Linear Reweighted Regularization Algorithms for Graph Matching Problem

Rongxuan Li[†]

Project advisor: Prof. Yangyang Xu[‡]

Abstract. The graph matching problem is a sparse optimization problem, which is also a significant special case of the Quadratic Assignment Problem, with extensive applications in pattern recognition, computer vision, protein alignment, and related fields. As the problem is NP-hard, relaxation and regularization techniques are frequently employed to improve tractability. However, many existing regularizers are nonconvex which poses optimization challenges. In this paper, we propose a linear reweighted regularization framework for solving the relaxed graph matching problem while preserving convexity. By solving a sequence of relaxed problems with the linear reweighted regularization term, one can obtain a sparse solution. Moreover, we establish the local convergence of the proposed method to the solution of the original graph matching problem. Furthermore, we present a practical version of the algorithm by incorporating the projected gradient method. The proposed framework is applied to synthetic instances and demonstrates promising numerical results.

1. Introduction.

The Quadratic Assignment Problem (QAP), a key optimization problem over permutation matrices, is one of the hardest combinatorial optimization problems and has wide applications in fields such as statistics, facility layout, and chip design [7, 6]. Specifically, the QAP involves the assignment of a set of facilities to a set of locations in such a way that minimizes a given cost function, and can be expressed as:

$$\min_{\mathbf{X}\in\Pi_n} \operatorname{tr}(\mathbf{A}^\top\mathbf{X}\mathbf{B}\mathbf{X}^\top), \tag{1.1}$$

where Π_n is the set of n-order permutation matrices, namely, $\Pi_n = \{\mathbf{X} \in \mathbb{R}^{n\times n} \mid \mathbf{X}\mathbf{e} = \mathbf{X}^\top\mathbf{e} = \mathbf{e}, \mathbf{X}_{ij} \in \{0,1\}\}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times n}$, $\mathbf{e} \in \mathbb{R}^n$ is a vector of all ones. As the QAP Problem is NP-hard [15], both vertex-based methods and interior-point methods have been proposed to address this challenging problem. Vertex-based methods [1, 8, 16] update iterates directly from one permutation matrix to another. Alternatively, interior-point methods relax the nonconvex permutation matrix constraint Π_n to a doubly stochastic matrix constraint $\mathcal{D}_n$, where $\mathcal{D}_n = \left\{\mathbf{X} \in \mathbb{R}^{n\times n} \mid \mathbf{X}\mathbf{e} = \mathbf{X}^\top\mathbf{e} = \mathbf{e}, \mathbf{X}_{ij} \geq 0\right\}$. To enhance sparsity in solutions derived from this relaxation, various regularization techniques have been introduced. Xia [17] proposed L_2 regularization approach, Huang [9] proposed quartic term regularization, and Jiang et al [10] suggested using L_p norms [11]. These regularization techniques are generally nonconvex, which can pose computational challenges. While adding a nonconvex term to the relaxed problem may improve solution quality, it does not necessarily lead to greater computational efficiency.

Notice that when $\mathbf{X} \in \Pi_n$,

$$\|\mathbf{A}\mathbf{X} - \mathbf{X}\mathbf{B}\|_{\mathsf{F}}^2 = -2\operatorname{tr}(\mathbf{A}^\top\mathbf{X}\mathbf{B}\mathbf{X}^\top) + \|\mathbf{A}\|_{\mathsf{F}}^2 + \|\mathbf{B}\|_{\mathsf{F}}^2, \tag{1.2}$$

[†]Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China (1155173725@link.cuhk.edu.hk).

[‡]Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York, USA (xuy21@rpi.edu).

where $\|\mathbf{A}\|_{\mathsf{F}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$. Hence, the QAP problem can be expressed in the form of a graph matching problem. In this paper, we focus on graph matching problem:

$$\min_{\mathbf{X}\in\Pi_n} \|\mathbf{AX}-\mathbf{XB}\|_{\mathsf{F}}^2. \tag{1.3}$$

We relax the the permutation matrix constraint Π_n of graph matching problem to the doubly stochastic matrix constraint $\mathcal{D}_n$ and obtain the relaxed graph matching problem (2.1). We first extend nonconvex regularization terms and demonstrate that the relaxed problem (2.1), when combined with certain concave regularization term, can achieve equivalence between the relaxed and original graph matching problem when regularization term is sufficiently large. However, incorporating a nonconvex term into the relaxed problem may enhance solution quality but does not necessarily improve computational efficiency. To better utilize the convexity of the relaxed problem which is shown in Proposition 2.1, we propose a novel algorithmic framework based on linear reweighted regularization to preserve the convexity of the relaxed problem. Inspired by the effectiveness of reweighted L_1 minimization in promoting sparsity [5], our approach solves a sequence of relaxed problems. We show that, under certain conditions on the initial guess, the model can achieve equivalence to the original graph matching problem. We design a solver based on the projected gradient method and incorporate line search techniques to enhance computational efficiency. Additionally, we introduce the mathematical formulation of the network alignment problem, illustrating how graph matching can address problems in fields such as social networks and computer vision. Finally, we validate our approach through numerical experiments on synthetic full-rank dense datasets, demonstrating promising results across a variety of instances.

2. Regularization.

2.1. Relaxation.

The graph matching problem (1.3) can be relaxed to an optimization problem with n *th* order doubly stochastic matrix constraint $\mathcal{D}_n$:

$$\min_{\mathbf{X}\in\mathcal{D}_n} \|\mathbf{AX}-\mathbf{XB}\|_{\mathsf{F}}^2. \tag{2.1}$$

The relaxed problem is continuous optimization problem and has several properties.

Proposition 2.1. *Let* $f(\mathbf{X}) = \|\mathbf{AX}-\mathbf{XB}\|_{\mathsf{F}}^2$, *then* $f(\mathbf{X})$ *is convex on* $\mathbb{R}^{n\times n}$.

Proof. Define the function $g(\mathbf{Y}) = \|\mathbf{Y}\|_{\mathsf{F}}^2$. Its gradient and Hessian are given by

$$\nabla g(\mathbf{Y}) = 2\mathbf{Y}, \quad \nabla^2 g(\mathbf{Y}) = 2\mathbf{I}.$$

Since the Hessian $\nabla^2 g(\mathbf{Y})$ is a constant positive definite matrix, it follows that $g(\mathbf{Y})$ is convex. Define the function h as

$$h(\mathbf{X}) = \mathbf{AX}-\mathbf{XB}.$$

Since $g(\mathbf{Y})$ is convex and $h(\mathbf{X})$ is a linear transformation, the composition $f(\mathbf{X}) = g(h(\mathbf{X}))$ is convex. ∎

By the Birkhoff-von Neumann theorem [3], we know that $\mathcal{D}_n$ is the convex hull of Π_n, and the set of vertices of $\mathcal{D}_n$ is exactly Π_n. Hence, $f(\mathbf{X})$ is convex on $\mathcal{D}_n$.

