On the Convergence of Graph Matching: Graduated Assignment Revisited

Yu Tian[1], Junchi Yan[1,⋆], Hequan Zhang[3]
Ya Zhang[1], Xiaokang Yang[1], and Hongyuan Zha[2]

[1] Shanghai Jiao Tong University, 200240 Shanghai, China
[2] Georgia Institute of Technology, 30032 Atlanta, USA
[3] Beijing Institute of Technology, 100081 Beijing, China
{tianyu0124,ethanyan1985}@gmail.com, zhanghequan1987@126.com,
{ya_zhang,xkyang}@sjtu.edu.cn, zha@cc.gatech.edu

Abstract. We focus on the problem of graph matching that is fundamental in computer vision and machine learning. Many state-of-the-arts frequently formulate it as integer quadratic programming, which incorporates both unary and second-order terms. This formulation is in general NP-hard thus obtaining an exact solution is computationally intractable. Therefore most algorithms seek the approximate optimum by relaxing techniques. This paper commences with the finding of the "*circular*" character of solution chain obtained by the iterative *Gradient Assignment* (via Hungarian method) in the discrete domain, and proposes a method for guiding the solver converging to a fixed point, resulting a convergent algorithm for graph matching in discrete domain. Furthermore, we extend the algorithms to their counterparts in continuous domain, proving the classical graduated assignment algorithm will converge to a double-circular solution chain, and the proposed Soft Constrained Graduated Assignment (SCGA) method will converge to a fixed (discrete) point, both under wild conditions. Competitive performances are reported in both synthetic and real experiments.

1 Introduction and Problem Formulation

In computer vision and machine learning, many tasks that require finding correspondences between two node sets can be formulated as graph matching such that the relations between nodes can be preserved as much as possible. Although extensive research [1] has been done for decades, graph matching is still challenging mainly due to two reasons: 1) In general, the objective function is non-convex and prone to local optimum; 2) the solution needs to satisfy combinatorial constraints. The graph matching problem are widely formulated as integer quadratic programming (IQP), by considering both unary and second-order terms reflecting the local similarities together with the pairwise relations. Concretely, given two graphs $G^L(V^L, E^L, A^L)$ and $G^R(V^R, E^R, A^R)$, where V denotes nodes, E, edges and A, attributes, there is an affinity matrix defined as $\mathbf{M}_{ia;jb}$ that measures the affinity with the candidate edge pair (v_i^L, v_j^L) vs. (v_a^R, v_b^R).

⋆ The work is supported by NSF IIS-1116886, NSF IIS-1049694, NSFC 61129001/F010403 and the 111 Project (B07022).

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part III, LNCS 7574, pp. 821–835, 2012.

And the diagonal term $\mathbf{M}_{ia;ia}$ describes the unary affinity of a node match (v_i^L, v_a^R). By introducing a permutation matrix $\mathbf{x} \in \{0,1\}^{n_l \times n_R}$ whereby $\mathbf{x}_{ia} = 1$ if node V_i^L matches node V_a^R (and $\mathbf{x}_{ia} = 0$ otherwise), it leads to the following formulation:

$$\mathbf{x}^* = \arg_{\mathbf{x}} \max(\mathbf{x}^T \mathbf{M} \mathbf{x}) \quad s.t. \quad \mathbf{A}\mathbf{x} = \mathbf{1} \quad \mathbf{x} \in \{0,1\} \tag{1}$$

here $\mathbf{x}$ is deformed into vectorized version, constraints $\mathbf{A}\mathbf{x} = \mathbf{1}$ refer to the one-to-one matching from G^L to G^R such that one feature from one image can be matched to at most one other feature from the other image. The difficulty depends on the structure of matrix $\mathbf{M}$, in general it is NP-hard and no polynomial algorithm exists.

2 Algorithm and Theories

We commence with a simple gradient based method - *Gradient Assignment (GA)*: given a initial value $\mathbf{x}_0$, relax the objective from the quadratic assignment $\mathbf{x}_k \mathbf{M} \mathbf{x}_k$ to a linear assignment in an iterative fashion: $\mathbf{x}_{k+1} \mathbf{M} \mathbf{x}_k$, whereby the Hungarian method [8] (denoted by H_d in the sequel) is performed to obtain the discrete solution in each iteration: $\mathbf{x}_{k+1}$= $H_d(\mathbf{M}\mathbf{x}_k)$. We will show it induces a "circular" solution sequence given that the returned solution from Hungarian method is *unique* - as described in Assumption 1:

Assumption 1 *(Δ_{min})*

$$\forall i; \forall j, k, j \neq k; (\boldsymbol{p}_j - \boldsymbol{p}_k)^T \boldsymbol{M} \boldsymbol{p}_i \neq 0 \tag{2}$$

$\boldsymbol{p}_i$,$\boldsymbol{p}_j$,$\boldsymbol{p}_k$ are vectorized permutation matrix. Δ_{min} equals the smallest difference among nonzero results of Equ.2: $\Delta_{min} = \min_{p_j, p_k, p_i, j \neq k} |(\boldsymbol{p}_j - \boldsymbol{p}_k)^T \boldsymbol{M} \boldsymbol{p}_i|$

Theorem 1. *If Assumption 1 holds, after k finite number of iterations, the gradient assignment (GA) method will converge to a sequence $\{\boldsymbol{p}_i, \boldsymbol{p}_{i+1}, \boldsymbol{p}_i \cdots\}$ or $\{\boldsymbol{p}_i, \boldsymbol{p}_i, \boldsymbol{p}_i \cdots\}$.*

Proof: For an ascending solution path induced by Hungarian method: $\mathbf{p}_{i+1}^T \mathbf{M} \mathbf{p}_i = \mathbf{p}_i^T \mathbf{M} \mathbf{p}_{i+1} \leq \mathbf{p}_{i+2}^T \mathbf{M} \mathbf{p}_{i+1} \leq \cdots \leq \mathbf{p}_{i+k+1}^T \mathbf{M} \mathbf{p}_{i+k} = \mathbf{p}_{i+1}^T \mathbf{M} \mathbf{p}_i$: 1) there must exist a k satisfying $\mathbf{p}_{i+k} = \mathbf{p}_i$ due to that the finite permutation space can be covered by a large enough k; 2) Assumption 1 ensures equal scores denote identical solutions. □

This "circular" property is unwanted due to the relaxation largely deviates the original objective $\mathbf{x}^T \mathbf{M} \mathbf{x}$ when $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$ are not close. Thus we are motivated to add a constraint term $||\mathbf{x}_k - \mathbf{x}_{k-1}|| < \delta$, in the spirit of encouraging converging to a fixed point. Using L_2 norm, one can obtain the following formulation:

$$\max_{\mathbf{x}_{k+1}} \mathbf{x}_{k+1}^T (\mathbf{M} + \lambda \mathbf{I}) \mathbf{x}_k - \lambda \mathbf{x}_{k+1}^T \mathbf{x}_{k+1} - \lambda \mathbf{x}_k^T \mathbf{x}_k \tag{3}$$

We drop the quadratic terms for three reasons: a) the new quadratic terms increase the difficulty for optimizing; b) $\mathbf{x}^{*T} \mathbf{x}^* = n$ holds for one-to-one matching as considered in this paper; c) the resultant quadratic-free formulation will not change the optimal solution against the original objective since the deviated score associated with the diagonal $\lambda \mathbf{I}$ plays an equal role to all matching candidates:

$$\max_{\mathbf{x}_{k+1}} \mathbf{x}_{k+1}^T (\mathbf{M} + \lambda * \mathbf{I}) \mathbf{x}_k \tag{4}$$

Using Equ. 4 the *Constrained Gradient Assignment (CGA)* algorithm is proposed in Alg. 1. It always measures the current solution via the *original* affinity matrix $\mathbf{M}_0$ (the updted $\mathbf{M}_k$ is only used for the Hungarian method's input to obtain a new solution) and select the best along the whole path. Theorem 2 explores its convergence property.

