L_p-NORM REGULARIZATION ALGORITHMS FOR OPTIMIZATION OVER PERMUTATION MATRICES

BO JIANG[*], YA-FENG LIU[†], AND ZAIWEN WEN[‡]

Abstract. Optimization problems over permutation matrices appear widely in facility layout, chip design, scheduling, pattern recognition, computer vision, graph matching, etc. Since this problem is NP-hard due to the combinatorial nature of permutation matrices, we relax the variable to be the more tractable doubly stochastic matrices and add an L_p-norm ($0 < p < 1$) regularization term to the objective function. The optimal solutions of the L_p-regularized problem are the same as the original problem if the regularization parameter is sufficiently large. A lower bound estimation of the nonzero entries of the stationary points and some connections between the local minimizers and the permutation matrices are further established. Then we propose an L_p regularization algorithm with local refinements. The algorithm approximately solves a sequence of L_p regularization subproblems by the projected gradient method using a nonmontone line search with the Barzilai-Borwein step sizes. Its performance can be further improved if it is combined with certain local search methods, the cutting plane techniques as well as a new negative proximal point scheme. Extensive numerical results on QAPLIB and the bandwidth minimization problem show that our proposed algorithms can often find reasonably high quality solutions within a competitive amount of time.

Key words. permutation matrix, doubly stochastic matrix, quadratic assignment problem, L_p regularization, cutting plane, negative proximal point, Barzilai-Borwein method

AMS subject classifications. 65K05, 90C11, 90C26, 90C30

1. Introduction. In this paper, we consider optimization over permutation matrices:

$$(1.1) \qquad \min_{X \in \Pi_n} \ f(X),$$

where $f(X) \colon \mathbb{R}^{n \times n} \to \mathbb{R}^n$ is differentiable and Π_n is the set of n-order permutation matrices, namely,

$$\Pi_n = \{X \in \mathbb{R}^{n \times n} \mid X\mathbf{e} = X^\mathsf{T}\mathbf{e} = \mathbf{e}, X_{ij} \in \{0,1\}\},$$

in which $\mathbf{e} \in \mathbb{R}^n$ is a vector of all ones. Given two groups of correlative objects associated with the rows and columns of a square matrix, respectively, each permutation matrix implies an assignment from objects in one group to objects in the other group. Thus problem (1.1) is also referred to as the nonlinear assignment problem since it looks for the best assignment with the smallest nonlinear cost among the two groups.

One of the most famous special cases of problem (1.1) is the quadratic assignment problem (QAP) [33], one of the hardest combinatorial optimization problems:

$$(1.2) \qquad \min_{X \in \Pi_n} \ \mathrm{tr}(A^\mathsf{T} X B X^\mathsf{T}),$$

where $A, B \in \mathbb{R}^{n \times n}$. The QAP has wide applications in statistics, facility layout, chip design, keyboards design, scheduling, manufacturing, etc. For more details, one can refer to [11, 21] and references therein. Some generalizations of QAP are also investigated, such as the cubic, quartic, and generally N-adic assignment problems, see [35]. Burkard *et al.* [48] discussed one application of biquadratic assignment, a special quartic assignment, in very-large-scale integrated circuit design. For other applications of general nonlinear assignment, one can refer to [49].

It is generally not practical to solve problem (1.1) exactly due to its strong NP-hardness. The goal of this paper is to develop fast algorithms which can find high quality permutation matrices for problem (1.1), especially when the problem dimension is large.

1.1. Related works. Although most of the existing methods are mainly tailored for QAP (1.2), many of them can be easily extended to the general problem (1.1). Thus in this subsection, we simply review some related works on QAP. Since in this paper we are interested in methods which can quickly find high quality solutions, we shall not introduce the lower bound methods [32, 56] and the exact methods [1, 23] in details. For these algorithms, we refer the readers to the review papers [3, 11, 12, 21, 24, 40]. We next briefly introduce a few approximation methods, which often return good approximate solutions (which are permutation matrices) for QAP (1.2) in a reasonable time.

The philosophy of the vertex based methods is similar to that of the simplex methods for linear programming. Specifically, it updates the iterates from one permutation matrix to another. Different strategies of updating permutation matrices lead to different methods, including local search methods [26, 45], greedy randomized adaptive search procedures [22, 47], tabu search methods [6, 50], genetic algorithms [2, 52]. For a comprehensive review on the vertex based methods, one can refer to [11] and references therein.

The interior-point based methods generates a permutation matrix along some "central" path (of interior points) of the set of doubly stochastic matrices, also known as the Birkhoff polytope:

$$\mathcal{D}_n = \{X \in \mathbb{R}^{n \times n} \mid X\mathbf{e} = X^\mathsf{T}\mathbf{e} = \mathbf{e}, X \geq \mathbf{0}\},$$

where the symbol $\geq$ denotes the componentwise ordering and $\mathbf{0} \in \mathbb{R}^{n \times n}$ is the all-zero matrix. By the Birkhoff-von Neumann theorem [10], 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. Similar to the interior-point methods for linear programming, these methods construct a path of interior points by solving a sequence of regularization problems over $\mathcal{D}_n$.

Xia [55] proposed a Lagrangian smoothing algorithm for QAP (1.2) which solves a sequence of L_2 regularization subproblem

$$(1.3) \qquad \min_{X \in \mathcal{D}_n} \operatorname{tr}(A^\mathsf{T} X B X^\mathsf{T}) + \mu_k \|X\|_\mathsf{F}^2$$

with dynamically decreasing parameters $\{\mu_k \mid k = 0, 1, 2, \ldots\}$. The initial value μ_0 in subproblem (1.3) was chosen such that the problem is convex and the subproblem with fixed μ_k was approximately solved by the Frank-Wolfe algorithm up to fixed number of iterations (it was called the truncated Frank-Wolfe method in [55]). Note that Bazaraa and Sherali [7] first considered regularization problem (1.3) where μ_k is chosen such that the problem is strongly concave, but they only solved (1.3) once. In addition,

Huang [30] used the quartic term $\|X \odot (\mathbf{1} - X)\|_{\mathsf{F}}^2$ 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.

One special case of QAP (1.2) (and thus problem (1.1)) is the graph matching problem:

$$(1.4) \qquad \min_{X \in \Pi_n} \ \|AX - XB\|_{\mathsf{F}}^2,$$

which has wide applications in pattern recognition, computer vision, etc [17]. Note that the objective function in (1.4) is always convex in X and can be rewritten as $-2\mathrm{tr}(A^\mathsf{T} XBX^\mathsf{T}) + \|A\|_{\mathsf{F}}^2 + \|B\|_{\mathsf{F}}^2$, which is no longer convex in X. By constructing a convex and a concave quadratic optimization problem over doubly stochastic matrices, [39, 57] considered path following algorithms which solve a sequence of convex combinations of the convex and concave optimization problems, wherein the combination parameters were chosen such that the resulting problems change from convex to concave. Moreover, [42] observed that initialization with a convex problem can improve the practical performance of the regularization algorithms.

Recently, Fogel $et\ al.$ [25] used QAP to solve the seriation and 2-SUM problems. QAP considered therein takes the following special structure:

$$(1.5) \qquad \min_{X \in \Pi_n} \ \pi_I X^\mathsf{T} L_A X \pi_I,$$

where A is binary symmetric, $\pi_I = (1, 2, \ldots, n)^\mathsf{T}$, and $L_A = \mathrm{Diag}(A\mathbf{e}) - A$. Here, $\mathrm{Diag}(A\mathbf{e})$ represents a diagonal matrix whose diagonal elements are $A\mathbf{e}$. They used the regularization term $\|PX\|_{\mathsf{F}}^2$ with $P = I_n - \frac{1}{n}\mathbf{1}$, where I_n denotes the identity matrix of size n, to construct a convex relaxation problem over $\mathcal{D}_n$ for (1.5). Using a recent result of Goemans [29], Lim and Wright [36] constructed a new convex relaxation problem over a convex hull of the permutation vectors for (1.5). In their problem, the number of variables and constraints reduces to $O(n \log n)$ in theory and $O(n \log^2 n)$ in practice. The advantage of this new formulation was illustrated by numerical experiments. However, as pointed in [36], it is still unclear how to extend their new formulation to the more general QAP (1.2).