Proposition 2.2. *Let $f(\mathbf{X}) = \|\mathbf{AX} - \mathbf{XB}\|_{\mathsf{F}}^2$. Then f is Lipschitz continuous on $\mathcal{D}_n$. A candidate Lipschitz constant is $L = 2n\big(\|\mathbf{A}\|_{\mathsf{F}} + \|\mathbf{B}\|_{\mathsf{F}}\big)^2$.*

Proof. We compute

$$\begin{aligned}|f(\mathbf{X}) - f(\mathbf{Y})| &= |\|\mathbf{AX} - \mathbf{XB}\|_{\mathsf{F}}^2 - \|\mathbf{AY} - \mathbf{YB}\|_{\mathsf{F}}^2| \\ &= |\,\langle(\mathbf{AX} - \mathbf{XB}) + (\mathbf{AY} - \mathbf{YB}),\, (\mathbf{AX} - \mathbf{XB}) - (\mathbf{AY} - \mathbf{YB})\rangle\,| \\ &\le \|\mathbf{A}(\mathbf{X} + \mathbf{Y}) - (\mathbf{X} + \mathbf{Y})\mathbf{B}\|_{\mathsf{F}} \cdot \|\mathbf{A}(\mathbf{X} - \mathbf{Y}) - (\mathbf{X} - \mathbf{Y})\mathbf{B}\|_{\mathsf{F}}.\end{aligned}$$

Applying the triangle inequality, we obtain

$$\begin{aligned}\|\mathbf{A}(\mathbf{X} + \mathbf{Y}) - (\mathbf{X} + \mathbf{Y})\mathbf{B}\|_{\mathsf{F}} &\le \|\mathbf{A}\|_{\mathsf{F}}\,\|\mathbf{X} + \mathbf{Y}\|_{\mathsf{F}} + \|\mathbf{B}\|_{\mathsf{F}}\,\|\mathbf{X} + \mathbf{Y}\|_{\mathsf{F}}, \\ \|\mathbf{A}(\mathbf{X} - \mathbf{Y}) - (\mathbf{X} - \mathbf{Y})\mathbf{B}\|_{\mathsf{F}} &\le \|\mathbf{A}\|_{\mathsf{F}}\,\|\mathbf{X} - \mathbf{Y}\|_{\mathsf{F}} + \|\mathbf{B}\|_{\mathsf{F}}\,\|\mathbf{X} - \mathbf{Y}\|_{\mathsf{F}}.\end{aligned}$$

Therefore,

$$|f(\mathbf{X}) - f(\mathbf{Y})| \le \big(\|\mathbf{A}\|_{\mathsf{F}} + \|\mathbf{B}\|_{\mathsf{F}}\big)^2\,\|\mathbf{X} + \mathbf{Y}\|_{\mathsf{F}}\,\|\mathbf{X} - \mathbf{Y}\|_{\mathsf{F}}.$$

Finally, since $0 \le \mathbf{X}_{ij}, \mathbf{Y}_{ij} \le 1$, we have

$$-2 \le \mathbf{X}_{ij} + \mathbf{Y}_{ij} \le 2,$$

which implies

$$\|\mathbf{X} + \mathbf{Y}\|_{\mathsf{F}}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{X}_{ij} + \mathbf{Y}_{ij})^2 \le \sum_{i=1}^{n}\sum_{j=1}^{n} 2^2 = 4n^2, \quad \Rightarrow \quad \|\mathbf{X} + \mathbf{Y}\|_{\mathsf{F}} \le 2n.$$

Hence,

$$|f(\mathbf{X}) - f(\mathbf{Y})| \le 2n\big(\|\mathbf{A}\|_{\mathsf{F}} + \|\mathbf{B}\|_{\mathsf{F}}\big)^2\|\mathbf{X} - \mathbf{Y}\|_{\mathsf{F}}.$$ ∎

By carefully choosing regularization term and adding it to (2.1), one can better ensure the sparsity of the solution. More specifically, by adding $h(\mathbf{X})$ and $\lambda \ge 0$ results in:

$$\min_{\mathbf{X}\in\mathcal{D}_n} \|\mathbf{AX} - \mathbf{XB}\|_{\mathsf{F}}^2 + \lambda h(\mathbf{X}). \tag{2.2}$$

2.2. Nonconvex Sparsity Regularization Terms. Now we briefly introduce several nonconvex sparsity regularization terms proposed in literature. Huang [9] used the quartic term:

$$h(\mathbf{X}) = \|\mathbf{X} \odot (\mathbf{1} - \mathbf{X})\|_{\mathsf{F}}^2, \tag{2.3}$$

to construct the regularization problem, where $\odot$ is the Hadamard product and $\mathbf{1} \in \mathbb{R}^{n\times n}$ is the matrix of all ones. Jiang et al [10] proposed the L_p norm term by observing that Π_n can be equivalently characterized as $\Pi_n = \mathcal{D}_n \cap \{\mathbf{X} \mid \|\mathbf{X}\|_0 = n\}$, and hence use:

$$h(\mathbf{X}) = \|\mathbf{X} + \epsilon\mathbf{1}\|_p^p = \sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{X}_{ij} + \epsilon)^p, \tag{2.4}$$

with $0 < p < 1$ to continuously approximate the $\|X\|_0$. The aforementioned regularization all considered solving optimization over $\mathcal{D}_n$ with a concave regularization term. By a similar proof in Theorem 3.2 in [10], it is not hard to show for any strongly concave regularization term $h(\mathbf{X})$, it holds that $h(\mathbf{X}) = 0$, $\forall\ \mathbf{X} \in \Pi_n$ and $h(\mathbf{X}) > 0$, $\forall\ \mathbf{X} \in \mathcal{D}_n/\Pi_n$. Then with a finitely large λ, the problem (2.2) is equivalent to graph matching problem (1.3). For example, $h(\mathbf{X}) = \|(\log \mathbf{X}) \odot (\mathbf{X})\|_{\mathsf{F}}^2$ can yield the equivalence.

However, adding nonconvex regularization term $h(\mathbf{X})$ would make (2.2) nonconvex which may bring computational intractability when using iterative solvers.

