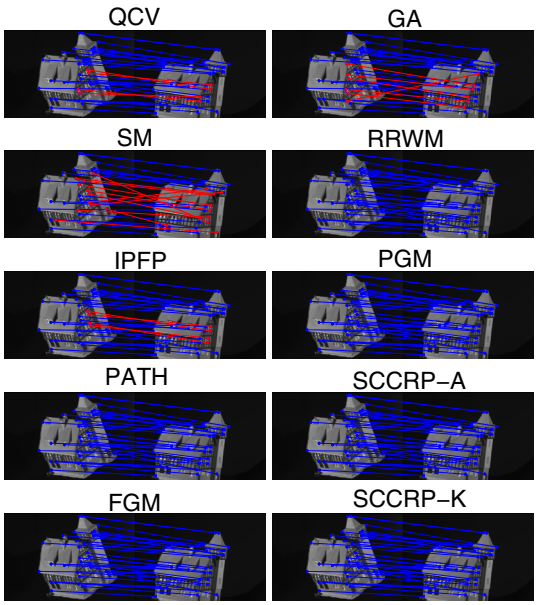

Fig. 9 A typical matching result by the ten algorithms on house sequence (baseline = 80), where *red lines* denote wrong matchings (Color figure online)

We manually labeled 28, 23, 28, and 23 feature points respectively, and constructed their structure representations roughly following their skeletons. The normalized *distance* and *direction* between the edge and horizontal axis with equal weights, i.e., $\alpha_1 = \alpha_2 = 0.5$, were used as the pairwise cues. We tested the algorithms on all of the possible sample pairs, i.e., 45 pairs for each character.

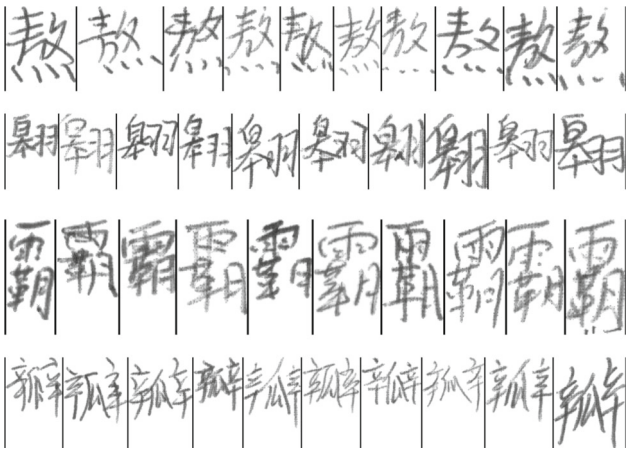

Fig. 10 The four hand-written Chinese characters used in the experiment, where each character consists of 10 samples

The experimental results are shown in Table 5, and one typical correspondence obtained by the two SCCRP algorithms is shown in Fig. 11. It is observed that on accuracy rate, by using objective function (1), the SCCRP_A and PATH achieved 3 and 2 best results respectively, while by objective function (2), the GA and SCCRP_K got 3 and 1 best results respectively. Thus, the SCCRP algorithms got a comparable or even slightly better result than their original CCRP algorithms, i.e., PATH and FGM. On the other hand, the RRWM, IPFP, PGM, and FGM got much better results on accuracy by using (4) than using (3).

5.3.4 Results on Pascal Image Dataset

In the experiment we used the 30 pairs of car images and 20 pairs of motorbike images used in Leordeanu et al. (2012), which were selected from Pascal 2007. The number of manually labeled nodes of each image varies from 15 to 52, and we used the Delaunay triangulation to construct their structure representation. Three types of pairwise constraints used in Leordeanu et al. (2012) were used in the experiment, including the normalized $distance$, the *angle1* between the edge and horizontal axis, and *angle2* between the normal of points i and j, with the weights being $\alpha_1 = 0.5, \alpha_2 = \alpha_3 = 0.25$, respectively.