Algorithm 1. Constrained Gradient Assignment (CGA)

```
Initial: $\lambda = \lambda^*$; $k = 0$; $\mathbf{x}_0 = [\frac{1}{n^2}, ..., \frac{1}{n^2}]^T$; $\mathbf{x}^* = \mathbf{x}^*_{-1} = \mathbf{x}_0$; $\mathbf{M} = \mathbf{M}_0$; $score = 0$
repeat
    $(\mathbf{x}_{k+1} = H_d(\mathbf{M}\mathbf{x}_k))$;
    if ($\mathbf{x}^T_{k+1}\mathbf{M}_0\mathbf{x}_{k+1} >$score) then
        $score = \mathbf{x}^T_{k+1}\mathbf{M}_0\mathbf{x}_{k+1}$; $\mathbf{x}^* = \mathbf{x}_{k+1}$; $\mathbf{x}^*_{-1} = \mathbf{x}_k$;
    end if
    if $(\mathbf{x}_{k+1} == \mathbf{x}_k)$ then
        return: $\mathbf{x}^*$;
    else if $(\mathbf{x}_{k+1} == \mathbf{x}_{k-1})$ then
        $\mathbf{x}_{k+1} = \mathbf{x}^*$; $\mathbf{x}_k = \mathbf{x}^*_{-1}$; $\mathbf{M} = \mathbf{M} + \lambda\mathbf{I}$;
    end if
    k++;
until $k$ exceeds max iteration: return $\mathbf{x}^*$;
```

Theorem 2. *CGA converges to a fixed point provided on Assumption 1(Δ_{min}) holds.*

Proof: As stated in Theorem 1, when Assumption 1(Δ_{min}) holds, for a fixed $\mathbf{M}$, the GA algorithm must converge in the form of $\mathbf{p}_{k+1} = \mathbf{p}_{k-1}$. Meanwhile, $\mathbf{M}$ is updated as $\mathbf{M} = \mathbf{M} + \lambda\mathbf{I}$. After a finite number of rounds, $\mathbf{M}$ would be converted to a symmetric positive definite matrix ($\mathbf{M}$ is symmetric). If CGA converges before this positive definite matrix, the proof completes, else $\mathbf{M}$ can be decomposed into $\mathbf{M} = \mathbf{M}_{dec}\mathbf{M}^T_{dec}$. Without loss of generality, assume the iteration sequence converges to $\{\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_i ...\}$. Thus we can reach:

$$\begin{aligned} \mathbf{p}^T_{i+1}\mathbf{M}\mathbf{p}_i &= H_d(\mathbf{M}\mathbf{p}_i)^T\mathbf{M}\mathbf{p}_i \geq \mathbf{p}^T_i\mathbf{M}\mathbf{p}_i \\ \mathbf{p}^T_i\mathbf{M}\mathbf{p}_{i+1} &= H_d(\mathbf{M}\mathbf{p}_{i+1})^T\mathbf{M}\mathbf{p}_{i+1} \geq \mathbf{p}^T_{i+1}\mathbf{M}\mathbf{p}_{i+1} \end{aligned} \tag{5}$$

And by summing up the two above inequations we have:

$$2\langle\mathbf{M}^T_{dec}\mathbf{p}_{i+1}, \mathbf{M}^T_{dec}\mathbf{p}_i\rangle \geq \langle\mathbf{M}^T_{dec}\mathbf{p}_{i+1}, \mathbf{M}^T_{dec}\mathbf{p}_{i+1}\rangle + \langle\mathbf{M}^T_{dec}\mathbf{p}_i, \mathbf{M}^T_{dec}\mathbf{p}_i\rangle \tag{6}$$

Equation holds *i.f.f.* $\mathbf{p}^T_{i+1}\mathbf{M}\mathbf{p}_{i+1}$=$\mathbf{p}^T_{i+1}\mathbf{M}\mathbf{p}_i$=$\mathbf{p}^T_i\mathbf{M}\mathbf{p}_i$, using Assumption 1: $\mathbf{p}_i$=$\mathbf{p}_{i+1}$. □

CGA is sensitive to initial point and sometimes would trap to an unsatisfied point shortly. To address this issue, on one hand, by observing the affinity matrix may be distorted after finishing CGA, in order to extend the solution chain, we can frame an outer loop whereby the current iteration (CGA)'s output is the input to the next iteration where CGA is also performed. And the outer loop will stop if the solution with respect to the *original* objective cannot be improved by the new iteration - we call this outer loop CGA algorithm as *LCGA*. On the other hand, the Hungarian method confined in the discrete domain misses many optimums, which could otherwise have been explored by a "soft" counterpart in the continuous domain. Thus we leverage the tool of "softmax" [6] (instead of the Hungarian method) that has been applied in the classical Graduate Assignment algorithm (GAGM) [2], and extend CGA to its "soft" version: *Soft Constrained Graduate Assignment (SCGA)* described in Alg.2. The algorithmic procedure of Alg.2 is empirically established based on GAGM and CGA, yet we are more interested in the underlying question: will the continuous methods (SCGA/GAGM) have similar converge property as their discrete counterparts (CGA/GA)? Affirmative answer is given in Theorem 3 and Corollary 1 and Corollary 2 provided on Assumption 1 in addition with Assumption 2 hold, which is described in below:

Algorithm 2. Soften Constrained Gradient Assignment (SCGA)

```
Initial: $\beta = \beta_0$; $\lambda = \lambda^*$; $k = 0$; $\mathbf{x}_0 = [\frac{1}{n^2}, ..., \frac{1}{n^2}]^T$; $\mathbf{x}^* = \mathbf{x}^*_{-1} = H_d(\mathbf{M}\mathbf{x}_0)$; $\mathbf{M} = \mathbf{M}_0$; $score = 0$; $\varepsilon = \varepsilon_0$
repeat
    $\mathbf{x}_{k+1} = \text{SoftMax}\ (\mathbf{M}\mathbf{x}_k)$;
    increase $\beta$;
    if $(H_d(\mathbf{M}\mathbf{x}_k)^T \mathbf{M}_0 H_d(\mathbf{M}\mathbf{x}_k) \geq score)$ then
        $score = H_d(\mathbf{M}\mathbf{x}_k)^T \mathbf{M}_0 H_d(\mathbf{M}\mathbf{x}_k)$; $\mathbf{x}^* = \mathbf{x}_k$; $\mathbf{x}^*_{-1} = \mathbf{x}_{k-1}$;
    end if
    if $(\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon)$ then
        return: $H_d(\mathbf{M}\mathbf{x}^*)$
    end if
    if $(\|\mathbf{x}_{k+1} - \mathbf{x}_{k-1}\| < \varepsilon)$ then
        $\mathbf{x}_{k+1} = \mathbf{x}^*$; $\mathbf{x}_k = \mathbf{x}^*_{-1}$; $\mathbf{M} = \mathbf{M} + \lambda\mathbf{I}$;
    end if
    k++;
until $k$ exceeds max iteration: return $\mathbf{x}^*$;
```