3. Linear Reweighted Regularization Algorithmic Framework.

Considering the good performance of the linear reweighted regularization in recovering sparse solutions [5], we apply it to relax the graph matching problem.

3.1. Linear Regularization Term.

Linear regularization term is given by:

$$h_{\mathbf{W}}(\mathbf{X}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{W}_{ij}\mathbf{X}_{ij}, \tag{3.1}$$

where $\mathbf{W}_{ij}$ represents a weight for element $\mathbf{X}_{ij}$ for each (i, j).

3.2. Algorithm Description.

We present an iterative algorithm to solve the graph matching problem by using the linear reweighted regularization term. Specifically, the algorithm solves a sequence of graph matching problems, each added by a linear regularization term whose weight is determined based on the solution obtained in the previous iteration, subject to the doubly stochastic matrix constraint $\mathcal{D}_n$. With the reweighted regularization, the solution obtained at each subproblem is expected to approach a sparse solution. The detailed algorithmic framework is shown in Algorithm 3.1.

Algorithm 3.1 Linear Reweighted Regularization Algorithmic Framework

1: **Input**: $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times n}$, $\tau > 0$, $\epsilon_0 > 0$, and $\lambda_0 > 0$.
2: **Initialization**: $\mathbf{X}^{(0)} \in \mathbb{R}^{n\times n}$, set $k = 0$, $\mathbf{X}^{(0)} = \frac{1}{n}\mathbf{1}_n \in \mathcal{D}_n$.
(We denote $\mathbf{1}_n$ as $n \times n$ matrix of all ones)
3: **while** not convergent **do**
4: $\quad \mathbf{W}_{ij}^{(k)} = \frac{1}{\mathbf{X}_{ij}^{(k)} + \epsilon_k}$
5: $\quad \mathbf{X}^{(k+1)} = \arg\min_{\mathbf{X}\in\mathcal{D}_n} f(\mathbf{X}) + \lambda_k \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{W}_{ij}^{(k)}\mathbf{X}_{ij}$
6: $\quad$ Choose $\lambda_{k+1} \geq \lambda_k$, $\epsilon_{k+1} \leq \epsilon_k$
7: $\quad$ Let $k \leftarrow k + 1$
8: **end while**

3.3. Convergence.

We now justify the effectiveness o f l inear r eweighted regularization algorithm in solving the graph matching problem by showing its local convergence property. Specifically, T heorem 3 .2 s hows t hat w hen t he m atrix $\mathbf{X}^{(k)}$ s atisfies a sp ecific cri terion, the solution $\mathbf{X}^{(k+1)}$ obtained in the next iteration by reweighted algorithm will be the same as the globally optimal solution of the graph matching problem. To prove Theorem 3.2, we first establish the following lemma.

Lemma 3.1. *Let* $\mathbf{X}, \mathbf{Y} \in \mathcal{D}_n$. *Then*

$$\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_+ = -\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_- = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}|\mathbf{X}_{ij}-\mathbf{Y}_{ij}|,$$

where $[\mathbf{X}_{ij}]_+ = \max(\mathbf{X}_{ij}, 0)$ *and* $[\mathbf{X}_{ij}]_- = \min(\mathbf{X}_{ij}, 0)$.

Proof. Since $\mathbf{X}, \mathbf{Y} \in \mathcal{D}_n$, we have

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{X}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{Y}_{ij} = n.$$

Hence,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{X}_{ij}-\mathbf{Y}_{ij}) = 0,$$

and therefore by $\mathbf{X}_{ij}-\mathbf{Y}_{ij} = [\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_+ + [\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_-$, it follows that

$$\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_+ = -\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_-.$$

Moreover, it holds that $|\mathbf{X}_{ij}-\mathbf{Y}_{ij}| = [\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_+ - [\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_-$, and thus

$$\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_+ - \sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_- = \sum_{i=1}^{n}\sum_{j=1}^{n}|\mathbf{X}_{ij}-\mathbf{Y}_{ij}|.$$

Hence, we conclude that

$$\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_+ = -\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{X}_{ij}-\mathbf{Y}_{ij}]_- = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}|\mathbf{X}_{ij}-\mathbf{Y}_{ij}|.$$

Theorem 3.2. *Let* $f(\mathbf{X}) = \|\mathbf{AX}-\mathbf{XB}\|_{\mathsf{F}}^2$ *and* $\mathbf{X}^* = \arg\min_{\mathbf{X}\in\Pi_n} f(\mathbf{X})$. *If* $\|\mathbf{X}^{(k)}-\mathbf{X}^*\|_{\mathsf{F}} \le a$ *for some* $a \in [0, \frac{1}{2})$, *and* $\lambda \ge \frac{2(a+\epsilon)(1-a+\epsilon)}{(1-2a)}L$, *where* L *is the Lipschitz constant of* $f(\mathbf{X})$ *on* $\mathcal{D}_n$, *then* $\mathbf{X}^* = \arg\min_{\mathbf{X}\in\mathcal{D}_n} f(\mathbf{X}) + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij}$.

Proof. By the L-Lipschitz continuity of f, it holds $f(\mathbf{X}) \ge f(\mathbf{X}^*) - L\|\mathbf{X}^*-\mathbf{X}\|_{\mathsf{F}}$ for any $\mathbf{X} \in \mathcal{D}_n$. Thus we have

$$\begin{aligned}
&f(\mathbf{X}) + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij} \\
\ge& f(\mathbf{X}^*) - L\|\mathbf{X}^*-\mathbf{X}\|_{\mathsf{F}} + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij}^* + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)(\mathbf{X}_{ij}-\mathbf{X}_{ij}^*).
\end{aligned}$$

By $\mathbf{X}_{ij} - \mathbf{Y}_{ij} = [\mathbf{X}_{ij} - \mathbf{Y}_{ij}]_+ + [\mathbf{X}_{ij} - \mathbf{Y}_{ij}]_-$, it follows from the inequality above that

$$
\begin{aligned}
&f(\mathbf{X}) + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij} \\
\geq & f(\mathbf{X}^*) - L\|\mathbf{X}^* - \mathbf{X}\|_{\mathsf{F}} + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij}^* \\
&+ \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_+ + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_-.
\end{aligned} \tag{3.2}
$$