Both equal-sized and partial correspondence were conducted on the Pascal dataset. Firstly we evaluated the ten

Table 5 Comparative experimental results of ten algorithms on feature correspondence on the Chinese character dataset and Pascal image dataset

data	obj.	results	QCV	GA	SM	RRWM	IPFP	PGM	PATH	SCCRP_A	FGM	SCCRP_K
character1	(1)	accuracy (%)	37.8	–	21.7	32.7	30.3	38.2	82.1	**82.4**	–	–
		average error	8.97	–	15.57	11.90	4.69	11.93	**1.51**	1.53	–	–
	(2)	accuracy(%)	–	**70.3**	31.3	43.3	56.7	47.1	–	–	60.1	68.9
		average score	–	**43.8**	13.9	20.8	40.3	23.9	–	–	43.5	43.7
character2	(1)	accuracy (%)	80.4	–	46.8	61.3	64.7	62.6	88.2	**90.3**	–	–
		average error	4.15	–	17.0	11.6	3.55	11.9	0.69	**0.69**	–	–
	(2)	accuracy(%)	–	87.2	62.1	87.9	83.9	80.6	–	–	86.5	**89.1**
		average score	–	47.7	28.3	47.2	47.0	40.7	–	–	47.2	**48.3**
character3	(1)	accuracy (%)	45.6	–	24.5	30.6	33.8	25.4	**91.4**	**91.4**	–	–
		average error	12.20	–	24.41	22.25	5.56	23.43	**1.56**	**1.56**	–	–
	(2)	accuracy(%)	–	**83.3**	39.9	58.2	65.3	53.2	–	–	77.9	82.1
		average score	–	**64.6**	19.8	36.8	56.5	30.3	–	–	59.5	63.9
character4	(1)	accuracy (%)	60.8	–	31.1	43.7	60.4	43.8	**99.4**	98.3	–	–
		average error	5.87	–	13.83	10.62	2.76	10.58	**1.15**	1.17	–	–
	(2)	accuracy(%)	–	**94.8**	67.5	75.4	81.5	77.4	–	–	89.8	92.0
		average score	–	**42.35**	30.0	35.9	37.9	38.5	–	–	40.2	40.9
car	(1)	accuracy (%)	96.2	–	29.3	50.2	97.2	88.4	**100**	**100**	–	–
		average error	10.20	–	46.9	33.5	6.84	14.9	**4.82**	**4.82**	–	–
	(2)	accuracy(%)	–	95.9	59.5	92.6	96.1	96.7	–	–	**98.2**	96.5
		average score	–	106.5	74.3	102.7	107.1	108.1	–	–	**110.2**	106.9
motorbike	(1)	accuracy (%)	90.4	–	18.9	50.7	90.2	78.9	**100**	**100**	–	–
		average error	5.60	–	40.1	30.1	4.38	9.99	**3.98**	**3.98**	–	–
	(2)	accuracy(%)	–	92.4	63.2	88.0	**98.7**	90.9	–	–	92.3	93.1
		average score	–	130.9	89.9	126.0	**135.0**	130.4	–	–	130.7	131.9

The best result is shown in bold face

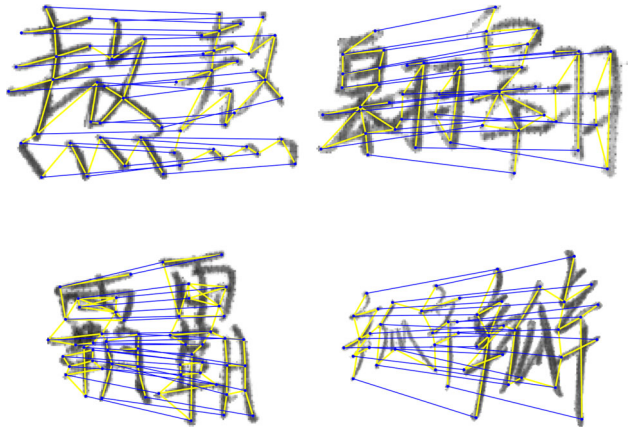

Fig. 11 One typical totally correct correspondence obtained by SCCRP_A and SCCRP_K on Chinese characters, where the *yellow lines* show their structure representations (Color figure online)

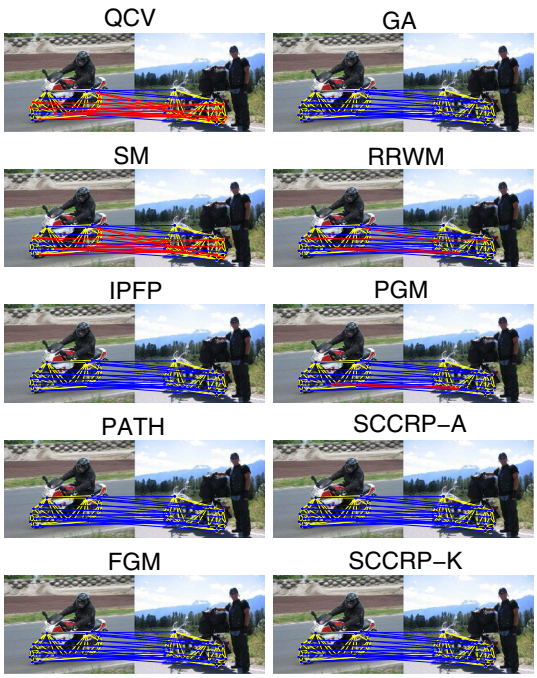

Fig. 12 A typical correspondece result obtained by the ten algorithms on *motorbike* images, where *yellow lines* denote their structure representation and *red lines* denote wrong matchings (Color figure online)