Assumption 2. *($s_j, T_j, \varDelta$) Let $T = \frac{1}{\beta}$ where β is the parameter in SoftMax. During the iterations, there exist $\mathbf{s}_j$, T_j, $\varDelta$ satisfying* [1]*:*

$$\forall i \neq j+1; (\boldsymbol{p}_{j+1} - \boldsymbol{p}_i)^T \boldsymbol{M}\boldsymbol{s}_j \geq \varDelta \tag{7}$$

$$\frac{T_j}{\varDelta} \alpha n N^2 \log N < \frac{\varDelta_{min}}{2} \tag{8}$$

$$0 < \varDelta \leq \varDelta_{min} - 2\frac{T_j}{\varDelta} \alpha n N^2 \log N \tag{9}$$

[1] $\varDelta_{min}$ is defined in Assumption 1, $N = n^2$ where n is the number of nodes in one graph, $\varDelta$ and α are defined in Lemma 4 while $\mathbf{s}_j$ and $\mathbf{p}_j$ are defined in Equ.10 and Equ.11 in the appendix respectively.

Theorem 3. *If Assumption 1(Δ_{min}) and Assumption 2 ($\boldsymbol{s}_j, T_j, \Delta$) hold, using the continuous solution chain $\{\boldsymbol{s}_j\}$ as defined in Equ. 10, the induced discrete solution chain $\{\boldsymbol{p}_j\}$ defined by Equ. 11 will asymptotically (if only j is big enough) satisfy $\boldsymbol{s}_{j+1} = \boldsymbol{p}_{j+1}$, and $\{\boldsymbol{p}_j\}$ will satisfy the definition of the Hungarian method: $\boldsymbol{p}_{j+k+1} \triangleq H_d(\boldsymbol{M}\boldsymbol{p}_{j+k})$.*

Theorem 3 directly leads two important resultant corollaries:

Corollary 1. *During the iteration, if there exists $\boldsymbol{s}_j$ s.t. Assumption 1(Δ_{min}) and Assumption 2 ($\boldsymbol{s}_j, T_j, \Delta$) hold, then GAGM will converge to a double-circular solution chain.*

Corollary 2. *During the iteration, if there exists $\boldsymbol{s}_j$ s.t. Assumption 1(Δ_{min}) and Assumption 2 ($\boldsymbol{s}_j, T_j, \Delta$) hold, then SCGA will converge to a fixed discrete point.*

Proof details can be found in the appendix.

3 Related Work

Early work by Christmas, Kittler and Petrou [9] and Wilson and Hancock [10] showed how relaxation labeling could be used to the graph matching problem by modeling the probabilistic distribution of matching errors. Drawing on ideas from these connectionist literature, Gold and Rangarajan [2] developed a relaxation scheme based on soft-assign, which gradually updates the derivative of the relaxed IQP i.e. Graduate Assignment (GAGM) [2]. Leordeanu and Hebert [3] present a spectral matching (SM) method which computes the leading eigenvector of symmetric nonnegative affinity. The constraint are entirely dropped and they only consider the integer constraints in the final discretization step. Later, spectral graph matching with affine constraints was developed [11] (SMAC). In addition, [11] suggests a preprocessing on the input matrix, by normalizing it into a doubly stochastic matrix. More recently, Cho et al. [4] apply random walk and introduce a reweighting jump scheme to incorporate mapping constraints. None of these aforementioned approaches are concerned with the original integer constraints during optimization while they assume the final continuous solution can lead to an optimal feasible solution after discretization. Contrary to this idea, an iterative matching method (IPFP) is proposed by Leordeanu et al. [5]. In their method, the optimized solution $\mathbf{x}_k$ in each iteration is continuous, yet they keep the discrete one $\mathbf{b}_k$ induced by $\mathbf{x}_k$ if it suffices the optimal condition. When the iteration ends, the solution is selected among the stored discrete set instead from the binarization of the continues solution in the final iteration. This paper addresses the general learning-free graph matching problem. In terms of convergence study as most emphasized in this paper, Rangarajan et al [7] show that GAGM will converge to a discrete point provided the affinity matrix is positive definite, while having no clear answer for general cases. This paper clarifies that for the general graph matching problem, GAGM will converge to a circular solution chain, and progressively making additive to the diagonal of the affinity matrix will result in an algorithm that is deemed to converge to a fixed discrete point after a finite number of iterations, under mild conditions. Motivated by this theoretical finding, we further carefully design the mechanism for when and how much the addition is performed during iterations. In general, the proposed methods prolong the searching path in both discrete and continuous domain, avoiding early trap to unwanted point. Moreover, the proposed mechanism keeps the "relaxed" solver off deviating from the original objective function thus improves the solution quality.

4 Experiments and Discussion

Choice of Step Size and Max Iterations: We keep the CGA/SCGA's max iterations as 500, and LCGA's as 10. As the maximum value of input affinity matrix $\mathbf{M}$ is 1, from Gershgorin circle theorem, we know when the diagonal element of $\mathbf{M}$ increase to N, the matrix become positive definite, thus the step size of λ is $\frac{N}{500}$.

Experiments on both synthetic and real image are designed to evaluate the proposed algorithms (SCGA, LCGA) on various graph matching tasks against state-of-the-art methods: SM [3], GAGM [2], IPFP [5], RRWM [4]. For RRWM and IPFP, the publicly available codes by [4] were used, and SM, GAGM were implemented by self. The experimental methodology and design details are quite similar with the protocol in [4] for fair comparison. All methods were implemented using MATLAB and tested on a 2.67G HZ PC. For each trial, the same affinity matrix was shared as the input and Hungarian algorithm was commonly used at final discretization step for all methods. Each quantitative result in the synthetic experiments was acquired from averages of 20 random trials. Due to the common character of LCGA and IPFP that can refine other approaches' output as their input, we tested them independently, as well as in conjunction with others as a post-processing step. When tested as standalone, LCGA and IPFP are always initialized with a flat, uniform solution. Control parameters of GAGM, RRWM was based on authors' papers and tuned for better performance. For SCGA, we set the same β as [4] and λ, k as discussed before in all experiments.