The aforementioned regularization or relaxation methods all considered solving optimization over $\mathcal{D}_n$. By using the fact that $\Pi_n = \{X \mid X^\mathsf{T} X = I_n, X \geq 0\}$, Wen and Yin [54] reformulated QAP (1.2) as

$$\min_{X \in \mathbb{R}^{n \times n}} \ \mathrm{tr}(A^\mathsf{T} (X \odot X) B (X \odot X)^\mathsf{T}) \quad \text{s.t.} \quad X^\mathsf{T} X = I_n, \ X \geq 0$$

and proposed an efficient algorithm by applying the augmented Lagrangian method to handle the constraint $X \geq 0$. In their algorithm, a sequence of augmented Lagrangian subproblems with orthogonality constraint is solved inexactly.

1.2. Our contribution. In this paper, we consider optimization problem (1.1) over permutation matrices and propose an L_p-norm regularization model with $0 < p < 1$. The L_p-norm regularization is exact in the sense that its optimal solutions are the same as the original problem if the regularization parameter is sufficiently large. We derive a lower bound estimate for nonzero entries of the stationary points and establish connections between the local minimizers and the permutation matrices. Then we propose an algorithm with local refinements. Basically, it solves approximately a sequence of L_p regularization subproblems by the projected gradient method using nonmontone line search with the Barzilai-Borwein (BB) step sizes. The projection of a

matrix onto the set of doubly stochastic matrices is solved by a dual gradient method which is often able to find a highly accurate solution faster than the commercial solver MOSEK. The performance of our algorithm can be further improved if it is combined with certain local search methods, cutting plane techniques as well as a new negative proximal point scheme. In fact, we show that the global solution of the regularized model with a negative proximal point term is also the optimal solution of problem (1.1). In particular, it is the farthest point in the solution set of (1.1) to the given point in the proximal term. Therefore, this technique enables us to get rid of a large portion of non-optimal permutation matrices. Numerical results on QAPLIB show that the proposed algorithm can quickly find high quality approximate solutions. For example, our proposed algorithm can find a solution of the largest problem instance "tai256c" in QAPLIB with a gap 0.2610% to the best known solution in less than a half minute on an ordinary PC.

1.3. Notation. For any $X \in \mathbb{R}^{n \times n}$, $\|X\|_0$ denotes the number of nonzero entries of X and its L_p-norm is $\|X\|_p^p = \sum_{i=1}^n \sum_{j=1}^n |X_{ij}|^p$ for $0 < p < 1$. The matrix $X^p \in \mathbb{R}^n$ is defined by $(X^p)_{ij} = X_{ij}^p$ for each $(i,j) \in \mathcal{N} \times \mathcal{N}$ with $\mathcal{N} := \{1, \ldots, n\}$. The inner product of $M, N \in \mathbb{R}^{m \times n}$ is defined as $\langle M, N \rangle = \operatorname{tr}(M^\mathsf{T} N)$, where $\operatorname{tr}(\cdot)$ is the trace operator. The Kronecker product of any two matrices M and N is denoted as $M \otimes N$. We define $\mathcal{D}_{[0,1]^n} = \{X \in \mathbb{R}^{n \times n} \mid \mathbf{0} \leq X \leq \mathbf{1}\}$, where the symbol $\leq$ denotes the componentwise ordering. For any $X \in \mathcal{D}_n$, let the support set of X be $\Lambda(X) = \{(i,j) \in \mathcal{N} \times \mathcal{N} \mid X_{ij} > 0\}$. Denote $\Lambda(X_{i \cdot}) = \{j \in \mathcal{N} \mid (i,j) \in \Lambda(X)\}$ and $\Lambda(X_{\cdot j}) = \{i \in \mathcal{N} \mid (i,j) \in \Lambda(X)\}$. For any $X \in \Pi_n$, we define its corresponding permutation vector as $\pi \in \mathbb{R}^n$ with the j-th element being $\pi_j = i$, where $i \in \mathcal{N}$ satisfies $X_{ij} = 1$. We denote the 2-neighborhood of X as $\mathcal{N}_2(X) = \{Z \in \Pi_n : \|Z - X\|_\mathsf{F}^2 \leq 4\} = \{Z \in \Pi_n \mid \langle Z, X \rangle \geq n - 2\}$ and $\mathcal{N}_2^{\text{best}}(X)$ is the set of permutation matrices which take the least function value, in the sense of $f(\cdot)$, among all permutation matrices in $\mathcal{N}_2(X)$. We say that X is locally 2-optimal if $X \in \mathcal{N}_2^{\text{best}}(X)$.

1.4. Organization. The rest of this paper is organized as follows. Some preliminaries are provided in Section 2. The L_0 regularization problem is introduced in Section 3.1 and the extension to the general L_p regularization problem is presented in Section 3.2. Our L_p regularization algorithmic framework is given in Section 4.1 and a practical L_p regularization algorithm is developed in Section 4.2. A fast dual BB method for computing the projection onto the set of doubly stochastic matrices is proposed in Section 4.3. We combine the L_p regularization algorithm with the cutting plane technique in Section 5.1 and propose a negative proximal point technique in Section 5.2. Some implementation details, including an efficient way of doing local 2-neighborhood search, are given in Section 6.1, while numerical results on QAPLIB are reported in Sections 6.2 – 6.4. We apply our proposed algorithms to solve a class of real-world problems, namely, the bandwidth minimization problem in Section 7. Finally, we make some concluding remarks in Section 8.

2. Preliminaries. Assume that $\nabla f(X)$ is Lipschitz continuous on $\mathcal{D}_{[0,1]^n}$ with Lipschitz constant L, that is,

$$(2.1) \qquad \|\nabla f(X) - \nabla f(Y)\|_\mathsf{F} \leq L \|X - Y\|_\mathsf{F}, \quad \forall\, X, Y \in \mathcal{D}_{[0,1]^n}.$$

Moreover, we assume

$$(2.2) \quad \frac{\nu_f}{2} \|Y - X\|_\mathsf{F}^2 \leq f(Y) - f(X) - \langle \nabla f(X), Y - X \rangle \leq \frac{\overline{\nu}_f}{2} \|Y - X\|_\mathsf{F}^2, \quad \forall\, X, Y \in \mathcal{D}_n.$$

Note that when $\bar{\nu}_f < 0$, (2.2) implies that $f(\cdot)$ is strongly concave. By some easy calculations, (2.2) indicates that

$$(2.3) \qquad f(tX + (1-t)Y) \geq tf(X) + (1-t)f(Y) - \frac{\bar{\nu}_f}{2}t(1-t)\|X - Y\|_{\mathsf{F}}^2$$

holds for all $X, Y \in \mathcal{D}_n$ and $t \in [0, 1]$. Notice that assumptions (2.1) and (2.2) are mild. It can be easily verified that the objective in QAP (1.2) satisfies the two assumptions with $L = \max\{|\bar{\nu}_f|, |\underline{\nu}_f|\}$, where $\bar{\nu}_f$ and $\underline{\nu}_f$ are taken as the largest and smallest eigenvalues of $B^{\mathsf{T}} \otimes A^{\mathsf{T}} + B \otimes A \in \mathbb{R}^{n^2 \times n^2}$, respectively.

Consider the function $h(x) = (x + \epsilon)^p$ over $x \in [0, 1]$ with $0 < p < 1$ and $\epsilon \geq 0$. It is straightforward to show that $h(\cdot)$ is strongly concave with the parameter

$$(2.4) \qquad \bar{\nu}_h := p(1 - p)(1 + \epsilon)^{p-2}$$

on the interval [0,1]. Moreover, the inequality $h(x) - h(y) > \langle h'(x), x - y \rangle$ holds for any $x \neq y \in [0, 1]$ and $x + \epsilon > 0$. Consider the function

$$(2.5) \qquad h(X) := \|X + \epsilon\mathbf{1}\|_p^p = \sum_{i=1}^{n}\sum_{j=1}^{n}(X_{ij} + \epsilon)^p.$$

Using the strong concavity of $h(x)$ and the separable structure of $h(X)$, we can establish the following result.

PROPOSITION 2.1. *The function $h(X)$ with $0 < p < 1$ and $\epsilon \geq 0$ is strongly concave with parameter $\bar{\nu}_h$ on $\mathcal{D}_{[0,1]^n}$, namely,*

$$(2.6) \qquad h(tX + (1-t)Y) \geq th(X) + (1-t)h(Y) + \frac{\bar{\nu}_h}{2}t(1-t)\|Y - X\|_{\mathsf{F}}^2$$

holds for any $X, Y \in \mathcal{D}_{[0,1]^n}$.