algorithms on equal-sized feature correspondence, and then evaluated the seven algorithms on partial correspondence in the presence of outliers. On the partial correspondence, the number of outliers was increased from 5 to 20 by 5.

The results of the first experiment are listed in Table 5, and one typical correspondence result is shown in Fig. 12. We can observe that, by using objective function (1), the PATH and SCCRP_A got perfectly correct correspondence on both categories, and by using (2) with $\mathbf{K}$ given by (4), while FGM and IPFP got one best result respectively, all the seven algorithms except SM also got promising results. Meanwhile, the SCCRP_K showed a comparable performance to FGM. It is once againg to notice that SM, RRWM, IPFP, and PGM got higher accuracies by using exponential measure (4) than using square error (3).

The results of the partial correspondence with outliers are plotted in Fig. 13 and one typical result with 10 outliers is shown in Fig. 14. We can observe that on matching accuracy, the IPFP, FGM and SCCRP_K respectively got 2, 4, and 2 best results out of the 8 settings. On the other hand, the FGM outperformed SCCRP_K on 4 settings, and SCCRP_K beat FGM on the remaining 4 settings. Thus, the partial correspondence results also revealed a comparable performance of SCCRP_K to FGM.

6 Related Works

The proposed simplified CCRP in this paper is a type of deterministic annealing technique, which is thus related to other deterministic annealing methods, e.g., Geiger and Yuille (1991); Rose (1998), in that all of them typically begin with a rough approximation of the original problem which is usually much easier to solve (usually a convex program), and then gradually transfer to the original problem. More directly, the proposed method has its root in the CCRP Zaslavskiy et al. (2009); Liu et al. (2012); Liu and Qiao (2012) in that both methods approximate the global solution of an optimization over permutation matrix by combining both convex and concave relaxations. However, the concave relaxation in CCRP was constructed in a problem-specific way, which is typically difficult to figure out in practice. By contrast, combined with the convex relaxation the SCCRP provides a problem-independent method to construct the concave relaxation, though the concave relaxation in SCCRP is inherently problem dependent. Some intuitive discussions between the two types of concave relaxations are given as follows. Any relaxation (concave or convex) will certainly reshape the originally relaxed objective function when $\mathbf{P} \in \mathcal{D}$. Both the convex and concave relaxations of CCRP, i.e., (E)PATH and FGM, reshape the original function; on the other hand, the SCCRP takes the same convex relaxation as (E)PATH/FGM, but with the different concave relaxation in (18), which is further simply realized by combining the convex relaxation and $\text{vec}(\mathbf{P})^T \text{vec}(\mathbf{P})$. $\text{vec}(\mathbf{P})^T \text{vec}(\mathbf{P})$ reshapes the function in a symmetric manner, implying somewhat that the bias of SCCRP comes mainly from the convex relaxation. Thus, we may get a rough feel that if the concave relaxation of (E)PATH/FGM provides more biases than its convex relaxation, SCCRP should be better than (E)PATH/FGM, and vice verse. However, the concave relaxation of (E)PATH/FGM takes a quite complicated form, and it is thus difficult to theoretically or even intuitively analyze it.