Synthetic Random Graph: We follow exactly the same experimental protocol of [2, 4, 11]. For each matching trial, two graphs are constructed, reference graph G^L with $n^L = n_{in}$ nodes and the perturbed graph G^R with $n^R = n_{in} + n_{out}$. The perturbed graph G^R is created by permuting nodes order of n_{in} and adding Gaussian noises.The affinity matrix $\mathbf{M}$ is constructed by $\mathbf{M}_{ia,jb}$ = exp$(-\|d^L_{ij} - d^R_{ab}\|/\sigma_s)$ where σ_s is set to 0.15 as the same value in [4]. The performances of each method are measured in both accuracy and scores as defined in [4]. Based on the aforementioned settings, we conduct two sub-experiments to verify the robustness against deformation and outlier. In the deformation test, we change the deformation noise parameter ε from 0 to 0.4 with the interval 0.05, to compare with RRWM on a larger baseline, we set inlier number n_{in} to 50 and 60. In outlier noise experiment, the number of outliers n_{out} varies from 0 to 20 by increments of 2, while fix n_{in} to 50. The experimental results are plotted in Fig. 1. As observed from the deformation test, SCGA outperforms in accuracy while LCGA and RRWM are comparable. And LCGA achieves a higher score given increased noise. These phenomenon may imply a continuous technique like SoftMax is a key factor against large noises. In contrast, a looping scheme like LCGA pays more attention to objective score. In case of small noise, the proposed two methods are both very fast, yet becoming relative slow in the existence of larger noise. Considering efficiency, all algorithms are stopped when the iteration exceeds a fixed threshold. In outlier test, both

SCGA and LCGA outperform others in accuracy and score, and in score test LCGA is better again. It suggests that LCGA is suitable when noise is small as in this case a better score always means a better match. We also tested our LCGA with its corresponding simple version CGA and Gradient Assignment (GA). The results in Fig. 1 verify the efficacy of gradually adding constrains and outer looping. Note CGA shows good performance and relative low time cost.

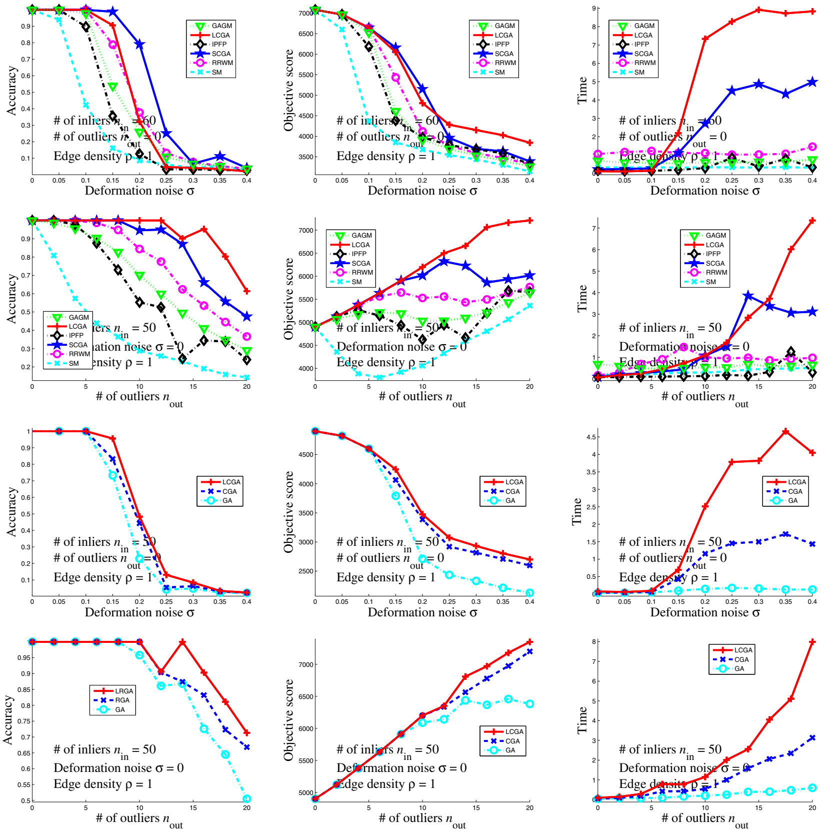


Fig. 1. Evaluation on synthetic random graphs. Two top rows: Deformation and outlier evaluation among GAGM, LCGA, IPFP, SCGA, RRWM, SM; Two bottom rows: Deformation and outlier evaluation among GA, CGA, LCGA.

Performance of Being a Post Step: As shown in Fig. 2, in this test we use IPFP and LCGA as a post processor using the results from RRWM, SM, SCGA and GAGM as their initial input, called RRWM-IPFP and so on. Solid line means LCGA for post processing while Dashes denotes IPFP. We find LCGA outperforms IPFP in all methods except for SM, which is worth our future study. Aware SCGA-IPFP and SCGA-LCGA are comparable and achieve the best results, as similar as in the outlier test.

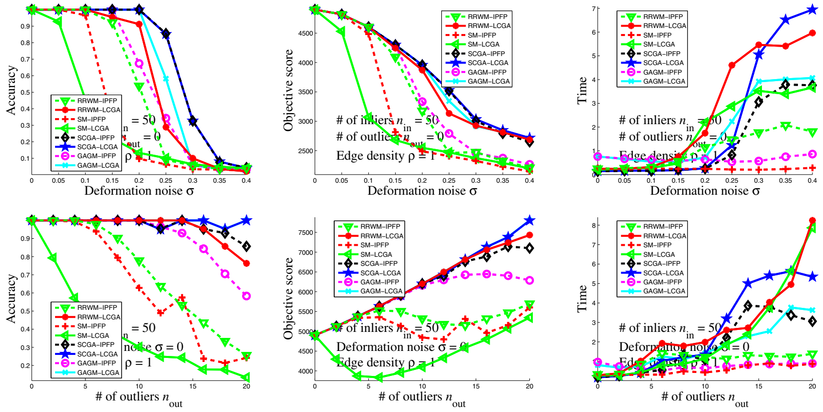
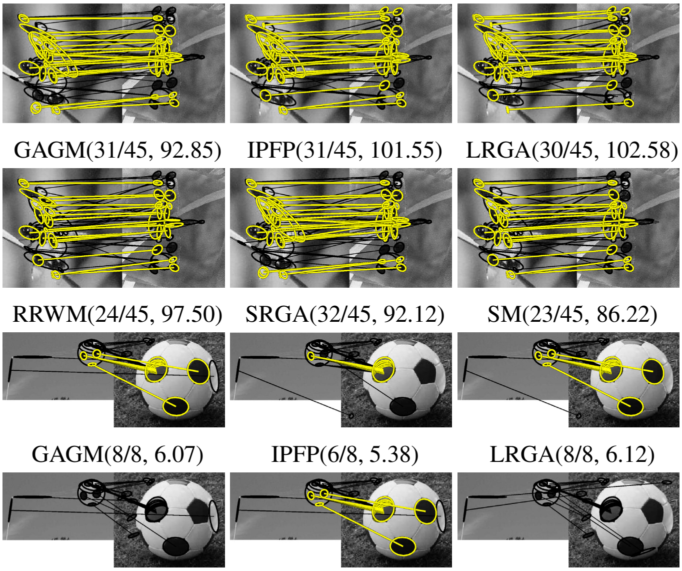


Fig. 2. Performance evaluation of using LCGA and IPFP as a post processor optimizing with the result from other algorithms as their initial input

Real Image: In the second image experiment, as the same with [4, 12], 30 image pairs from Caltech-101 and MSRC datasets are selected from which feature points are extracted by MSER detector and SIFT descriptor. We followed the setup from [4, 12] exactly. The complexity of this experiment is due to the large intra-category variations in shape and large number of outliers, posing the challenge regarding noise and outlier. Some comparative examples are shown in Fig. 3, and the average performance comparison on the dataset is summarized in Table 1, the results show LCGA outperform others, especially in score while SCGA is comparable with the state-of-the-art algorithms. We also compare our algorithm with IPFP as being post step in this test, the result is similar to synthetic experiment, both methods can elevate the performance, especially in score. LCGA is slightly better than IPFP in most cases except SM. Both SCGA-IPFP and SCGA-LCGA achieve competitive results in accuracy and score.