3. L_p-norm regularization model for (1.1). In this section, we first introduce the L_0-norm regularization model for (1.1) in Section 3.1. Then, we focus on the L_p-norm regularization model for (1.1) in Section 3.2.

3.1. L_0-norm regularization. Observe that Π_n can be equivalently characterized as

$$(3.1) \qquad \Pi_n = \mathcal{D}_n \cap \{X \mid \|X\|_0 = n\}.$$

Thus problem (1.1) is equivalent to the problem of minimizing $f(X)$ over $\mathcal{D}_n$ intersecting with the sparse constraint $\|X\|_0 = n$. This fact motivates us to consider the L_0 regularization model for (1.1) as follows:

$$(3.2) \qquad \min_{X \in \mathcal{D}_n} f(X) + \sigma\|X\|_0,$$

where $\sigma > 0$. Let X^* be one global solution of (1.1). Denote

$$(3.3) \qquad f^* := f(X^*) \quad \text{and} \quad \underline{f} := \min_{X \in \mathcal{D}_n} f(X).$$

We immediately have $f^* \geq \underline{f}$ and obtain the following lemma.

LEMMA 3.1. *Suppose $\sigma > \frac{1}{2}(f^* - \underline{f})$, where f^* and $\underline{f}$ are given in (3.3). Then any global solution $X(\sigma)$ of (3.2) is also the global solution of (1.1).*

Proof. First, we use the contradiction argument to show that $X(\sigma)$ is a permutation matrix. Suppose that $X(\sigma)$ is not a permutation matrix, then $\|X(\sigma)\|_0 \geq n+2$. From the optimality of $X(\sigma)$, we know that

$$(3.4) \qquad\qquad f^* + \sigma\|X^*\|_0 \geq f(X(\sigma)) + \sigma\|X(\sigma)\|_0.$$

Moreover, by the definition of $\underline{f}$, we have $f(X(\sigma)) + \sigma\|X(\sigma)\|_0 \geq \underline{f} + \sigma\|X(\sigma)\|_0$. This inequality, together with (3.4) and the fact $\|X(\sigma)\|_0 \geq n+2$, implies

$$\sigma \leq (f^* - \underline{f})/(\|X(\sigma)\|_0 - \|X^*\|_0) \leq (f^* - \underline{f})/2,$$

which is a contradiction to $\sigma > \frac{1}{2}(f^* - \underline{f})$.

Using $\|X(\sigma)\|_0 = \|X^*\|_0 = n$ and (3.4), we have $f^* \geq f(X(\sigma))$, which means that $X(\sigma)$ is the global solution of (1.2). $\square$

Lemma 3.1 shows that L_0 regularization (3.2) is exact in the sense that L_0 regularization problem (3.2) shares the same global solution with problem (1.1). However, $\|X\|_0$ is not continuous in X, which makes the L_0-norm regularization problem (3.2) hard to be solved.

3.2. L_p-norm regularization. The L_1-norm regularization $\|X\|_1$ has been shown in [14, 20] to be a good approximation of $\|X\|_0$ under some mild conditions. However, for any $X \in \mathcal{D}_n$, $\|X\|_1$ is always equal to the constant n. This fact implies that the L_1-norm regularization does not work for problem (1.1). Considering the good performance of the L_p-norm regularization in recovering sparse solutions [15, 31, 37], we aim to investigate this technique to handle the hard term $\|X\|_0$. Given any $\epsilon \geq 0$ and $0 < p < 1$, we use $h(X) = \|X + \epsilon\mathbf{1}\|_p^p$ to approximate $\|X\|_0$ in (3.2) and obtain the corresponding L_p-norm regularization model:

$$(3.5) \qquad\qquad \min_{X \in \mathcal{D}_n} \ F_{\sigma,p,\epsilon}(X) := f(X) + \sigma\|X + \epsilon\mathbf{1}\|_p^p.$$

Note that the problem of minimizing $h(X)$ over $\mathcal{D}_n$ and the problem of minimizing $\|X\|_0$ over $\mathcal{D}_n$ have the same solution set of permutation matrices. Therefore, $h(X)$ is a good approximation of $\|X\|_0$ and problem (3.5) is a good approximation of problem (1.1). The parameter σ in problem (3.5) mainly controls the sparsity of X while the parameter ϵ affects the smoothness of the regularizer. Roughly speaking, problem (3.5) with a large σ returns a sparse X close to a permutation matrix, while problem (3.5) with a relatively large (not too small) ϵ is often easier to be solved.

We now show the exactness of the L_p-norm regularization (3.5).

THEOREM 3.2. *Suppose that $X_{\sigma,p,\epsilon}$ is a global solution of problem* (3.5) *with*

$$(3.6) \qquad\qquad \sigma > \sigma_{p,\epsilon}^* := \max\left\{\frac{\overline{\nu}_f}{\underline{\nu}_h}, 0\right\},$$

where $\overline{\nu}_f$ and $\overline{\nu}_h$ are defined in (2.2) *and* (2.4), *respectively. Then $X_{\sigma,p,\epsilon}$ is also a global solution of problem* (1.1).

Proof. By (2.6) in Proposition 2.1, for any $\sigma > 0$, there holds

$$(3.7) \qquad \sigma h(tX + (1-t)Y) \geq \sigma\left(th(X) + (1-t)h(Y) + \frac{\overline{\nu}_h}{2}t(1-t)\|Y - X\|_{\mathsf{F}}^2\right)$$

for any $X, Y \in \mathcal{D}_n$ and $0 \leq t \leq 1$. By the assumption (2.2), we always have (2.3). Summing (3.7) and (2.3), and noticing the definition of $F_{\sigma,p,\epsilon}(\cdot)$, we conclude that

$F_{\sigma,p,\epsilon}(X)$ is strongly concave with the positive parameter $\sigma\bar{\nu}_h - \bar{\nu}_f$ over $\mathcal{D}_n$. Thus, the global minimum of (3.5) must be attained at the vertices of $\mathcal{D}_n$, namely, $X_{\sigma,p,\epsilon}$ must be a permutation matrix.

Since the global solution X^* of (1.1) is feasible for (3.5), it follows that

$$(3.8) \qquad f(X^*) + \sigma\|X^* + \epsilon\mathbf{1}\|_p^p \geq f(X_{\sigma,p,\epsilon}) + \sigma\|X_{\sigma,p,\epsilon} + \epsilon\mathbf{1}\|_p^p,$$

which, together with $\|X^*\|_p^p = \|X_{\sigma,p,\epsilon}\|_p^p = n(1+\epsilon)^p$, implies $f(X^*) \geq f(X_{\sigma,p,\epsilon})$. Recalling that $X_{\sigma,p,\epsilon}$ is a permutation matrix, we see that $X_{\sigma,p,\epsilon}$ is also a global solution of (1.1). $\square$

Similar to the results in [8, 15], we define the KKT point of (3.5) as follows.

DEFINITION 3.3. *A point $X \in \mathcal{D}_n$ is called as a KKT point of (3.5) if there exist $S \in \mathbb{R}^{n\times n}$ and $\lambda, \mu \in \mathbb{R}^n$ such that, for any $(i,j) \in \mathcal{N} \times \mathcal{N}$, there holds*

$$(3.9) \qquad \begin{cases} X_{ij}S_{ij} = 0, \ S_{ij} \geq 0, \\ (W_{ij} - \lambda_i - \mu_j)(X_{ij} + \epsilon) + \sigma p(X_{ij} + \epsilon)^p = \epsilon S_{ij}, \end{cases}$$

where $W_{ij} = (\nabla f(X))_{ij}$ and S, λ, and μ are the Lagrange multipliers corresponding to the constraints $X \geq 0$, $X\mathbf{e} = \mathbf{e}$, and $X^\mathsf{T}\mathbf{e} = \mathbf{e}$, respectively.

We now estimate the lower bound for nonzero elements of the KKT points of problem (3.5).

THEOREM 3.4. *Suppose that $\bar{X}$ is a KKT point of (3.5). Then, for any $(i,j) \in \Lambda(\bar{X})$, we have*

$$(3.10) \qquad \bar{X}_{ij} \geq \max\left(\bar{c} - \epsilon, 0\right),$$

where $\bar{c} = \left(\|\bar{X}\|_0^{1-p}(n + \|\bar{X}\|_0\epsilon)^p - (n-1)(1+\epsilon)^{p-1} + \sqrt{2n}\frac{L\sqrt{n}+\|\nabla f(\mathbf{0})\|_F}{\sigma p}\right)^{\frac{1}{p-1}}$.