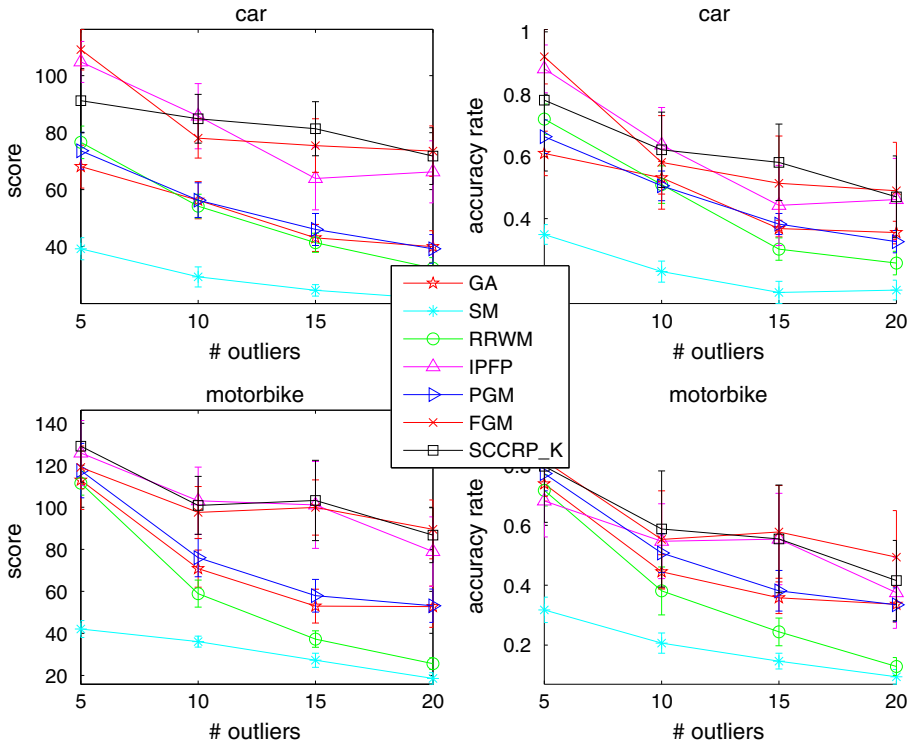

Fig. 13 A typical partial correspondece result obtained by the seven algorithms on *car* images, where the image on the *right-hand* in each pair contains 10 outliers (Color figure online)

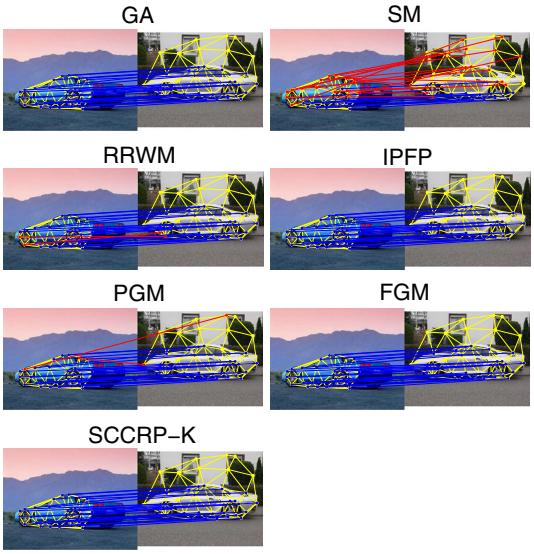

Fig. 14 A typical partial correspondece result obtained by the seven algorithms on *car* images, where the image on the *right-hand* in each pair contains 10 outliers (Color figure online)

In terms of back projection, our work is related to graduated assignment Gold and Rangarajan (1996), which also approaches the original discrete solution in a gradual way, and was also adopted by some other graph matching algorithms Cho et al. (2010). Instead of resorting to a concave relaxation, the graduated assignment introduced a soft-version of iterative conditional modes (ICM) by giving each matching a soft probability by a softmax controlled by a parameter. As the parameter is increased to be large enough, the graduated assignment becomes exactly the maximal linear assignment (9). However, unlike the concave relaxation, the softmax strategy cannot guarantee an equivalence to the original discrete problem, and this is probably the main reason that makes CCRP in general exhibit a better performance than graduated assignment Liu et al. (2012); Liu and Qiao (2012). The recently proposed integer-projected fixed point (IPFP) algorithm Leordeanu et al. (2009) provided also a graduated discretization technique. Similar to the Frank-Wolfe algorithm Frank and Wolfe (1956) adopted by CCRP and SCCRP, the IPFP iterates the same two steps, but further takes the best discrete solution found in the first step, i.e., the linear projection (21) as the final solution. A major difference between IPFP and SCCRP or CCRP is that IPFP attempts to optimize the originally relaxed function, but the objective function of SCCRP or CCRP changes gradually by (10), and finally becomes a concave relaxation of the original problem. Thus, the final discrete solution of SCCRP or CCRP is obtained naturally by this concave program, which is equal to the original discrete program. Regarding to the complexity, because SCCRP or CCRP needs two loops with the inner loop being a Frank-Wolfe algorithm and outer loop for the parameter ζ (or λ in CCRP), SCCRP or CCRP seems to be more computational demanding than IPFP (graduated assignment needs also an outer loop for the control parameter). However, specific to graph matching, the IPFP and some other algorithms

defined on the affinity matrix in general involves a $O(N^4)$ complexity, heavier than the $O(N^3)$ of SCCRP or CCRP, as also revealed by the previous experimental results.