Notice that since $\mathbf{X}^*$ is a permutation matrix, $[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_+$ is positive only if $\mathbf{X}_{ij}^* = 0$. In this case, it must hold that $\mathbf{X}_{ij}^{(k)} \leq a$ by the condition $\|\mathbf{X}^{(k)} - \mathbf{X}^*\|_{\mathsf{F}} \leq a$ and thus $\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon} \geq \frac{1}{a+\epsilon}$. Similarly, $[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_-$ is negative only if $\mathbf{X}_{ij}^* = 1$. In this case, $\mathbf{X}_{ij}^{(k)} \geq 1 - a$, and $\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon} \leq \frac{1}{1-a+\epsilon}$. Hence,

$$
\begin{aligned}
&\lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_+ + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_- \\
\geq & \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{a+\epsilon}\right)[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_+ + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{1-a+\epsilon}\right)[\mathbf{X}_{ij} - \mathbf{X}_{ij}^*]_- \\
= & \frac{\lambda}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{a+\epsilon}\right)|\mathbf{X}_{ij} - \mathbf{X}_{ij}^*| - \frac{\lambda}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{1-a+\epsilon}\right)|\mathbf{X}_{ij} - \mathbf{X}_{ij}^*| \\
= & \frac{(1-2a)\lambda}{2(a+\epsilon)(1-a+\epsilon)}\sum_{i=1}^{n}\sum_{j=1}^{n}|\mathbf{X}_{ij} - \mathbf{X}_{ij}^*|,
\end{aligned}
$$

where the first equality follows from Lemma 3.1.

Plugging the above equation into (3.2) yields

$$
\begin{aligned}
&f(\mathbf{X}) + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij} \geq f(\mathbf{X}^*) - L\|\mathbf{X}^* - \mathbf{X}\|_{\mathsf{F}} \\
&\quad + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij}^* + \frac{(1-2a)\lambda}{2(a+\epsilon)(1-a+\epsilon)}\sum_{i=1}^{n}\sum_{j=1}^{n}|\mathbf{X}_{ij} - \mathbf{X}_{ij}^*| \\
\geq & f(\mathbf{X}^*) - L\|\mathbf{X}^* - \mathbf{X}\|_{\mathsf{F}} + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij}^* + \frac{(1-2a)\lambda}{2(a+\epsilon)(1-a+\epsilon)}\|\mathbf{X}^* - \mathbf{X}\|_{\mathsf{F}} \\
= & f(\mathbf{X}^*) + \left(\frac{(1-2a)\lambda}{2(a+\epsilon)(1-a+\epsilon)} - L\right)\|\mathbf{X}^* - \mathbf{X}\|_{\mathsf{F}} + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij}^* \\
\geq & f(\mathbf{X}^*) + \lambda \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}_{ij}^{(k)}+\epsilon}\right)\mathbf{X}_{ij}^*,
\end{aligned}
$$

■

where the second inequality is by $\sum_{i=1}^{n}\sum_{j=1}^{n}|\mathbf{X}_{ij}-\mathbf{X}^*_{ij}| \geq \|\mathbf{X}^*-\mathbf{X}\|_{\mathsf{F}}$ and the last inequality follows from the condition on λ. This completes the proof.

By Theorem 3.2, we can conclude that if there exists $k_N > 0$ such that $\|\mathbf{X}^{(k_N)}-\mathbf{X}^*\|_{\mathsf{F}} < \frac{1}{2}$ and $\lambda_{k_N} > \frac{2(a+\epsilon)(1-a+\epsilon)}{(1-2a)}L$, then $\mathbf{X}^{(k)}$ converges to $\mathbf{X}^*$. However, Theorem 3.2 does not directly guarantee global convergence from arbitrary initial points. It only provides sufficient conditions for local convergence. In practice, these conditions may not be satisfied at the beginning of the iteration. Nevertheless, as the results in Section 6.1 show, convergence can still occur in some cases even when the initial iterate lies outside the local neighborhood.

4. A Practical Reweighted Algorithm for Graph Matching Problem. Since $\mathbf{X}$ update in Algorithm 3.1 is a constraint optimization problem and cannot be explicitly expressed. We present a practical regularization algorithm in this section. Specifically, a sequence of relaxed problems in the form of (4.1) is solved by using the project gradient method:

$$\min_{\mathbf{X}\in\mathcal{D}_n} F_{\lambda,\mathbf{X}^{(0)},\epsilon}(\mathbf{X}) = \min_{\mathbf{X}\in\mathcal{D}_n} \|\mathbf{AX}-\mathbf{XB}\|_{\mathsf{F}}^2 + \lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{\mathbf{X}^{(0)}_{ij}+\epsilon}\right)\mathbf{X}_{ij}. \tag{4.1}$$

The main algorithm is given in Algorithm 4.1. The algorithm with projected gradient subsolver to update $\mathbf{X}_k$ is given in Algorithm 4.2. The Projection (onto $\mathcal{D}_n$) subsolver is given in Algorithm 4.3.

Algorithm 4.1 A Practical Reweighted Regularization Algorithm

1: **Input**: $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathbf{B}\in\mathbb{R}^{n\times n}$, $\epsilon_0>0$, and $\lambda_0>0$
2: **Initialization**: Set $k=0$, $\tau>0$, $\mathbf{X}_{-1}=\frac{1}{n}\mathbf{1_n}\in\mathcal{D}_n$
3: **while** not convergent **do**
4: Set $\mathbf{X}_k^{(0)}=\mathbf{X}_{k-1}$
5: Find an approximate minimizer $\mathbf{X}_k$ of problem (4.1) with $\lambda_k, \mathbf{X}_k^{(0)}, \epsilon_k$
6: $\lambda_{k+1}\geq\lambda_k$, $\epsilon_{k+1}\leq\epsilon_k$
7: $k\leftarrow k+1$
8: **end while**

4.1. Projected Gradient Method. To conduct a comparable numerical experiment using reweighted regularization, we adopt a similar projected gradient approach as introduced in [10], applying the projected gradient method to solve the minimization problem and update $\mathbf{X}_k$ in Algorithm 4.1. Specifically, at the k-th iteration, starting from the initial $X_k^{(0)}$, the projected gradient method for solving problem (4.1) with λ_k, $\mathbf{X}_k^{(0)}$, and ϵ_k proceeds as follows:

$$\mathbf{X}_k^{(i+1)} = \mathbf{X}_k^{(i)} + \delta^j\mathbf{D}^{(i)}, \quad \delta\in(0,1), \tag{4.2}$$

where the search direction:

$$\mathbf{D}^{(i)} = \mathcal{P}_{\mathcal{D}_n}\left(\mathbf{X}_k^{(i)} - \alpha_i\nabla F_{\lambda_k,\mathbf{X}_k^{(0)},\epsilon_k}(\mathbf{X}_k^{(i)})\right) - \mathbf{X}_k^{(i)}, \tag{4.3}$$