Proof. Since $\bar{X}$ is a KKT point of (3.5), it follows from (3.9) that

$$(3.11) \qquad \sum_{i\in\Lambda(\bar{X}_{\cdot j})} \bar{X}_{ij} = 1, \ \forall j \in \mathcal{N}, \qquad \sum_{j\in\Lambda(\bar{X}_{i\cdot})} \bar{X}_{ij} = 1, \ \forall i \in \mathcal{N}$$

and

$$(3.12) \qquad \bar{W}_{ij} + \sigma p(\bar{X}_{ij} + \epsilon)^{p-1} = \lambda_i + \mu_j, \quad \forall (i,j) \in \Lambda(\bar{X}),$$

where $\bar{W}_{ij} = (\nabla f(\bar{X}))_{ij}$. Multiplying by $\bar{X}_{ij}$ on both sides of (3.12) and then summing them over $(i,j) \in \Lambda(\bar{X})$, we have

$$(3.13)$$
$$\sum_{(i,j)\in\Lambda(\bar{X})} \left(\bar{W}_{ij}\bar{X}_{ij} + \sigma p(\bar{X}_{ij} + \epsilon)^{p-1}\bar{X}_{ij}\right) = \sum_{(i,j)\in\Lambda(\bar{X})} (\lambda_i + \mu_j)\bar{X}_{ij} = (\lambda + \mu)^\mathsf{T}\mathbf{e},$$

where the second equality is due to (3.11) and

$$\sum_{(i,j)\in\Lambda(\bar{X})} (\lambda_i + \mu_j)\bar{X}_{ij} = \sum_{i=1}^n \lambda_i \left(\sum_{j\in\Lambda(\bar{X}_{i\cdot})} \bar{X}_{ij}\right) + \sum_{j=1}^n \mu_j \left(\sum_{i\in\Lambda(\bar{X}_{\cdot j})} \bar{X}_{ij}\right).$$

We next claim that one can pick n elements from different columns and rows of $\bar{W}_{ij} + \sigma p(\bar{X}_{ij} + \epsilon)^{p-1}$ with $i, j \in \Lambda(\bar{X})$ such that their summation is equal to $(\lambda + \mu)^\mathsf{T}\mathbf{e}$.

To prove this claim, we consider any $\tilde{X} \in \Pi_n$ and $\Lambda(\tilde{X}) \subseteq \Lambda(\bar{X})$. The same argument as in the derivation of (3.13) can be used to show

$$(3.14) \quad \sum_{(i,j) \in \Lambda(\bar{X})} \left(\bar{W}_{ij} \tilde{X}_{ij} + \sigma p (\bar{X}_{ij} + \epsilon)^{p-1} \tilde{X}_{ij} \right) = \sum_{(i,j) \in \Lambda(\bar{X})} (\lambda_i + \mu_j) \tilde{X}_{ij} = (\lambda + \mu)^\mathsf{T} \mathbf{e}.$$

Combining (3.13) and (3.14) and then rearranging the obtained equation, we have
$$(3.15)$$
$$\sum_{(i,j) \in \Lambda(\bar{X})} (\bar{X}_{ij} + \epsilon)^{p-1} \tilde{X}_{ij} = \frac{1}{\sigma p} \sum_{(i,j) \in \Lambda(\bar{X})} \bar{W}_{ij} (\bar{X}_{ij} - \tilde{X}_{ij}) + \sum_{(i,j) \in \Lambda(\bar{X})} (\bar{X}_{ij} + \epsilon)^{p-1} \bar{X}_{ij}$$

for any $\tilde{X} \in \Pi_n$ with $\Lambda(\tilde{X}) \subseteq \Lambda(\bar{X})$.

Given any $(i_0, j_0) \in \Lambda(\bar{X})$, we choose some special $\tilde{X} \in \Pi_n$ with $\Lambda(\tilde{X}) \subseteq \Lambda(\bar{X})$ such that $\tilde{X}_{i_0 j_0} = 1$. Hence, using $0 < \bar{X}_{ij} \leq 1$, we have

$$(3.16) \quad \sum_{(i,j) \in \Lambda(\bar{X})} (\bar{X}_{ij} + \epsilon)^{p-1} \tilde{X}_{ij} \geq (\bar{X}_{i_0, j_0} + \epsilon)^{p-1} + (n-1)(1+\epsilon)^{p-1}.$$

It is easy to see that $\|\bar{X} - \tilde{X}\|_\mathsf{F} \leq \sqrt{2n}$. Thus, we obtain

$$(3.17) \quad \sum_{(i,j) \in \Lambda(\bar{X})} \bar{W}_{ij} (\bar{X}_{ij} - \tilde{X}_{ij}) \leq \|\nabla f(\bar{X})\|_\mathsf{F} \|\bar{X} - \tilde{X}\|_\mathsf{F} \leq \sqrt{2n} (L\sqrt{n} + \| \nabla f(\mathbf{0})\|_\mathsf{F}).$$

On the other hand, by the concavity of $(z + \epsilon)^p$ and $(1/z + \epsilon)^p z$ on $(0, 1]$, we know

$$(3.18)$$
$$\sum_{(i,j) \in \Lambda(\bar{X})} (\bar{X}_{ij} + \epsilon)^{p-1} \bar{X}_{ij} \leq \sum_{i=1}^n \sum_{j \in \Lambda(\bar{X}_{i\cdot})} (\bar{X}_{ij} + \epsilon)^p \leq \sum_{i=1}^n \left(\frac{1}{\|\bar{X}_{i\cdot}\|_0} + \epsilon \right)^p \|\bar{X}_{i\cdot}\|_0$$
$$\leq \left(n + \|\bar{X}\|_0 \epsilon \right)^p \|\bar{X}\|_0^{1-p}.$$

Substituting (3.16), (3.17), and (3.18) into (3.15) and by some easy calculations, we have (3.10). □

It is worthwhile to remark that Chen *et al.* [15] and Lu [41] established the lower bound theory for the nonzero elements of the KKT points of unconstrained L_p regularization problems. Based on our limited knowledge, our lower bound estimate (3.10) appears to be a novel explicit lower bound for the nonzero elements of the KKT points of L_p regularization problem with linear (equality and inequality) constraints. It provides a theoretical foundation for rounding the approximate optimal solutions of (3.5) to be permutation matrices. In practice, we usually only solve problem (3.5) inexactly (i.e., stop the algorithm early) and perform the rounding procedure when the approximate solution is near to some permutation matrix. Consequently, the computational cost can be saved.

We next characterize the connections between the local minimizers of problem (3.5) and the permutation matrices. Our results are extentions of the properties in [27] and [38] in the sense that theirs are only for problem (3.5) with $\epsilon = 0$.

THEOREM 3.5. *Each permutation matrix is a local minimizer of* (3.5) *with*

$$(3.19) \quad \sigma > \bar{\sigma}_{p,\epsilon} := \frac{c}{p} \cdot \frac{L(2 + \sqrt{n}) + \|\nabla f(\mathbf{0})\|_\mathsf{F}}{\epsilon^{p-1} - (1/2 + \epsilon)^{p-1}},$$

where $c > 1$ is a constant. On the other hand, any local minimizer of (3.5) *with* $\sigma > \sigma_{p,\epsilon}^*$ *is a permutation matrix, where* $\sigma_{p,\epsilon}^*$ *is defined in* (3.6).