Table 1. Evaluation of various matching algorithms. (+IPFP) and (+LCGA) denote perform post IPFP and LCGA using other methods' output as their input.

Method	GAGM	LCGA	IPFP	SCGA	RRWM	SM
Accuracy(%)	72.45	75.84	73.6	71.2	73.61	62.58
Score(%)	91.06	98.69	94.63	92.21	93.01	79.63
Time(S)	0.23	0.38	0.12	0.36	0.31	0.03
Accuracy(+IPFP)	75.82			74.78	76.28	71.32
Accuracy(+LCGA)	75.88			75.12	74.94	62.58
Score(+IPFP)	99.51			99.15	97.23	92.45
Score(+LCGA)	99.87			99.78	97.84	79.62



GAGM(31/45, 92.85) IPFP(31/45, 101.55) LRGA(30/45, 102.58)

RRWM(24/45, 97.50) SRGA(32/45, 92.12) SM(23/45, 86.22)

GAGM(8/8, 6.07) IPFP(6/8, 5.38) LRGA(8/8, 6.12)

RRWM(0/8, 5.41) SRGA(8/8, 6.07) SM(0/8, 4.56)

Fig. 3. Evaluation on real images, in the bracket: method name, accuracy, and score

Conclusion: We have proposed a novel framework and induced algorithms for graph matching, whose convergence are also proved. The experiments show that it outperforms the state-of-the-art methods in the presence of outliers and deformation, especially when the number of graph node is large. The comparison reveals that the matching accuracy and convergence rate in the challenging situations largely depends on the effective exploitation of the matching constraints in the searching space.

References

1. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. IJPRAI (2004)
2. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. PAMI (1996)
3. Leordeanu, M., Hebert, M.: A Spectral Technique for Correspondence Problems using Pairwise Constraints. In: ICCV (2005)
4. Cho, M., Lee, J., Lee, K.M.: Reweighted Random Walks for Graph Matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 492–505. Springer, Heidelberg (2010)
5. Leordeanu, M., Herbert, M.: An integer projected fixed point method for graph matching and map inference. In: NIPS (2009)
6. Gold, S., Rangarajan, A.: Softmax to softassign: neural network algorithms for combinatorial optimization. J. Artif. Neural Netw., 3810–399 (1995)
7. Rangarajan, A., Yuille, A.: Convergence properties of the softassign quadratic assignment algorithm. Neural Computation (1999)

8. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)
9. Christmas, W., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. PAMI 19(6), 634–648 (1997)
10. Wilson, R., Hancock, E.: Structural matching by discrete relaxation. IEEE Trans. Pattern Anal. Mach. Intell. 17(8), 749–764 (1995)
11. Cour, T., Srinivasan, P., Shi, J.: Balanced Graph Matching. In: NIPS (2006)
12. Lee, J., Cho, M., Lee, K.M.: Hyper-graph Matching via Reweighted RandomWalks. In: CVPR (2011)
13. Kosowsky, J.J., Yuille, A.L.: The Invisible Hand Algorithm: Solving the Assignment Problem With Statistical Physics. Neural Networks 7(3) (1994)

Appendix: Proof Details of Theorem 3

First we introduce some preliminaries to facilitate later proving process: Let the initial value for SCGA be $\mathbf{s}_0$, and define the solution chain $\{\mathbf{s}_j\}$, $\{\mathbf{p}_j\}$ as:

$$\mathbf{s}_{j+1} \triangleq \text{SoftMax}(\mathbf{M}\mathbf{s}_j) \tag{10}$$

$$\mathbf{p}_{j+1} \triangleq H_d(\mathbf{M}\mathbf{s}_j) \tag{11}$$

The proof of GA and CGA are both based on Assumption 1(Δ_{min}), though Assumption 1(Δ_{min}) seems strong while our following analysis shows that it is built on an even more weaker precondition as shown in the following:

$$\forall e \in \mathbf{M}, \quad \text{exist} \quad s, k \in Z \qquad s.t. \quad e = s \times 10^{-k} \tag{12}$$

Equ.12 suggests any element e in matrix $\mathbf{M}$ can be measured in a bounded precision by k. From a theoretical perspective, it excludes the *irrational* and *circulator*; yet from a practical perspective based on current computer architecture, all the elements are truncated in a bounded precision. Thus in practice, Equ.12 always holds.

We then show a technique by which Assumption 1(Δ_{min}) would always hold, and in turn the convergence properties of GA and CGA are ensured.

Lemma 1. *If Assumption 1(Δ_{min}) does not hold, one can add specks to **M** such that satisfies the presumption Equ.12 and induces the establishment of Assumption 1(Δ_{min}) and keep $\mathbf{p}^T\mathbf{M}\mathbf{p}$ almost unchanged (the slight distortion to **M** will not change the optimum property of the original score). Furthermore, such distortion is constructible.*

Proof: Assume the precision of **M**'s elements is 10^{-k} (Equ.12). And the number of the elements in **M** is N^2. One can add specks on **M**: $10^{-k-1-K} * 2^{-1}$,$10^{-k-1-K} * 2^{-2}$,...,$10^{-k-1-K} * 2^{-N^2}$ to obtain the new matrix $\mathbf{M}_{new}$. Thus for $\forall i; \forall j, k, j \neq k$, each element in $\mathbf{p}_j - \mathbf{p}_k$ is from the discrete set $\{-1, 0, 1\}$ and each item in $\mathbf{p}_i$ is $\{0, 1\}$. As a result, $(\mathbf{p}_j - \mathbf{p}_k)^T\mathbf{M}\mathbf{p}_i$ is the linear combination of the elements of original **M** plus the additional specks. When $(\mathbf{p}_j - \mathbf{p}_k)^T\mathbf{M}\mathbf{p}_i = 0$ holds for the original **M**, we can see that the new $\mathbf{M}_{new}$ is nonequal to zero due to the additional specks whose corresponding $(\mathbf{p}_j - \mathbf{p}_k)^T(\mathbf{M}_{new} - \mathbf{M})\mathbf{p}_i$ (linear combination) cannot be zero in that each basis is $2^{-L}, L \in N^2$ while the coefficients are $\{-1, 0, 1\}$. Furthermore, when k is big enough, the precision of $(\mathbf{p}_j - \mathbf{p}_k)^T(\mathbf{M}_{new} - \mathbf{M})\mathbf{p}_i$ is less than the original function, so the original result that is nonequal to zero is still nonzero after this modification. □

Using Lemma 1 and Theorem 1, Theorem 2 ensure the convergence for GA, CGA respectively. In what follows, we will present the proof flow details of the convergence of SCGA and GAGM as stated in the main text.