and $\mathcal{P}_{\mathcal{D}_n}(\cdot)$ is the projection onto $\mathcal{D}_n$. A fast dual gradient is used for computing projection onto $\mathcal{D}_n$. The parameter j is the smallest nonnegative integer satisfying the nonmonotone line search condition introduced in [19]:

$$F_{\lambda_k,\mathbf{X}_k^{(0)},\epsilon_k}(\mathbf{X}_k^{(i)}+\delta^j\mathbf{D}^{(i)}) \le C_i+\theta\delta^j\langle\nabla F_{\lambda_k,\mathbf{X}_k^{(0)},\epsilon_k}(\mathbf{X}_k^{(i)}),D^{(i)}\rangle,\quad \theta\in(0,1), \tag{4.4}$$

where the reference function value C_{i+1} is updated as the convex combination of C_i and $F_{\sigma_k,p,\epsilon_k}(X_k^{(i)})$:

$$C_{i+1}=\frac{\eta Q_i C_i+F_{\lambda_k,\mathbf{X}_k^{(0)},\epsilon_k}(\mathbf{X}_k^{(i)})}{Q_{i+1}}, \tag{4.5}$$

with $Q_{i+1}=\eta Q_i+1,\ \eta=0.85,\ C_0=F_{\lambda_k,\mathbf{X}_k^{(0)},\epsilon_k}(\mathbf{X}_k^{(0)}),\ Q_0=1.$

Algorithm 4.2 A Practical Algorithm with Project Gradient Method

Initialization: Set $k=0$, $\tau>0$, $\mathbf{X}_{-1}=\frac{1}{n}\mathbf{1_n}\in\mathcal{D}_n$, ϵ_0, $\lambda_0\ge 0$, $\theta,\delta,\gamma\in(0,1)$.
while $\|\mathbf{X}_{k-1}\|_0>n+\tau$ **do**
  Choose $\mathbf{X}_k^{(0)}=\mathbf{X}_{k-1}$. Set $i=0$, $\alpha_i>0$
  while $\|X_k^{(i)}-X_k^{(i-1)}\|_{\mathsf{F}}/\sqrt{n}>\tau$ **do**
    Compute $D^{(i)}=P_{\mathcal{D}_n}\big(X_k^{(i)}-\alpha_i\nabla F_{\lambda_k,\mathbf{X}_k^{(0)},\epsilon_k}(X_k^{(i)})\big)-X_k^{(i)}$.
    Find the smallest j such that δ^j satisfies (4.4)
    Set $X_k^{(i+1)}=X_k^{(i)}+\delta^j D_k^{(i)}$
    $i\leftarrow i+1$
  end while
  $\epsilon_{k+1}=\max(\delta\epsilon_k,\epsilon_{\min})$, $\lambda_{k+1}=\min(\lambda_k+\gamma,\lambda_{\max})$
  $k\leftarrow k+1$
end while

4.2. Fast Dual Gradient Algorithm for Computing the Projection onto $\mathcal{D}_n$.

In this subsection, we briefly summarize the fast dual gradient method introduced in [10] for computing the projection onto $\mathcal{D}_n$ (4.6).

$$\mathcal{P}_{\mathcal{D}_n}(\mathbf{C})=\underset{\mathbf{X}\in\mathbb{R}^{n\times n}}{\arg\min}\ \frac{1}{2}\|\mathbf{X}-\mathbf{C}\|_{\mathrm{F}}^2\quad\text{subject to}\quad \mathbf{Xe}=\mathbf{e},\ \mathbf{X}^\top\mathbf{e}=\mathbf{e},\ \mathbf{X}\ge 0, \tag{4.6}$$

The Lagrangian dual problem of (4.6) is

$$\max_{\mathbf{y},\mathbf{z}}\ \min_{\mathbf{X}\ge 0}\mathcal{L}(\mathbf{X},\mathbf{y},\mathbf{z}), \tag{4.7}$$

where

$$\mathcal{L}(\mathbf{X},\mathbf{y},\mathbf{z})=\frac{1}{2}\|\mathbf{X}-\mathbf{C}\|_{\mathrm{F}}^2-\langle\mathbf{y},\mathbf{Xe}-\mathbf{e}\rangle-\langle\mathbf{z},\mathbf{X}^\top\mathbf{e}-\mathbf{e}\rangle, \tag{4.8}$$

and $\mathbf{y}, \mathbf{z} \in \mathbb{R}^n$ are the Lagrange multipliers of the linear constraints $\mathbf{X}\mathbf{e} = \mathbf{e}$ and $\mathbf{X}^\top \mathbf{e} = \mathbf{e}$ respectively.

Let $\mathcal{P}_+(\cdot)$ denote the projection onto the nonnegative orthant. The dual problem (4.7) can then be equivalently rewritten as:

$$\min_{\mathbf{y},\mathbf{z}} \theta(\mathbf{y}, \mathbf{z}) := \frac{1}{2} \| \mathcal{P}_+ \left(\mathbf{C} + \mathbf{y}\mathbf{e}^\top + \mathbf{e}\mathbf{z}^\top \right) \|_{\mathrm{F}}^2 - \langle \mathbf{y} + \mathbf{z}, \mathbf{e} \rangle. \tag{4.9}$$

The derivative of $\theta(\mathbf{y}, \mathbf{z})$ can be written as:

$$\nabla \theta(\mathbf{y}, \mathbf{z}) = \begin{bmatrix} \mathcal{P}_+ \left(\mathbf{C} + \mathbf{y}\mathbf{e}^\top + \mathbf{e}\mathbf{z}^\top \right) \mathbf{e} - \mathbf{e} \\ \mathcal{P}_+ \left(\mathbf{C} + \mathbf{y}\mathbf{e}^\top + \mathbf{e}\mathbf{z}^\top \right)^\top \mathbf{e} - \mathbf{e} \end{bmatrix}. \tag{4.10}$$

The gradient method using the Barzilai-Borwein step sizes [2] to solve problem (4.9) is outlined in Algorithm 4.3. After obtaining the optimal solution $\mathbf{y}^*$ and $\mathbf{z}^*$ of (4.9), one can recover the projection of $\mathbf{C}$ by

$$\mathcal{P}_{\mathcal{D}_n}(\mathbf{C}) = \mathcal{P}_+ \left(\mathbf{C} + \mathbf{y}^* \mathbf{e}^\top + \mathbf{e}(\mathbf{z}^*)^\top \right). \tag{4.11}$$

Algorithm 4.3 Dual Barzilai–Borwein Method for Projection (Dual BB)