Proof. Let $\bar{X}$ be a permutation matrix and $X^l, l = 1, \ldots, n! - 1$ are all the remaining permutation matrices. For any fixed l, consider the feasible direction $D^l = X^l - \bar{X}$. Then $\bar{X} + tD^l \in \mathcal{D}_n$ for any $t \in [0,1]$. Denote

$$
\begin{aligned}
\Lambda_1^l &= \{(i,j) \mid \bar{X}_{ij} = 0, \ X_{ij}^l = 0\}, \ \Lambda_2^l = \{(i,j) \mid \bar{X}_{ij} = 0, \ X_{ij}^l = 1\}, \\
\Lambda_3^l &= \{(i,j) \mid \bar{X}_{ij} = 1, \ X_{ij}^l = 0\}, \ \Lambda_4^l = \{(i,j) \mid \bar{X}_{ij} = 1, \ X_{ij}^l = 1\}.
\end{aligned}
\tag{3.20}
$$

Clearly, we can see that $|\Lambda_1^l|, |\Lambda_4^l| \geq 0$ and $|\Lambda_2^l| = |\Lambda_3^l| \geq 2$, and $\mathcal{N} \times \mathcal{N} = \bigcup_{i=1}^4 \Lambda_i^l$. In addition, we also have that $\|D^l\|_{\mathsf{F}}^2 = 2|\Lambda_2^l| \geq 4$. By (2.1) and the mean-value theorem, for any $t \in (0,1]$, there exists $\xi \in (0,1)$ such that

$$
\begin{aligned}
f(\bar{X} + tD^l) - f(\bar{X}) &= t\langle \nabla f(\bar{X} + \xi tD^l), D^l \rangle \\
&\geq -t\|D^l\|_{\mathsf{F}}\|\nabla f(\bar{X} + \xi tD^l)\|_{\mathsf{F}} \\
&\geq -t\|D^l\|_{\mathsf{F}} \cdot \left(L\|\bar{X} + \xi tD^l\|_{\mathsf{F}} + \|\nabla f(\mathbf{0})\|_{\mathsf{F}}\right) \\
&\geq -t\left(L(\|D^l\|_{\mathsf{F}}^2 + \|D^l\|_{\mathsf{F}}\sqrt{n}) + \|D^l\|_{\mathsf{F}}\|\nabla f(\mathbf{0})\|_{\mathsf{F}}\right),
\end{aligned}
\tag{3.21}
$$

where the last inequality is due to $\|\bar{X}\|_{\mathsf{F}} = \sqrt{n}$. Moreover, for any $t \in (0, 1/2)$, we have

$$
\begin{aligned}
h(\bar{X} + tD^l) - h(\bar{X}) &= \sum_{(i,j) \in \bigcup_{i=1}^4 \Lambda_i^l} \left((\bar{X}_{ij} + tD_{ij}^l + \epsilon)^p - (\bar{X}_{ij} + \epsilon)^p\right) \\
&= |\Lambda_2^l|\left((t + \epsilon)^p - \epsilon^p\right) + |\Lambda_3^l|\left((1 - t + \epsilon)^p - (1 + \epsilon)^p\right) \\
&> \frac{p}{2}\|D^l\|_{\mathsf{F}}^2\left((t + \epsilon)^{p-1} - (1 - t + \epsilon)^{p-1}\right)t \\
&> \frac{p}{2}\|D^l\|_{\mathsf{F}}^2\left((t + \epsilon)^{p-1} - (1/2 + \epsilon)^{p-1}\right)t,
\end{aligned}
\tag{3.22}
$$

where the second equality is due to (3.20); the first inequality uses $|\Lambda_2^l| = |\Lambda_3^l| = \frac{\|D^l\|_{\mathsf{F}}^2}{2}$ and the strong concavity of the function $h(x) = (x + \epsilon)^p$; the second inequality follows from $0 < p < 1$ and $t \in (0, 1/2)$. Denote $\tilde{t} := \left(c^{-1}\epsilon^{p-1} + (1 - c^{-1})(1/2 + \epsilon)^{p-1}\right)^{\frac{1}{p-1}} - \epsilon$. Using the fact that the function $(\xi_1 + \xi_2(1/2 + z)^{p-1})^{\frac{1}{p-1}}$ with $\xi_1, \xi_2 > 0$ is increasing with z in $(0, \infty)$, we have $\tilde{t} > \left(c^{-1}\epsilon^{p-1} + (1 - c^{-1})\epsilon^{p-1}\right)^{\frac{1}{p-1}} - \epsilon = 0$. Furthermore, if we restrict $t \in (0, \bar{t})$ with $\bar{t} = \min(1/2, \tilde{t})$, we obtain from (3.22) that

$$
h(\bar{X} + tD^l) - h(\bar{X}) > \frac{p}{2}\|D^l\|_{\mathsf{F}}^2\frac{\epsilon^{p-1} - (1/2 + \epsilon)^{p-1}}{c},
$$

which together with (3.19) and $\|D^l\|_{\mathsf{F}} \geq 2$ implies that

$$
\sigma(h(\bar{X} + tD^l) - h(\bar{X})) > t\left(L(\|D^l\|_{\mathsf{F}}^2 + \|D^l\|_{\mathsf{F}}\sqrt{n}) + \|D^l\|_{\mathsf{F}}\|\nabla f(\mathbf{0})\|_{\mathsf{F}}\right).
\tag{3.23}
$$

Combining (3.21) and (3.23), we have

$$
F_{\sigma,p,\epsilon}(\bar{X} + tD^l) - F_{\sigma,p,\epsilon}(\bar{X}) > 0
\tag{3.24}
$$

for any $t \in (0, \bar{t})$. This fact means that $D^l, l = 1, \ldots, n! - 1$, are strictly increasing feasible directions. Let $\mathrm{Conv}(\bar{X}, \bar{t})$ denote the convex hull spanned by points $\bar{X}$ and

$\bar{X} + \bar{t}D^l$, $l = 1, \ldots, n! - 1$. Thus for any $X \in \text{Conv}(\bar{X}, \bar{t})$, we have $F_{\sigma,p,\epsilon}(X) > F_{\sigma,p,\epsilon}(\bar{X})$ by (3.24) and the strict concavity of $h(X)$. Moreover, one can always choose a sufficiently small but fixed $t > 0$ such that $\mathcal{B}(\bar{X}, t) \cap \mathcal{D}_n \subset \text{Conv}(\bar{X}, \bar{t})$, where $\mathcal{B}(\bar{X}, t) = \{X \in \mathbb{R}^{n \times n} \mid \|X - \bar{X}\|_{\mathsf{F}} \le t\}$. Consequently, $\bar{X}$ is a local minimizer of problem (3.5).

On the other hand, if a local minimizer $\bar{X}$ of (3.5) with $\sigma > \sigma_{p,\epsilon}^*$ is not a permutation matrix, there must exist a feasible direction D such that $\bar{X} + t'D$ and $\bar{X} - t'D$ both belong to $\mathcal{D}_n$ for some sufficiently small positive t'. Note that the function $F_{\sigma,p,\epsilon}$ is strongly concave when $\sigma > \sigma_{p,\epsilon}^*$. Thus, we must have

$$\min\left(F_{\sigma,p,\epsilon}(\bar{X} - t'D), F_{\sigma,p,\epsilon}(\bar{X} + t'D)\right) < F_{\sigma,p,\epsilon}(\bar{X}),$$

which is a contradiction to the local optimality of $\bar{X}$. Therefore, $\bar{X}$ must be a permutation matrix. □

Theorem 3.5 on the equivalence between the local minimizers of problem (3.5) and the permutation matrices implies that finding a local minimizer of problem (3.5) is easy but finding the global solution of problem (3.5) is as difficult as finding the global solution of the original NP-hard problem (1.1). Our extensive numerical experiments show that our algorithm can often efficiently identify high quality solutions, in particular, with the help of certain techniques to exclude a large portion of non-optimal permutation matrices.

4. An L_p regularization algorithm for (1.1). In this section, we give an L_p regularization algorithmic framework and its practical version for solving problem (1.1). Since the projection onto the set of doubly stochastic matrices needs to be computed many times in our proposed algorithm, we also propose a fast dual gradient method for solving it.

4.1. An L_p regularization algorithmic framework. Theorem 3.2 implies that, for any $\epsilon > 0$, we can choose a fixed σ larger than $\sigma_{p,\epsilon}^*$ to solve (3.5) and obtain a permutation matrix for problem (1.1). In practical implementation, it might be better to dynamically increase σ from an initial value σ_0 since a large σ may result in the ill-conditioned subproblem. Since $\sigma_{p,\epsilon}^*$ is increasing with respect to ϵ if $\bar{\nu}_f > 0$ (see Theorem 3.2), dynamically decreasing ϵ from a relatively large value ϵ_0 is helpful in finding a permutation matrix relatively faster.