Besides CCRP, there is very limited work in literature that involves concave relaxation, in which Maciel and Costeira (2003) is probably the most related one. It introduced a concave relaxation by constructing a strictly diagonally dominant negative definite Hessian matrix by subtracting N^2 number of large enough constants. Specifically, it introduces a constant ϵ_i for each element $\mathbf{q}_i$ of $\mathbf{q} := \text{vec}(\mathbf{P})$ where ϵ_i satisfies

$$\epsilon_i \geq \frac{1}{2}\left[\max_{\mathbf{q}}\left(\sum_{j=1,j\neq i}^{N^2}|H_{ij}(\mathbf{q})|\right)+\max_{\mathbf{q}}(H_{ii}(\mathbf{q}))\right],$$

and then construct the concave relaxation as

$$F_c(\mathbf{q}) = F(\mathbf{q}) - \epsilon_i\sum_{i=1}^{N^2}\mathbf{q}_i^2 + \epsilon_i\sum_{i=1}^{N^2}\mathbf{q}_i.$$

Therefore, such a construction involves not only an explicit Hessian matrix, but also a $O(N^4)$ complexity. By contrast, the concave relaxation of SCCRP is realized in an implicit way without needing to figure out the Hessian matrix or the constant explicitly.

7 Conclusions and Future Work

As a state-of-the-art optimization algorithm, the CCRP showed superior performance but has difficulties in finding convex or concave relaxation, which greatly limits its practical applications. This paper proposed the simplified CCRP scheme, which needs only convex relaxation, and thus greatly simplifies the process of developing a CCRP algorithm. When being applied to the graph matching problem, two simplified CCRP algorithms were proposed based on two convex relaxations defined respectively on adjacency matrices and affinity matrix. Extensive experimental results validate the efficacy of the two simplified CCRP graph matching algorithms. Future efforts will address the evaluation of the simplified CCRP scheme on some other related problems, such as the MAP inference of the Markov Random Field (MRF).

Acknowledgments The authors thank Dr. Feng Zhou at Carnegie Mellon University for some helpful discussions on his factorized graph matching algorithm Zhou and De la Torre (2012). Many thanks also go to the anonymous reviewers and associate editor whose comments and suggestions greatly improved the manuscripts. This work was supported by the National Science Foundation of China (NSFC) (grants 61375005, 61033011, 61210009), and by Singapore MOE tier 1 research grant (RG33/11).

Appendix: Proof of Corollary 1

To prove Corollary 1, we prove $-\mathbf{x}^\top\mathbf{K}\mathbf{x} = \| \mathbf{A}_D - \mathbf{X}\mathbf{A}_M\mathbf{X}^\top \|_F^2$, where $\mathbf{K}$ is given by (3), and $\mathbf{x} = \text{vec}(\mathbf{X})$. Writing the partial permutation matrix $\mathbf{X} \in \mathbb{R}^{N\times M}$ as

$$\mathbf{X}^\top = [\mathbf{e}_{\pi(1)}, \mathbf{e}_{\pi(2)}, \ldots \mathbf{e}_{\pi(N)}]$$

where $\mathbf{e}_{\pi(i)}$ denotes a column vector of length M with 1 at the position $\pi(i)$ and 0 every other position, it can be then shown that

$$\begin{aligned}-\mathbf{x}^\top\mathbf{K}\mathbf{x} &= -\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_{i\pi(i),j\pi(j)}\\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}(A_{Dij} - A_{M\pi(i)\pi(j)})^2\end{aligned} \tag{30}$$

On the other hand, the term $\mathbf{X}\mathbf{A}_M\mathbf{X}^\top$ can be equivalently written as $\{\mathbf{A}_{M\pi(i)\pi(j)}\}^{N\times N}$, and consequently,

$$\begin{aligned}\| \mathbf{A}_D - \mathbf{X}\mathbf{A}_M\mathbf{X}^\top \|_F^2 &= \| \mathbf{A}_D - \{\mathbf{A}_{M\pi(i)\pi(j)}\}^{N\times N} \|_F^2\\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}(A_{Dij} - A_{M\pi(i)\pi(j)})^2\end{aligned} \tag{31}$$

Thus, $-\mathbf{x}^\top\mathbf{K}\mathbf{x} = \| \mathbf{A}_D - \mathbf{X}\mathbf{A}_M\mathbf{X}^\top \|_F^2$, and the proof is accomplished. □

References

Blake, A., & Zisserman, A. (1987). *Visual Reconstruction*. Cambridge, MA, USA: MIT Press.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. New York: Cambridge University Press.