Input: Matrix $\mathbf{C}$, tolerance *tol*
Initialize: $\mathbf{y} = 0$, $\mathbf{z} = 0$, $\mathbf{e} = \mathbf{1}$ (all ones vector)
Set initial learning rate: $\alpha = 0.01$
Initialize previous gradients and iterates: $\nabla\theta_{\mathbf{y}0} = 0$, $\nabla\theta_{\mathbf{z}0} = 0$, $\mathbf{y}_0 = \mathbf{y}$, $\mathbf{z}_0 = \mathbf{z}$
while $\| \begin{bmatrix} \nabla\theta_{\mathbf{y}} \\ \nabla\theta_{\mathbf{z}} \end{bmatrix} \|_{\mathsf{F}} > tol$ **do**
  Calculate $\mathbf{M} = \mathbf{C} + \mathbf{y} \cdot \mathbf{e}^\top + \mathbf{e} \cdot \mathbf{z}^\top$
  Calculate projection: $\mathbf{M}_+ = \max(\mathbf{M}, 0)$
  Compute gradients: $\nabla\theta_{\mathbf{y}} = \mathbf{M}_+ \cdot \mathbf{e} - \mathbf{e}$, $\nabla\theta_{\mathbf{z}} = \mathbf{M}_+^\top \mathbf{e} - \mathbf{e}$
  Compute step sizes for $\mathbf{y}$ and $\mathbf{z}$: $\mathbf{s}_{\mathbf{y}} = \mathbf{y} - \mathbf{y}_0$, $\mathbf{s}_{\mathbf{z}} = \mathbf{z} - \mathbf{z}_0$
  Update previous iterates: $\mathbf{y}_0 = \mathbf{y}$, $\mathbf{z}_0 = \mathbf{z}$
  Compute gradient differences: $\delta_{\mathbf{y}} = \nabla\theta_{\mathbf{y}} - \nabla\theta_{\mathbf{y}0}$, $\delta_{\mathbf{z}} = \nabla\theta_{\mathbf{z}} - \nabla\theta_{\mathbf{z}0}$
  Compute BB step sizes: $\alpha_{\mathbf{y}} = \frac{\mathbf{s}_{\mathbf{y}}^\top \mathbf{s}_{\mathbf{y}}}{\mathbf{s}_{\mathbf{y}}^\top \delta_{\mathbf{y}}}$, $\alpha_{\mathbf{z}} = \frac{\mathbf{s}_{\mathbf{z}}^\top \mathbf{s}_{\mathbf{z}}}{\mathbf{s}_{\mathbf{z}}^\top \delta_{\mathbf{z}}}$
  Set $\alpha = 0.5 \cdot (\alpha_{\mathbf{y}} + \alpha_{\mathbf{z}})$
  Update iterates: $\mathbf{y} = \mathbf{y} - \alpha\nabla\theta_{\mathbf{y}}$, $\mathbf{z} = \mathbf{z} - \alpha\nabla\theta_{\mathbf{z}}$
  Store current gradients for next iteration: $\nabla\theta_{\mathbf{y}0} = \nabla\theta_{\mathbf{y}}$, $\nabla\theta_{\mathbf{z}0} = \nabla\theta_{\mathbf{z}}$
end while
Output: Optimal values $\mathbf{y}^* = \mathbf{y}$, $\mathbf{z}^* = \mathbf{z}$

5. Application.

5.1. Network Alignment Review. The quadratic programming formulation of a network alignment objective is given in [12]. Specifically, given two undirected graphs $\mathbf{A} = G(\mathbf{V}_A, \mathbf{E}_A)$ and $\mathbf{B} = G(\mathbf{V}_B, \mathbf{E}_B)$, with vertex sets $\mathbf{V}_A$ and $\mathbf{V}_B$ of size $|\mathbf{V}_A|$ and $|\mathbf{V}_B|$, the goal of network alignment is to find a matching $\mathcal{M}$ between the vertices using a prior knowledge matrix $\mathbf{L}$ that encodes the likelihood of vertex alignments. A binary matrix $\mathbf{X}$ is introduced to represent the matching, where $\mathbf{X}_{ij} = 1$ indicates a match between vertex i in $\mathbf{A}$ and vertex j in $\mathbf{B}$. Then, the corresponding quadratic program can be formulated as

$$\max_{\mathbf{X}} \alpha \mathbf{L} \cdot \mathbf{X} + \beta \mathbf{A} \cdot \mathbf{X}\mathbf{B}\mathbf{X}^\top, \tag{5.1}$$

subject to the constraints,

$$\sum_{i=1}^{|\mathbf{V}_A|} \mathbf{X}_{ij} \leq 1, \quad \forall j = 1, \ldots, |\mathbf{V}_B|, \ \mathbf{X}_{ij} \in \{0, 1\}, \tag{5.2}$$

$$\sum_{j=1}^{|\mathbf{V}_B|} \mathbf{X}_{ij} \leq 1, \quad \forall i = 1, \ldots, |\mathbf{V}_A|, \ \mathbf{X}_{ij} \in \{0, 1\}, \tag{5.3}$$

where $\mathbf{A} \cdot \mathbf{X}\mathbf{B}\mathbf{X}^T = \sum_{i=1}^{|\mathbf{V}_A|} \sum_{j=1}^{|\mathbf{V}_B|} \mathbf{A}_{ij} (\mathbf{X}\mathbf{B}\mathbf{X}^\top)_{ij}$ is called overlap of matching $\mathcal{M}$. Here, α and β are non-negative constants that allow for tradeoff between matching weights from the prior and the number of overlapping edges.

In our numerical experiments, we consider the matrix $\mathbf{A}$ and $\mathbf{B}$ as adjacency or distance matrices for undirected and weighted graph with $|\mathbf{V}_A| = |\mathbf{V}_B| = n$, $\mathbf{X}_{ij} \in \Pi_n$ and no prior knowledge is given (i.e. $\mathbf{L} = 0$). Notice that when $\mathbf{X} \in \Pi_n$,

$$\|\mathbf{A}\mathbf{X} - \mathbf{X}\mathbf{B}\|_{\mathsf{F}}^2 = -2\, \mathbf{A} \cdot \mathbf{X}\mathbf{B}\mathbf{X}^T + \|\mathbf{A}\|_{\mathsf{F}}^2 + \|\mathbf{B}\|_{\mathsf{F}}^2. \tag{5.4}$$

Hence, problem (1.3) is equivalent to (5.1) when $\mathbf{L} = 0$, $|\mathbf{V}_A| = |\mathbf{V}_B| = n$.