To summarize, we solve a sequence of (3.5) with strictly increasing parameters $\{\sigma_k \mid k = 0, 1, 2, \ldots\}$ and decreasing parameters $\{\epsilon_k \mid k = 0, 1, 2, \ldots\}$. For a fixed k, we denote $X_k^{(0)}$ as the starting point for solving (3.5) with σ_k and ϵ_k. After computing a KKT point X_k for (3.5) with σ_k and ϵ_k, we use a warm start technique to set

$$(4.1) \qquad X_{k+1}^{(0)} := X_k$$

if X_k is not a KKT point of problem (3.5) with σ_{k+1} and ϵ_{k+1}. Otherwise, we set

$$(4.2) \qquad X_{k+1}^{(0)} = \text{a perturbation of } X_k \text{ in } \mathcal{D}_n.$$

The following theorem shows that once we find a permutation matrix, which is a KKT point of problem (3.5) with some σ_k and ϵ_k, it is reasonable to stop the iterative procedure.

THEOREM 4.1. *If a permutation matrix X is a KKT point of (3.5) with fixed $\hat{\sigma}$ and $\hat{\epsilon} > 0$, then it is also a KKT point of (3.5) with σ and ϵ satisfying $\sigma \ge \hat{\sigma}$ and $0 < \epsilon \le \hat{\epsilon}$.*

Proof. Denote the corresponding permutation vector of X as π. Let the corresponding multipliers be $\lambda^{\hat{\sigma},\hat{\epsilon}}$, $\mu^{\hat{\sigma},\hat{\epsilon}}$, and $S^{\hat{\sigma},\hat{\epsilon}}$. Then we know from (3.9) that for $i, j \in \mathcal{N}$, there holds

$$(4.3) \qquad S_{ij}^{\hat{\sigma},\hat{\epsilon}} = \begin{cases} W_{ij} - \lambda_i^{\hat{\sigma},\hat{\epsilon}} - \mu_j^{\hat{\sigma},\hat{\epsilon}} + \hat{\sigma}p(1+\hat{\epsilon})^{p-1}, & i = \pi(j), \\ W_{ij} - \lambda_i^{\hat{\sigma},\hat{\epsilon}} - \mu_j^{\hat{\sigma},\hat{\epsilon}} + \hat{\sigma}p\hat{\epsilon}^{p-1}, & i \neq \pi(j), \end{cases}$$

and $S_{ij}^{\hat{\sigma},\hat{\epsilon}} = 0$ if $i = \pi(j)$ and $S_{ij}^{\hat{\sigma},\hat{\epsilon}} \geq 0$ if $i \neq \pi(j)$. For any $\sigma \geq \hat{\sigma}$ and $0 < \epsilon \leq \hat{\epsilon}$, define

$$\lambda^{\sigma,\epsilon} = \lambda^{\hat{\sigma},\hat{\epsilon}} + p\left(\sigma(1+\epsilon)^{p-1} - \hat{\sigma}(1+\hat{\epsilon})^{p-1}\right)\mathbf{e}, \quad \mu^{\sigma,\epsilon} = \mu^{\hat{\sigma},\hat{\epsilon}}$$

and

$$(4.4) \qquad S_{ij}^{\sigma,\epsilon} = \begin{cases} W_{ij} - \lambda_i^{\sigma,\epsilon} - \mu_j^{\sigma,\epsilon} + \sigma p(1+\epsilon)^{p-1}, & i = \pi(j), \\ W_{ij} - \lambda_i^{\sigma,\epsilon} - \mu_j^{\sigma,\epsilon} + \sigma p\epsilon^{p-1}, & i \neq \pi(j). \end{cases}$$

We now show that $S^{\sigma,\epsilon}$, $\lambda^{\sigma,\epsilon}$, and $\mu^{\sigma,\epsilon}$ satisfy KKT condition (3.9). For any $i = \pi(j)$, we have from (4.3) and (4.4) that

$$(4.5) \qquad S_{ij}^{\sigma,\epsilon} = W_{ij} - \lambda_i^{\sigma,\epsilon} - \mu_j^{\sigma,\epsilon} + \sigma p(1+\epsilon)^{p-1} = S_{ij}^{\hat{\sigma},\hat{\epsilon}} = 0;$$

for any $i \neq \pi(j)$, we have

$$(4.6) \qquad \begin{aligned} S_{ij}^{\sigma,\epsilon} &= W_{ij} - \lambda_i^{\sigma,\epsilon} - \mu_j^{\sigma,\epsilon} + \sigma p\epsilon^{p-1} \\ &= S_{ij}^{\hat{\sigma},\hat{\epsilon}} + p\hat{\sigma}\left((1+\hat{\epsilon})^{p-1} - \hat{\epsilon}^{p-1}\right) - p\sigma\left((1+\epsilon)^{p-1} - \epsilon^{p-1}\right) \\ &\geq S_{ij}^{\hat{\sigma},\hat{\epsilon}} + p(\hat{\sigma} - \sigma)\left((1+\epsilon)^{p-1} - \epsilon^{p-1}\right) \geq S_{ij}^{\hat{\sigma},\hat{\epsilon}} \geq 0, \end{aligned}$$

where the first inequality is due to that $(1+z)^{p-1} - z^{p-1}$ is increasing in z in $(0,\infty)$. Combining (4.5) and (4.6), we can see that X is also a KKT point of (3.5) with $\sigma \geq \hat{\sigma}$ and $0 < \epsilon \leq \hat{\epsilon}$. $\square$

The L_p regularization algorithmic framework for solving problem (1.1) is outlined in Algorithm 1.

Algorithm 1: An L_p regularization algorithmic framework for problem (1.1)

---

1 Given $X_0 \in \mathcal{D}_n$, set $k = 0$, $\epsilon_0 > 0$, $\sigma_0 > 0$, $\hat{c} > 0$.

2 while $\|X_k\|_p^p > n$ **do**

3   Set $X_k^{(0)}$ according to (4.1) and (4.2).

4   Find a KKT point X_k of problem (3.5) with σ_k and ϵ_k staring from $X_k^{(0)}$ such that $F_{\sigma_k,p,\epsilon_k}(X_k) \leq F_{\sigma_k,p,\epsilon_k}(X_k^{(0)})$.

5   Choose $\sigma_{k+1} > \sigma_k + \hat{c}$, $0 < \epsilon_{k+1} \leq \epsilon_k$ and set $k = k + 1$.

The convergence of Algorithm 1 is summarized as follows.

THEOREM 4.2. *Algorithm 1 returns a permutation matrix in a finite number of iterations.*

Proof. By the update scheme of σ_k and ϵ_k, we have $\sigma_k > \max\{\sigma_{p,0}^*, \bar{\sigma}_{p,\epsilon_0}\}$ for some finite k, where $\sigma_{p,\epsilon}^*$ and $\bar{\sigma}_{p,\epsilon}$ are defined in (3.6) and (3.19), respectively. The update scheme of ϵ_k yields that $\sigma_{p,0}^* > \sigma_{p,\epsilon_k}^*$ and $\bar{\sigma}_{p,\epsilon_0} \geq \bar{\sigma}_{p,\epsilon_k}$. It follows from the proof of Theorem 3.2 that $F_{\sigma_k,p,\epsilon_k}(X)$ with $\sigma_k > \max\{\sigma_{p,0}^*, \bar{\sigma}_{p,\epsilon_0}\}$ and $\epsilon_k > 0$ is strongly concave. Consequently, any KKT point X_k of problem (3.5) with $\sigma > \max\{\sigma_{p,0}^*, \bar{\sigma}_{p,\epsilon_0}\}$

and $0 < \epsilon \le \epsilon_0$ is either a local minimizer (which must be a permutation matrix) or the unique global maximizer of problem (3.5). Consider the case when X_k is the unique global maximizer of problem (3.5) with $(\sigma, \epsilon) = (\sigma_{k+1}, \epsilon_{k+1})$. By the choice of $X_{k+1}^{(0)}$, we have $F_{\sigma_{k+1}, p, \epsilon_{k+1}}(X_{k+1}^{(0)}) < F_{\sigma_{k+1}, p, \epsilon_{k+1}}(X_k)$. This inequality together with the fact that $F_{\sigma_{k+1}, p, \epsilon_{k+1}}(X_{k+1}) \le F_{\sigma_{k+1}, p, \epsilon_{k+1}}(X_{k+1}^{(0)})$ from Line 4 in Algorithm 1 implies that the KKT point X_{k+1} must be a local minimizer of problem (3.5), which means that X_{k+1} is a permutation matrix. In the other case, we choose $X_{k+1}^{(0)} = X_k$ by (4.1). A similar argument shows that X_{k+1} is a permutation matrix. $\square$