Cho, M., Alahari, K., Ponce, J., et al. (2013). Learning graphs to match. In: ICCV 2013-IEEE International Conference on Computer Vision.

Cho, M., Lee, J., & Lee, K. M. (2010). *Reweighted random walks for graph matching. In: Computer Vision-ECCV 2010*. Berlin: Springer.

Cho, M., Lee, K.M. (2012). Progressive graph matching: Making a move of graphs via probabilistic voting. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 398–405. IEEE.

Conte, D., Foggia, P., Sansone, C., & Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *18*(3), 265–298.

Cour, T., Srinivasan, P., & Shi, J. (2007). Balanced graph matching. *Advances in Neural Information Processing Systems*, *19*, 313.

Demirci, M. F., Shokoufandeh, A., Keselman, Y., Bretzner, L., & Dickinson, S. (2006). Object recognition as many-to-many feature matching. *International Journal of Computer Vision*, *69*(2), 203–222.

Duchenne, O., Joulin, A., Ponce, J. (2011). A graph-matching kernel for object categorization. IEEE International Conference on Computer Vision pp. 1792–1799.

Egozi, A., Keller, Y., & Guterman, H. (2013). A probabilistic approach to spectral graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 18–27.

Fischler, M. A., & Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, *C–22*(1), 67–92.

Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, *3*(1–2), 95–110.

Geiger, D., & Yuille, A. (1991). A common framework for image segmentation. *International Journal of Computer Vision*, *6*(3), 227–243.

Gold, S., & Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(4), 377–388.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1–2), 83–97.

Leordeanu, M., & Hebert, M. (2005). A spectral technique for correspondence problems using pairwise constraints. *Tenth IEEE International Conference on Computer Vision*, *2*, 1482–1489.

Leordeanu, M., Herbert, M., Sukthankar, R. (2009). An integer projected fixed point method for graph matching and map inference. Advances in Neural Information Processing Systems p. 1114C1122.

Leordeanu, M., Sukthankar, R., & Hebert, M. (2012). Unsupervised learning for graph matching. *International journal of computer vision*, *96*(1), 28–45.

Liu, C. L., Yin, F., Wang, D. H., & Wang, Q. F. (2011). Casia online and offline chinese handwriting databases. In: Preceedings of the International Conference on Document Analysis and Recognition, *2011*, 37–41.

Liu, Z. Y., & Qiao, H. (2012). A convex-concave relaxation procedure based subgraph matching algorithm. *Journal of Machine Learning Research: W&CP*, *25*, 237–252.

Liu, Z. Y., Qiao, H., & Xu, L. (2012). An extended path following algorithm for graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(7), 1451–1456.

Maciel, J., & Costeira, J. P. (2003). A global solution to sparse correspondence problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(2), 187–199.

Philbin, G., Sivic, J., & Zisserman, A. (2011). Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *International Journal of Computer Vision*, *95*(2), 138–153.

Ravikumar, P., Lakerty, J. (2006). Quadratic programming relaxations for metric labeling and markov random field map estimation. International Conference on Machine Learning.

Rose, K. (1998). Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, *86*(11), 2210–2239.

Suh, Y., Cho, M., & Lee, K. M. (2012). *Graph matching via sequential monte carlo. In: Computer Vision-ECCV 2012*. Berlin: Springer.

Tian, Y., Yan, J., Zhang, H., Zhang, Y., Yang, X., & Zha, H. (2012). *On the convergence of graph matching: graduated assignment revisited. In: Computer Vision-ECCV 2012*. Berlin: Springer.

Torresani, L., Kolmogorov, V., Rother, C. (2008). Feature correspondence via graph matching: Models and global optimization. In D. Forsyth, P. Torr, A. Ziseerman (eds.), *ECCV 2008, Part II, LNCS 5303*, (pp. 596–609).

Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(5), 695–703.

Zaslavskiy, M., Bach, F., & Vert, J. P. (2009). A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(12), 2227–2242.

Zhou, F., De la Torre, F. (2012). Factorized graph matching. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 127–134.

Zhou, F., De la Torre, F. (2013). Deformable graph matching. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 2922–2929. IEEE.