5.2. Graph in Social Network Alignment Problem. The social network alignment problem aims to identify individuals in two different graphs who share similar connection patterns. Given two social network graphs (e.g. Figure 1), this task can be completed by solving (5.1), where the matrices $\mathbf{A}$ and $\mathbf{B}$ represent the distance matrices for each social network graph. Specifically, each element in the distance matrix corresponds to the shortest distance between a pair of nodes in the graph. The distance matrix for a graph can be computed using the Breadth-First Search (BFS) algorithm [4]. This algorithm operates by growing a tree from a given node, expanding outward while incrementing the hop count at each expansion step and nodes that have already been visited are ignored.

5.3. Graph in Shape correspondence problem. Given two manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$ sampled by point clouds $P_1 = \{x_i\}_{i=1}^n$ and $P_2 = \{y_i\}_{i=1}^n$, the task of dense shape correspondence is to find a point-to-point map between P_1 and P_2. The task can be achieved by solving graph matching problem (1.3) where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$ are two pairwise descriptors, such as geodesic distances, between points in P_1 and P_2 (e.g. Figure 2).

From a more practical view, the surface $\mathcal{M}$ can be discretized using a triangular mesh $T = \{\tau_l\}_{i=1}^n$ with edges $E = \{e_{ij}\}$, and vertices of the mesh are denoted by $V = \{\mathbf{x}_i\}_{i=1}^n$. For each edge e_{ij} connecting vertices p_i and p_j, the angles opposite to the edge are defined as α_{ij} and β_{ij}. The stiffness matrix $\mathbf{A}_s$ is given by [13, 14]:

$$\mathbf{A}_{s_{ij}} = \begin{cases} -2\left(\cot\alpha_{ij} + \cot\beta_{ij}\right) & \text{if } i \sim j \\ \sum_{k\sim i} \mathbf{A}_s(i,j) & \text{if } i = j, \end{cases} \tag{5.5}$$

where $i \sim j$ indicates that i and j are connected by an edge. For more information on sparse pairwise descriptors, we refer the readers to [18].

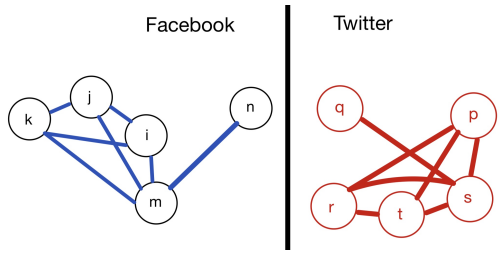

Figure 1. *An illustration of the graph matching problem in social network alignment. The two graphs represent connections between individuals on different platforms. Facebook (left) and Twitter (right). The goal is to identify corresponding nodes (e.g. individuals) between the two graphs based on their structural connectivity.*

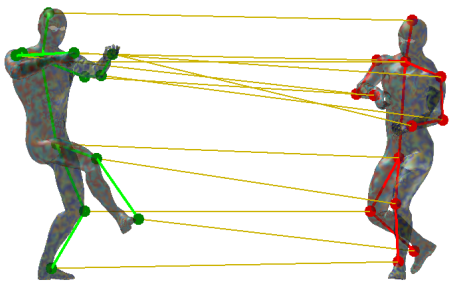

Figure 2. *The body matching result visualization. Matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *represent the pairwise geodesic distances between joints in two individuals. The visualization is obtained by solving relaxed graph matching problem with linear reweighted regularization term that we proposed.*

6. Experimental results.

6.1. Experimental Dataset. In our experiments, matrix $\mathbf{A}$ in (1.3) is obtained by first creating n random 2D points using `rand(n, 2)*10` in MATLAB and then each entry $\mathbf{A}_{ij}$ is computed as the Euclidean distance between points i and j. Matrix $\mathbf{B} = \mathbf{P} * \mathbf{A} * \mathbf{P}' + \mathbf{C}$, where $\mathbf{P}$ is randomly generated permutation matrix and $\mathbf{C}$ is generated by 2D points using `rand(n, 2)*0.5`, and then each entry $\mathbf{C}_{ij}$ is computed as the Euclidean distance between points i and j.

6.2. Implementation Details. To compare our proposed method with L_p norm regularization, we implemented L_p-regularization-based solvers for the graph matching problem using $L_{p=0.75}$ and $L_{p=0.5}$ regularization terms as $h(\mathbf{X})$, following Algorithm 2 in [10]. We applied the linear reweighted regularization term using Algorithm 4.2, where the projection is computed using Algorithm 4.3. The initial point for all regularization algorithms is chosen as $\mathbf{X}_0 = \frac{1}{n}\mathbf{1}_{\boldsymbol{n}}$, with the initial $\epsilon_0 = 1$, safeguards $\epsilon_{\min} = 10^{-3}$, and $\lambda_{\max} = 10^6$. The shrinkage parameter for updating λ_k is set to $\gamma = 0.9$. We define $r(\mathbf{X}) = \|\mathbf{AX} - \mathbf{XB}\|_{\mathsf{F}}^2$, and denote the best numerical solution obtained at real time t as $\mathbf{X}_k$. Since only a relatively small perturbation $\mathbf{C}$ is introduced into the permuted matrix $\mathbf{B}$ in Section 6.1, we assume $\mathbf{X}^* = \arg\min_{\mathbf{X}\in\Pi_n} r(\mathbf{X}) = \mathbf{P}$,

where $\mathbf{P}$ is the permutation matrix generated in Section 6.1. The objective error at real time t is defined as $|r(\mathbf{X}_k) - r(\mathbf{P})|$, and the residual at real time t is given by $\|\mathbf{X}_k - \mathbf{P}\|_{\mathsf{F}}$. The experimental results using different regularization terms are presented in Figures 3 and 4.