4.2. A practical L_p regularization algorithm. We first use the projected gradient method with the BB step sizes [5, 9] to compute an approximate KKT point of problem (3.5). Specifically, starting from initial $X_k^{(0)}$, the projected gradient method for solving problem (3.5) with σ_k and ϵ_k iterates as follows:

$$(4.7) \qquad X_k^{(i+1)} = X_k^{(i)} + \delta^j D^{(i)}, \quad \delta \in (0, 1),$$

where the search direction $D^{(i)} = \mathcal{P}_{\mathcal{D}_n}\big(X_k^{(i)} - \alpha_i \nabla F_{\sigma_k, p, \epsilon_k}(X_k^{(i)})\big) - X_k^{(i)}$ and $\mathcal{P}_{\mathcal{D}_n}(\cdot)$ is the projection onto $\mathcal{D}_n$. The step size α_i is set to be the alternative usage of the large and short BB steps [18, 54]. The parameter j is the smallest nonnegative integer satisfying the nonmonotone line search condition:

$$(4.8) \qquad F_{\sigma_k, p, \epsilon_k}(X_k^{(i)} + \delta^j D^{(i)}) \le C_i + \theta \delta^j \langle \nabla F_{\sigma_k, p, \epsilon_k}(X_k^{(i)}), D^{(i)} \rangle, \quad \theta \in (0, 1),$$

where the reference function value C_{i+1} is updated as the convex combination of C_i and $F_{\sigma_k, p, \epsilon_k}(X_k^{(i)})$, i.e., $C_{i+1} = (\eta Q_i C_i + F_{\sigma_k, p, \epsilon_k}(X_k^{(i)}))/Q_{i+1}$, where $Q_{i+1} = \eta Q_i + 1$, $\eta = 0.85$, $C_0 = F_{\sigma_k, p, \epsilon_k}(X_k^{(0)})$ and $Q_0 = 1$. See [58] for more detailed information.

Any intermediate point $X \in \mathcal{D}_n$ can be converted to a permutation matrix via rounding or a greedy procedure, which can be further improved by performing local search. The specific algorithm will be introduced later in Section 6.1. Other suitable greedy approaches can be adopted as well. The greedy procedure not only offers a better permutation matrix, but also guides the update of the parameter ϵ. Specifically, for any point $X_k^{(i)}$, we generate $\hat{X}_k^{(i)}$ from $X_k^{(i)}$ such that $\hat{X}_k^{(i)} \in \mathcal{N}_2^{\text{best}}(\hat{X}_k^{(i)})$ and update

$$f_k^{\text{best}} = \min_i \{f(\hat{X}_k^{(i)})\} \text{ and } X_k^{\text{best}} = \arg\min_i \{f(\hat{X}_k^{(i)})\}.$$

Let f^{best} be the best function value among $f_1^{\text{best}}, \ldots, f_{k-1}^{\text{best}}$. If f_k^{best} is less than or equal to f^{best}, we set $\epsilon_k = \epsilon_{k-1}$; otherwise we set $\epsilon_k = \gamma \epsilon_{k-1}$ with $\gamma \in (0, 1)$. Moreover, recall that $\bar{\sigma}_{p, \epsilon}$ in Theorem 3.5 is increasing with respect to ϵ and $\lim_{\epsilon \to 0} \bar{\sigma}_{p, \epsilon} = 0$. Therefore, if ϵ_k is too small, then nearly all of the permutation matrices are local minimizers of (3.5). This might lead to finding a bad permutation matrix (in terms of solving problem (1.1)) with a high probability. To avoid this drawback, we propose to set a safeguard $\epsilon_{\min}$ for ϵ_k. We summarize the strategy of updating ϵ_k as follows

$$(4.9) \quad \begin{aligned} &\text{if} \quad f_k^{\text{best}} < f^{\text{best}} \\ &\qquad f^{\text{best}} = f_k^{\text{best}}, X^{\text{best}} = X_k^{\text{best}}, \epsilon_k = \epsilon_{k-1}, \\ &\text{else} \\ &\qquad \epsilon_k = \max\{\gamma \epsilon_{k-1}, \epsilon_{\min}\}. \\ &\text{end} \end{aligned}$$

Finally, we use the continuation scheme to dynamically update σ such that problem (3.5) changes from being strongly convex to being strongly concave. To achieve this goal, we choose $\sigma_0 = \min\left\{\frac{\nu_f}{p(1-p)}\epsilon_0^{2-p}, \ \sigma_-\right\}$, where $\underline{\nu}_f$ is defined in (2.2) and $\sigma_- < 0$ is given. Motivated by the strategy used in [55], we update σ_{k+1} by

$$(4.10) \qquad \sigma_{k+1} := \min\{\tilde{\sigma}_{k+1}, \sigma_{\max}\} \text{ and } \tilde{\sigma}_{k+1} = \begin{cases} \frac{1}{2}\sigma_k, & \sigma_k \leq \sigma_-, \\ 0, & \sigma_- < \sigma_k < 0, \\ \sigma_+, & \sigma_k = 0, \\ 2\sigma_k, & \sigma_k \geq \sigma_+, \end{cases}$$

where $\sigma_+ = -2^{-l}\sigma_0$, $l = \lceil \log_2(-\sigma_0) \rceil$ and $\sigma_{\max} > 0$ is a safeguard for σ.

We summarize the practical L_p regularization algorithm for solving problem (1.1) in Algorithm 2. Specifically, the stopping rules on line 5 are $\text{tol}_i^x = \|X_k^{(i)} - X_k^{(i-1)}\|_{\mathsf{F}}/\sqrt{n}$, $\text{tol}_i^f = \frac{\left|F_{\sigma_k,p,\epsilon_k}(X_k^{(i)}) - F_{\sigma_k,p,\epsilon_k}(X_k^{(i-1)})\right|}{1 + \left|F_{\sigma_k,p,\epsilon_k}(X_k^{(i-1)})\right|}$ and $\tau_k^x = \max\left\{\tau_0^x/k^3, \ \tau_{\min}^x\right\}$, $\tau_k^f = \max\{\tau_0^f/k^3, \ \tau_{\min}^f\}$, where $\tau_{\min}^x, \tau_{\min}^f > 0$ are some small parameters. Of course, other types of criteria can also be applied as well. As discussed after the proof of Theorem 3.4, the lower bound (3.10) may be helpful in Line 9.

Algorithm 2: A practical L_p regularization algorithm for problem (1.1).

1 Given $X_0 \in \mathcal{D}_n$, set $\epsilon_0, \tau_0^x, \tau_0^f, \tau_{\min}^x, \tau_{\min}^f > 0$, $\sigma_0 \leq 0$, $\theta, \delta, \gamma, \text{tol} \in (0,1)$.
2 Set $k = 0$, $f^{\text{best}} = \infty$.
3 **while** $\|X_k\|_p^p/n - 1 > \text{tol}$ **do**
        `// Lines 4 - 9:  compute an approximate KKT point of` (3.5)
4   Choose $X_k^{(0)}$ by (4.1) and (4.2). Set $i = 0$, $\alpha_i > 0$, $\text{tol}_i^x = \text{tol}_i^f = \infty$.
5   **while** $\text{tol}_i^x > \tau_k^x$ or $\text{tol}_i^f > \tau_k^f$ **do**
6   Compute $D^{(i)} = \mathcal{P}_{\mathcal{D}_n}\left(X_k^{(i)} - \alpha_i \nabla F_{\sigma_k,p,\epsilon_k}(X_k^{(i)})\right) - X_k^{(i)}$.
7   Find the smallest j such that δ^j satisfies (4.8).
8   Set $X_k^{(i+1)} = X_k^{(i)} + \delta^j D_k^{(i)}$ and $i = i + 1$.
9   Compute $\hat{X}_k^{(i)} \in \mathcal{N}_2^{\text{best}}(\hat{X}_k^{(i)})$ starting from $X_k^{(i)}$ and update f_k^{best}.
10   Update ϵ_{k+1}, σ_{k+1}, f^{best}, and X^{best} by (4.9) and (4.10).
11   Set $X_{k+1} = X_k^{(i)}$ and $k = k + 1$.