Lemma 2. *($\boldsymbol{s}_j, T$) Given a fixed β, or $T = 1/\beta$. SoftMax($\boldsymbol{Ms}_j$) must be converged to the solution $\mathbf{s}_{j+1}$, whose matrix form is doubly stochastic.*

Lemma 3. *(T) Given a fixed T, let $\boldsymbol{p}_{j+1} = H_d(\boldsymbol{Ms}_j)$, we have: $\boldsymbol{p}_{j+1}^T\boldsymbol{Ms}_j - TN\log N \leq \boldsymbol{s}_{j+1}^T\boldsymbol{Ms}_j \leq \boldsymbol{p}_{j+1}^T\boldsymbol{Ms}_j$ where $N = n^2$ and n is the number of graph node.*

Lemma 4. *($\boldsymbol{s}_j, T, \Delta$) Given $\boldsymbol{Ms}_j$, if there exists a* unique *best solution: $\boldsymbol{p}_{j+1}$=$H_d(\boldsymbol{Ms}_j)$ returned by the Hungarian method, let $\boldsymbol{p}_i$ be the second best one (discrete), and let Δ=$|(\boldsymbol{p}_{j+1} - \boldsymbol{p}_i)^T\boldsymbol{Ms}_j|$ then we have:*

$$\max_i |\boldsymbol{p}_{j+1}^i - \boldsymbol{s}_{j+1}^i| \leq \frac{TN\log N}{\Delta} \tag{13}$$

where $\boldsymbol{p}_{j+1}^i$ denotes the ith element in the vectorized permutation matrix $\boldsymbol{p}_{j+1}$.

Lemma 2, Lemma 3, Lemma 4 have been proven by Kosowsky and Yuille in [13]. And one can find Assumption 2 ($\mathbf{s}_j, T_j, \Delta$) derives the establishment of Lemma 4 ($\mathbf{s}_j, T_j, \Delta$).

Lemma 5. *($\boldsymbol{s}_j, T_j, \Delta$) If Assumption 1 ($\Delta_{min}$), Assumption 2($\boldsymbol{s}_j, T_j, \Delta$) hold then*

$$\forall i, |\boldsymbol{p}_i^T\boldsymbol{Mp}_{j+1} - \boldsymbol{p}_i^T\boldsymbol{Ms}_{j+1}| \leq \frac{T_j}{\Delta}\alpha n N^2 \log N \tag{14}$$

Proof: Obviously Lemma 4 ($\mathbf{s}_j, T_j, \Delta$) holds.

$$|\mathbf{p}_i^T\mathbf{Mp}_{j+1} - \mathbf{p}_i^T\mathbf{Ms}_{j+1}| = |\mathbf{p}_i^T\mathbf{M}(\mathbf{p}_{j+1} - \mathbf{s}_{j+1})| \leq \mathbf{p}_i^T\alpha\mathbf{E}\frac{T_j}{\Delta}\mathbf{C} = \frac{T_j}{\Delta}\alpha n N^2 \log N$$

where $\mathbf{E}$ is the $N \times N$ matrix whose elements are all one, and $\mathbf{C} = [N\log N, ..., N\log N]^T$. The above equation is based on the observation that $\mathbf{p}_i$ is a vectorized permutation matrix (so it contains n 1s), and the elements of $\mathbf{M}$ are bounded within $[0, \alpha]$.

□

Lemma 6. *($\boldsymbol{s}_j, T_j, \Delta$) If Assumption 1($\Delta_{min}$), 2($\boldsymbol{s}_j, T_j, \Delta$) hold, then $\boldsymbol{p}_{j+2}$=H_d ($\boldsymbol{Mp}_{j+1}$).*

Proof: By contradiction: Obviously Lemma 5 holds, if Lemma 6 does not hold, due to Assumption 1(Δ_{min}), there exists i, s.t. $\mathbf{p}_{j+2}^T\mathbf{Mp}_{j+1} < \mathbf{p}_i^T\mathbf{Mp}_{j+1}$, and

$$\mathbf{p}_{j+2}^T\mathbf{Mp}_{j+1} \leq \mathbf{p}_i^T\mathbf{Mp}_{j+1} - \Delta_{min} \tag{15}$$

According to Lemma 5, we have

$$\begin{aligned}\mathbf{p}_i^T\mathbf{Mp}_{j+1} - \Delta_{min} &\leq \mathbf{p}_i^T\mathbf{Ms}_{j+1} - \Delta_{min} + \frac{T_j}{\Delta}\alpha n N^2 \log N \\ \mathbf{p}_{j+2}^T\mathbf{Ms}_{j+1} - \frac{T_j}{\Delta}\alpha n N^2 \log N &\leq \mathbf{p}_{j+2}^T\mathbf{Mp}_{j+1}\end{aligned} \tag{16}$$

By combining the above three equations we have:

$$\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{s}_{j+1} - \mathbf{p}_j^T\mathbf{M}\mathbf{s}_{j+1} + \varDelta_{min} \leq 2\frac{T_j}{\varDelta}\alpha n N^2 \log N \tag{17}$$

By the definition $\mathbf{p}_{j+2} \triangleq H_d(\mathbf{M}\mathbf{s}_{j+1})$, we have $\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{s}_{j+1} - \mathbf{p}_j^T\mathbf{M}\mathbf{s}_{j+1} \geq 0$, By Assumption 2 ($\mathbf{s}_j, T_j, \varDelta$) have: $\frac{T_j}{\varDelta}\alpha n N^2 \log N < \frac{\varDelta_{min}}{2}$. Thus contradiction exists. □

Lemma 7. *If Assumption 1($\varDelta_{min}$) and Assumption 2 ($\boldsymbol{s}_j, T_j, \varDelta$) hold, we have:*

$$\forall i \neq j+2, \quad \boldsymbol{p}_{j+2}^T\boldsymbol{M}\boldsymbol{s}_{j+1} - \boldsymbol{p}_i^T\boldsymbol{M}\boldsymbol{s}_{j+1} \geq \varDelta \tag{18}$$

i.e. Assumption 2 ($\boldsymbol{s}_{j+1}, T_j, \varDelta$) also holds.

Proof: Obviously Lemma 5 ($\mathbf{s}_j, T_j, \varDelta$) & Lemma 6 ($\mathbf{s}_j, T_j, \varDelta$) hold. By Lemma 5:

$$\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{s}_{j+1} \geq \mathbf{p}_{j+2}^T\mathbf{M}\mathbf{p}_{j+1} - \frac{T_j}{\varDelta}\alpha n N^2 \log N \tag{19}$$

By Assumption 1($\varDelta_{min}$) and Lemma 6 ($\mathbf{s}_j, T_j, \varDelta$) we have:

$$\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{p}_{j+1} - \frac{T_j}{\varDelta}\alpha n N^2 \log N \geq \mathbf{p}_i^T\mathbf{M}\mathbf{p}_{j+1} + \varDelta_{min} - \frac{T_j}{\varDelta}\alpha n N^2 \log N \tag{20}$$

By Lemma 5 ($\mathbf{s}_j, T_j, \varDelta$) we have:

$$\mathbf{p}_i^T\mathbf{M}\mathbf{p}_{j+1} + \varDelta_{min} - \frac{T_j}{\varDelta}\alpha n N^2 \log N \geq \mathbf{p}_i^T\mathbf{M}\mathbf{s}_{j+1} + \varDelta_{min} - 2\frac{T_j}{\varDelta}\alpha n N^2 \log N$$

Combining the three equations we reach:

$$\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{s}_{j+1} \geq \mathbf{p}_i^T\mathbf{M}\mathbf{s}_{j+1} + \varDelta_{min} - 2\frac{T_j}{\varDelta}\alpha n N^2 \log N \tag{21}$$