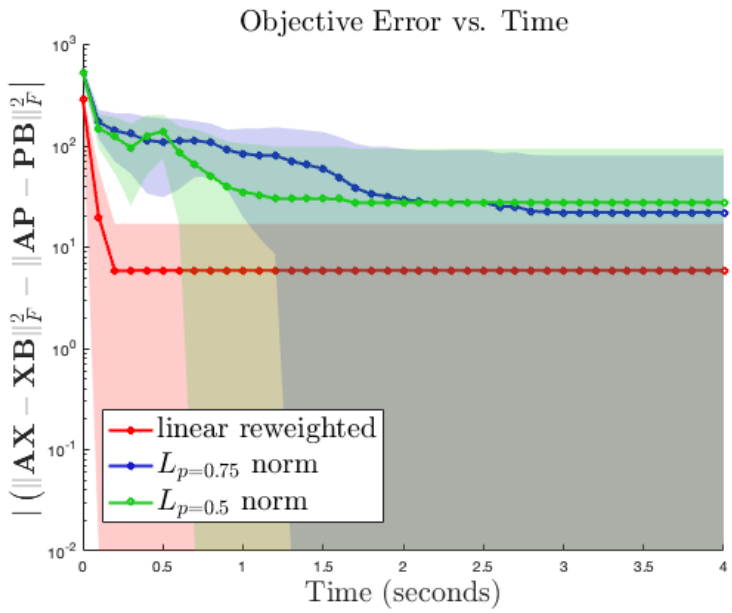

Figure 3. *Average and standard deviation results of Objective Error by linear reweighted, $L_{p=0.75}$ and $L_{p=0.5}$ regularization term, on solving 50 independently random generated instances of graph matching problem of dimension $n = 50$.*

7. Discussion. In this work, we established the theory behind the linear reweighted regularization term. We provided a comprehensive justification for the use of linear reweighted regularization in solving the relaxed graph matching problem. Additionally, we conducted numerical experiments to empirically demonstrate that our reweighted regularization framework outperforms other regularization terms. While our current experiments are based on synthetic data, it is important to discuss potential performance on large-scale real-world datasets. In particular, our method is expected to be effective for large-scale datasets that are not sparse.

In our future work, we plan to extend the graph matching problem to more applicable topics and focus on large-scale real-world data. We also aim to propose an Augmented Lagrange Multiplier (ALM)-based algorithm to ensure the effectiveness of our reweighted regularization term when facing large-scale data.

Acknowledgments. The first author would like to thank Ms. Yueshan Ai for validating the proof of Theorem 3.2. The author also thanks one anonymous reviewer for their valuable comments and suggestions.

Conflict of interest declaration. The author(s) has/have no competing interests to declare.

Data Availability. The codes and source data files are available on GitHub at

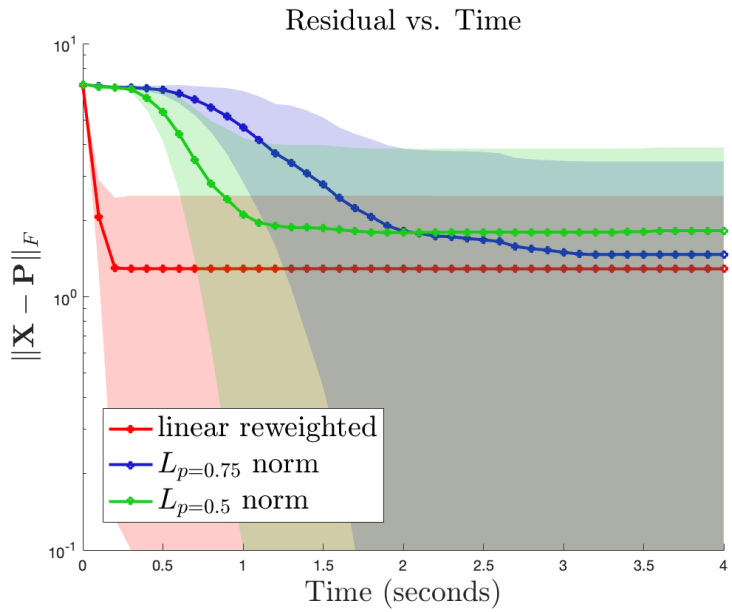

Figure 4. *Average and standard deviation results of Residual by linear reweighted, $L_{p=0.75}$ and $L_{p=0.5}$ regularization term, on solving 50 independently random generated instances of graph matching problem of dimension $n = 50$.*

https://github.com/rongxuan-li/graph-match

REFERENCES

[1] R. K. Ahuja, J. B. Orlin, and A. Tiwari, *A greedy genetic algorithm for the quadratic assignment problem*, Computers & Operations Research, 27 (2000), pp. 917–934.

[2] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA journal of numerical analysis, 8 (1988), pp. 141–148.

[3] G. Birkhoff, *Tres observaciones sobre el algebra lineal*, Univ. Nac. Tucuman, Ser. A, 5 (1946), pp. 147–154.

[4] A. Bundy and L. Wallen, *Breadth-first search*, Catalogue of artificial intelligence tools, (1984), pp. 13–13.

[5] E. J. Candes, M. B. Wakin, and S. P. Boyd, *Enhancing sparsity by reweighted l1 minimization*, Journal of Fourier analysis and applications, 14 (2008), pp. 877–905.

[6] Z. Drezner, *The quadratic assignment problem*, Location science, (2015), pp. 345–363.

[7] P. Foggia, G. Percannella, and M. Vento, *Graph matching and learning in pattern recognition in the last 10 years*, International Journal of Pattern Recognition and Artificial Intelligence, 28 (2014), p. 1450001.

[8] A. M. Frieze, J. Yadegar, S. El-Horbaty, and D. Parkinson, *Algorithms for assignment problems on an array processor*, Parallel computing, 11 (1989), pp. 151–162.

[9] T. Huang, *Continuous optimization methods for the quadratic assignment problem*, PhD thesis, The University of North Carolina at Chapel Hill, 2008.

[10] B. Jiang, Y.-F. Liu, and Z. Wen, *Lp-norm regularization algorithms for optimization over permutation matrices*, SIAM Journal on Optimization, 26 (2016), pp. 2284–2313.

[11] Z. Lu, *Iterative reweighted minimization methods for lp regularized unconstrained nonlinear programming*, Mathematical Programming, 147 (2014), pp. 277–307.

[12] V. Ravindra, H. Nassar, D. F. Gleich, and A. Y. Grama, *Rigid graph alignment*, in International Workshop on Complex Networks & Their Applications, 2019.
[13] J. Reddy, *An Introduction to the Finite Element Method*, McGraw-Hill, 1993.
[14] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo, *Discrete laplace–beltrami operators for shape analysis and segmentation*, Computers & Graphics, 33 (2009), pp. 381–390.
[15] S. Sahni and T. Gonzalez, *P-complete approximation problems*, Journal of the ACM (JACM), 23 (1976), pp. 555–565.
[16] É. Taillard, *Robust taboo search for the quadratic assignment problem*, Parallel computing, 17 (1991), pp. 443–455.
[17] Y. Xia, *An efficient continuation method for quadratic assignment problems*, Computers & Operations Research, 37 (2010), pp. 1027–1032.
[18] R. Xiang, R. Lai, and H. Zhao, *Efficient and robust shape correspondence via sparsity-enforced quadratic assignment*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9513–9522.
[19] H. Zhang and W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM journal on Optimization, 14 (2004), pp. 1043–1056.