By Assumption 2 ($\mathbf{s}_j, T_j, \varDelta$), $\varDelta \leq \varDelta_{min} - 2\frac{T_j}{\varDelta}\alpha n N^2 \log N$, the lemma holds. □

Lemma 8. *If Assumption 2 ($\boldsymbol{s}_{j+1}, T_j, \varDelta$) holds, Assumption 2 ($\boldsymbol{s}_{j+1}, T_{j+1}, \varDelta$) holds.*

Proof: Since $T_{j+1} < T_j$, the below relations hold, either thus Assumption 2 ($\mathbf{s}_{j+1}, T_{j+1}, \varDelta$).

$$\frac{T_{j+1}}{\varDelta}\alpha n N^2 \log N < \frac{T_j}{\varDelta}\alpha n N^2 \log N < \frac{\varDelta_{min}}{2} \tag{22}$$

$$0 < \varDelta < \varDelta_{min} - 2\frac{T_j}{\varDelta}\alpha n N^2 \log N < \varDelta_{min} - 2\frac{T_{j+1}}{\varDelta}\alpha n N^2 \log N \tag{23}$$

□

We commence with the satisfaction of Assumption 2 ($\mathbf{s}_j, T_j, \varDelta$) at a specific iteration j, and then prove Assumption 2 ($\mathbf{s}_{j+1}, T_{j+1}, \varDelta$) will also hold in the next iteration. Thus for each $k > 0$ in the later iterations, Assumption 2 ($\mathbf{s}_{j+k}, T_{j+k}, \varDelta$) will always hold and this leads to our main results: here we prove our main theoretical conclusion as shown in Theorem 3 in the main text.

Proof: For $k = 0, 1, ...$ Lemma 4 ($\mathbf{s}_{j+k+1}, T_{j+k+1}, \Delta$) and Lemma 6 ($\mathbf{s}_{j+k+1}, T_{j+k+1}, \Delta$) obviously hold. By Lemma 6 ($\mathbf{s}_{j+k}, T_{j+k}, \Delta$), we know $\mathbf{p}_{j+k+2} = H_d(\mathbf{M}\mathbf{p}_{j+k+1})$. According to Assumption 1 (Δ_{min}) and Lemma 4 ($\mathbf{s}_{j+k+1}, T_{j+k+1}, \Delta$), when k is big enough, we have sequence $\mathbf{s}_{j+k}$ will converged to discrete sequence $\mathbf{p}_{j+k}$. □

Now we prove the different convergence property of the classical GAGM and the proposed SCGA as has been stated in Corollary 1 and Corollary 2 in the main text.

Proof: By combining Theorem 1 and Theorem 3, we know GAGM will converged to circular sequence. By combining Theorem 2 and Theorem 3, we know when **M** becomes positive definite, SCGA will converge to a fixed discrete point. □

Remark 1. (The wildness of Assumption 2) According to Equ.8 and Equ.9, Δ lies in:

$$[\frac{\Delta_{min} - \sqrt{\Delta_{min}^2 - 8T_j\alpha n N^2 \log N}}{2}, \frac{\Delta_{min} + \sqrt{\Delta_{min}^2 - 8T_j\alpha n N^2 \log N}}{2}] \quad (24)$$

When $T_j \mapsto 0$, this formula imply $\Delta \in (0, \Delta_{min})$, also when T_j goes to zero, the SoftMax result $SoftMax(\mathbf{s}_j)$ approaches to discrete result (vectorized permutation matrix). From Lemma 1 we know as $\mathbf{s}_j$ goes to discrete result, the score difference between best solution $\mathbf{p}_{j+1}$ and suboptimal solution $\mathbf{p}_i$ approaches Δ_{min}, thus after several iteration, we can choose$\Delta = \frac{\Delta_{min}}{2}$, which lies in the range $(0, \Delta_{min})$ - and smaller than the difference between best and suboptimal score, thus the existence of Assumption 2 is assured. In addition, based on our previous derivation, for a given $\mathbf{s}_j$, once there exists a Δ sufficing the condition for Assumption 2, all later $\mathbf{s}_{j+1}$ would satisfy the constraint. These facts consolidate our judgement that Assumption 2 is weak. Furthermore, we have a more concrete observation as described in Lemma 9.

Lemma 9. *By adding specks to **M**, there exist $\mathbf{s}_1$ and T_1, s.t. Assumption 2 holds.*

Proof: One can select $\mathbf{s}_0$=$(\frac{1}{N^2}, \dots, \frac{1}{N^2})^T$, obviously Δ lies within $[\frac{1}{N^2}\Delta_{min}, \frac{N^2-1}{N^2}\Delta_{min}]$, and then we can select an appropriate T_1 s.t. Δ lies in the scope of Equ.24. □

At the end of this appendix, we give conceptual illustrations showing the idea of our proof, and suppose $T \mapsto 0$ without loss of generality. First we introduce the initial status. The dotted arrows mean the generate process, red dotted arrows denote iterative step of GAGM [2], where $\mathbf{s}_{j+1} = \text{SoftMax}(\mathbf{M}\mathbf{s}_j)$; blue dotted arrows denote discrete optimum where $\mathbf{p}_{j+1} = H_d(\mathbf{M}\mathbf{s}_j)$. Yellow solid lines mean the corresponding function score. The bidirectional-arrow means the quantity relationship between targets. Dashed bidirectional-arrow denotes the "Gap" relationship while solid bidirectional-arrow denotes the "asymptotic" relationship.

As illustrated in Fig.4, according to Lemma 3 we know as T approximates to 0, for all k, $\mathbf{s}_{j+k+1}^T\mathbf{M}\mathbf{s}_{j+k}$ approaches to $\mathbf{p}_{j+k+1}^T\mathbf{M}\mathbf{s}_{j+k}$. Based on Assumption 2 ($\mathbf{s}_j, T_j, \Delta$) we know there is a gap between optimal score $\mathbf{p}_{j+1}^T\mathbf{M}\mathbf{s}_j$ and any other score. Thus from

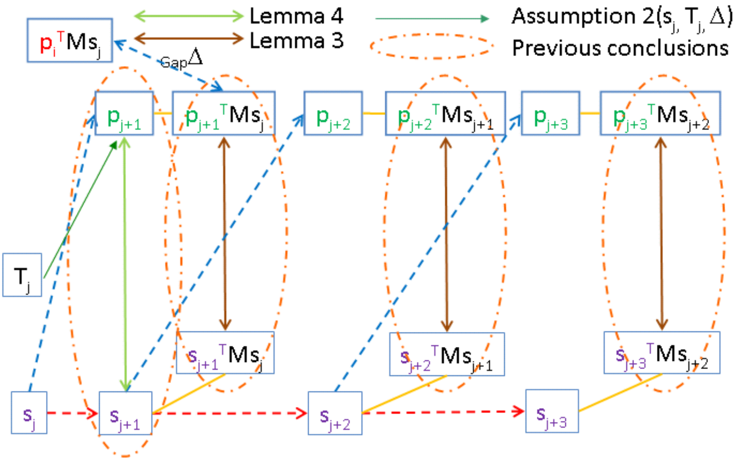


Fig. 4. Conceptualization of applying [13]'s conclusions to the proposed iteration chain

Lemma 4 we get the "asymptotic" relationship between $\mathbf{s}_{j+1}$ and $\mathbf{p}_{j+1}$. The green curve arrow illustrates the role of Assumption 2 - here we use T_j to represent Assumption 2($\mathbf{s}_j, T_j, \varDelta$). Kosowsky and Yuille focus on the connection of the Hungarian solution and the SoftMax one, which is first order version versus the pairwise graph matching task we address in this paper - a quadratic assignment problem solved by transforming it into a series of iterative first-order assignment subproblems. By repeatedly applying their conclusion to the separate steps in the quadratic case, we know that starting from $\mathbf{s}_j$, each score $\mathbf{s}_{j+k+1}^T\mathbf{M}\mathbf{s}_{j+k}$ would approximate to $\mathbf{p}_{j+k+1}^T\mathbf{M}\mathbf{s}_{j+k}$ (Roughly speaking, the approximation of two scores do not necessarily ensure $\mathbf{s}_{j+k+1} \mapsto \mathbf{p}_{j+k+1}, \forall k$, while we show in Fig.5 that the answer is affirmative if only Assumption 2 holds for the first iteration). In addition, under the precondition of Assumption 2 ($\mathbf{s}_j, T_j, \varDelta$), according the conclusion from [13], one can obtain a single $\mathbf{s}_{j+1}$ that approaches $\mathbf{p}_{j+1}$. Note that [13]'s conclusion does not address the chain property in the quadratic assignment problem where first-order assignment is performed in each iteration, thus involving multiple ($\mathbf{s}_{j+k+1}, \mathbf{p}_{j+k+1}$) pairs. In another saying, if directly applying their conclusion to the chain case, one has to check the existence of a specific and separate $\varDelta_k$ that builds the asymptotic relation between respective $\mathbf{s}_{j+k+1}, \mathbf{p}_{j+k+1}$ in each iteration - and further the convergence property of the iterative algorithm. As an extension, we strictly analyze and give the affirmative answer in terms of convergence for this iteration chain, under the rather weak precondition: if only Assumption 2 ($\mathbf{s}_j, T_j, \varDelta$) holds once in one beginning iteration, say j, then in all of the next iterations, the updated Assumption 2 ($\mathbf{s}_{j+k+1}, T_{j+k+1}, \varDelta$) holds, and $\mathbf{s}_{j+k+1}$ is bound to approximate to $\mathbf{p}_{j+k+1}$.

We also prove that $\mathbf{p}_{j+k+2}$ is the Hungarian solution of $\mathbf{M}\mathbf{p}_{j+k+1}$, thus for other $\mathbf{p}_i$, the score gap exists - see the dotted eclipses for attention (thus the discrete sequence $\mathbf{p}_{j+k+1}$ induced by $\mathbf{s}_{j+k}$ composes an iteration chain by self). We show the proof flow in Fig.5 and summarize key steps as following:

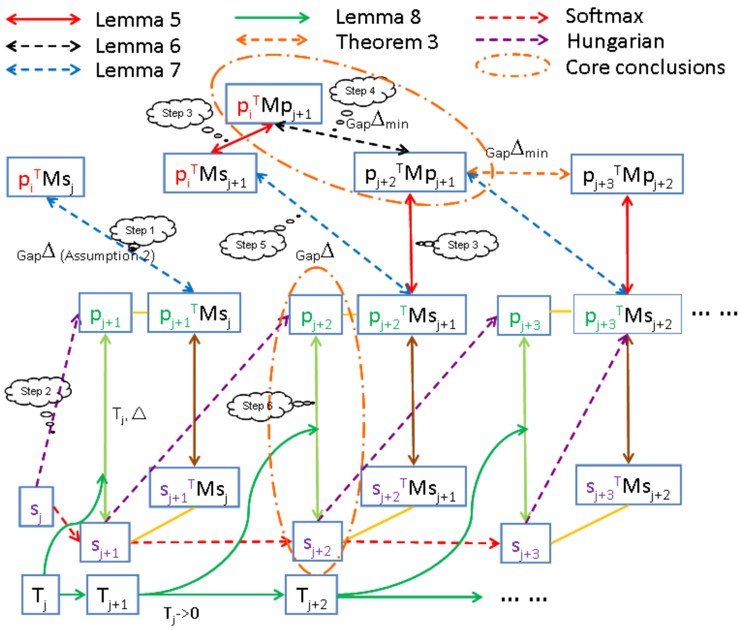


Fig. 5. Conceptualization of proof flow for the iteration chain in the pairwise graph matching problem. The main idea is exploring the connection between continuous solution and discrete one: $\mathbf{s}_j \mapsto \mathbf{p}_j$, as well as $\mathbf{p}_{j+1} = H_d(\mathbf{M}\mathbf{p}_j)$, where $\mathbf{p}_{j+1}$ is indeed calculated from $H_d(\mathbf{M}\mathbf{s}_j)$. In this spirit, SCGA's convergence proof is similar to the one of CGA.

1. Using Assumption 2, suppose there is a gap between function score of optimal solution $\mathbf{p}_{j+1}$ and suboptimal $\mathbf{p}_i$;
2. From Lemma 4 we know $\mathbf{p}_{j+1} \mapsto \mathbf{s}_{j+1}$, which we will use in the next step;
3. Prove the "asymptotic" relationship between $\mathbf{p}_i^T\mathbf{M}\mathbf{s}_{j+1}$ and $\mathbf{p}_i^T\mathbf{M}\mathbf{p}_{j+1}$ (also $\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{s}_{j+1}$ and $\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{p}_{j+1}$) by Lemma 5, as illustrated by red solid bidirectional-arrows;
4. In Lemma 6, we proved $\mathbf{p}_{j+2}$ is the Hungarian solution of $\mathbf{M}\mathbf{p}_{j+1}$, thus $\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{p}_{j+1}$ is bigger than any $\mathbf{p}_i^T\mathbf{M}\mathbf{p}_{j+1}$, according to Assumption 1, there exists a gap Δ_{min} between them, as illustrated by black dashed bidirectional-arrows;
5. We know $\mathbf{p}_i^T\mathbf{M}\mathbf{s}_{j+1} \mapsto \mathbf{p}_i^T\mathbf{M}\mathbf{p}_{j+1}$ and $\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{s}_{j+1} \mapsto \mathbf{p}_{j+2}^T\mathbf{M}\mathbf{p}_{j+1}$, using gap Δ_{min} we obtain gap Δ in Lemma 7;
6. Update T_j to T_{j+1} in Lemma 8, thus Assumption 2 $(\mathbf{s}_{j+1}, T_{j+1}, \Delta)$ holds, and obtain "asymptotic" relationship between $\mathbf{p}_{j+2}$ and $\mathbf{s}_{j+2}$;
7. Finally, by combining above results, Theorem 3 proves actually discrete sequence $\mathbf{p}_{j+k}$ is a Hungarian sequence, thus there exists a gap Δ_{min} between adjacent discrete scores (indicated by the nodes $\mathbf{p}_{j+2}^T\mathbf{M}\mathbf{p}_{j+1}$, $\mathbf{p}_{j+3}^T\mathbf{M}\mathbf{p}_{j+2}$ in Fig.5).