4.3. Fast dual gradient algorithm for computing projection onto $\mathcal{D}_n$. Note that the dominant computational cost of Algorithm 2 is to compute projections onto the set $\mathcal{D}_n$. In this subsection, we propose a fast dual gradient method for solving the projection problem

$$(4.11) \qquad \min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2}\|X - C\|_{\mathsf{F}}^2 \quad \text{s.t.} \quad X\mathbf{e} = \mathbf{e}, \ X^{\mathsf{T}}\mathbf{e} = \mathbf{e}, \ X \geq \mathbf{0},$$

where $C \in \mathbb{R}^{n \times n}$ is given.

The Lagrangian dual problem of (4.11) is

$$(4.12) \qquad \max_{y,z} \min_{X \geq 0} \mathcal{L}(X, y, z),$$

where $\mathcal{L}(X, y, z) = \frac{1}{2}\|X - C\|_{\mathsf{F}}^2 - \langle y, X\mathbf{e} - \mathbf{e}\rangle - \langle z, X^{\mathsf{T}}\mathbf{e} - \mathbf{e}\rangle$, in which $y, z \in \mathbb{R}^n$ are the Lagrange multipliers associated with the linear constraints $X\mathbf{e} = \mathbf{e}$ and $X^{\mathsf{T}}\mathbf{e} = \mathbf{e}$,

respectively. Let $\mathcal{P}_+(\cdot)$ denote the projection onto the nonnegative orthant. Then the dual problem (4.12) can be equivalently rewritten as

$$(4.13) \qquad \min_{y,z} \theta(y,z) := \frac{1}{2}\|\mathcal{P}_+\left(C + y\mathbf{e}^\mathsf{T} + \mathbf{e}z^\mathsf{T}\right)\|_\mathsf{F}^2 - \langle y + z, \mathbf{e}\rangle.$$

It can be verified that $\theta(y,z)$ is convex and continuously differentiable and

$$\nabla\theta(y,z) = \left[\begin{array}{c} \mathcal{P}_+\left(C + y\mathbf{e}^\mathsf{T} + \mathbf{e}z^\mathsf{T}\right)\mathbf{e} - \mathbf{e} \\ \mathcal{P}_+\left(C + y\mathbf{e}^\mathsf{T} + \mathbf{e}z^\mathsf{T}\right)^\mathsf{T}\mathbf{e} - \mathbf{e} \end{array}\right].$$

Note that [4] computed a nearest doubly stochastic matrix with a fixed entry via solving a dual problem similar to (4.13), by the nonsmooth Newton method.

We propose to use the gradient method using the BB step sizes to solve the unconstrained problem (4.13). The method is named as "dualBB". After obtaining the solution y^* and z^* of (4.13), we recover the projection of C as $\mathcal{P}_{\mathcal{D}_n}(C) = \mathcal{P}_+\left(C + y^*\mathbf{e}^\mathsf{T} + \mathbf{e}(z^*)^\mathsf{T}\right)$. Our preliminary numerical results on the QAPLIB demonstrate that our proposed dualBB method is faster than MOSEK in many cases.

5. Enhanced variants of L_p regularization algorithm. In this section, we combine L_p regularization Algorithm 2 with two techniques to further enhance its performance. The first variant uses the cutting plane technique and the second one is based on a novel L_p regularization model, which has an additional negative proximal point term compared with the model (3.5).

5.1. L_p regularization algorithm with cutting plane technique. Let $\tilde{X}$ be a permutation matrix and let $\tilde{f} = f(\tilde{X})$. Assume that $\tilde{X}$ is not optimal. Then there must exist a constant $c_1 > 0$ such that $f^* \leq \tilde{f} - c_1$. Next, we construct some cuts to shrink the feasible region Π_n of problem (1.1) based on the information of $\tilde{X}$.

Denote $F_\omega(X) := f(X) + \omega\|X\|_\mathsf{F}^2$, where the parameter ω is chosen such that F_ω is strongly convex. The ideal cut

$$(5.1) \qquad\qquad \mathrm{QC}(\tilde{X}): \quad F_\omega(X) \leq \tilde{f} - c_1 + \omega n$$

is hard to be handled because of the nonlinear function $f(X)$. The first order approximation

$$(5.2) \qquad \mathrm{LC1}(\tilde{X}): \quad F_\omega(\tilde{X}) + \langle\nabla F_\omega(\tilde{X}), X - \tilde{X}\rangle \leq \tilde{f} - c_1 + \omega n.$$

is easy but it might not be tight.

Suppose that $\tilde{X}$ is the best point in its 2-neighborhood, a linear cut is proposed in [7] as

$$(5.3) \qquad\qquad \mathrm{LC2}(\tilde{X}): \quad \langle\tilde{X}, X\rangle \leq n - 3,$$

which precisely cuts $\frac{n(n-1)}{2} + 1$ permutation matrices from Π_n.

By using the above two cuts (5.2) and (5.3), we can shrink Π_n to a strictly smaller set $\mathcal{A}_{\Pi_n} := \Pi_n \cap \mathrm{LC1}(\tilde{X}) \cap \mathrm{LC2}(\tilde{X})$, which still contains the global solution of problem (1.1). Therefore, we can solve the following problem (instead of problem (1.1)):

$$(5.4) \qquad\qquad \min_{X \in \mathcal{A}_{\Pi_n}} f(X).$$

By replacing $\mathcal{D}_n$ by $\mathcal{A} := \mathcal{D}_n \cap \mathrm{LC1}(\tilde{X}) \cap \mathrm{LC2}(\tilde{X})$, Algorithm 2 can be employed to solve problem (5.4), where the L_p regularization subproblem becomes

$$(5.5) \qquad\qquad \min_{X \in \mathcal{A}} \ F_{\sigma,p,\epsilon}(X).$$

The L_p regularization algorithm with the cutting place technique (L_p-CP) is given as follows.

Algorithm 3: An L_p-CP algorithm for problem (1.1).

---

1 Given $\mathcal{A} = \mathcal{D}_n$, set $K_{\max} > 0$ and $K = 0$.
2 **while** $K < K_{\max}$ **do**
3 $\quad$ Solve (5.4) by Algorithm 2 with subproblem (3.5) there replaced by (5.5), and obtain X_K^{best}.
4 $\quad$ Set $\mathcal{A} = \mathcal{A} \cap \big(\mathrm{LC1}(X_K^{\mathrm{best}}) \cap \mathrm{LC2}(X_K^{\mathrm{best}})\big)$.
5 $\quad$ Set $X^{\mathrm{best}} = \arg\min_K \{f(X_K^{\mathrm{best}})\}$ and $K = K + 1$.

The solution quality of Algorithm 3 is generally better than that of Algorithm 2 due to the cutting plane technique. However, Algorithm 3 might be slower than Algorithm 2 because computing the projections onto the restricted domain $\mathcal{A}$ in Algorithm 3 is slightly time consuming than computing the projections onto the set $\mathcal{D}_n$ and computational cost of the projections is the dominant part in both algorithms. Note that Algorithm 3 with $K_{\max} = 1$ reduces to Algorithm 2.

5.2. L_p regularization with negative proximal point technique. The cutting plane technique may not improve the performance of Algorithm 2. For instance, Algorithm 3 with $K_{\max} = 4$ and Algorithm 2 find the same permutation matrix when they are used to solve the problem instance "chr20c" from QAPLIB. By investigating the trajectories of the permutation matrices generated by Algorithm 3, we find that the trajectories (corresponding to different K) are nearly the same for this problem. To avoid this phenomenon, we need to push the trajectories of the permutation matrices corresponding to different K far away from each other. It can be achieved by adding a negative proximal point term to $f(X)$.

Given $\mu > 0$ and $\hat{X} \in \mathcal{D}_n$, consider problem

$$(5.6) \qquad\qquad \min_{X \in \Pi_n} \ f(X) - \mu \|X - \hat{X}\|_{\mathsf{F}}^2.$$

The next theorem shows that the solution of (5.6) with an appropriate choice of μ is a solution of (1.1). Moreover, the solution of (5.6) is the farthest point away from the given point $\hat{X}$ among the solution set of (1.1).

THEOREM 5.1. *Let $c_2 > 0$ be such that*

$$(5.7) \qquad |f(X) - f(Y)| \geq c_2, \quad \forall \ X, Y \in \Pi_n \ with \ f(X) \neq f(Y).$$

Suppose that $0 < \mu < \frac{c_2}{2n}$. Then any solution X_μ^* of (5.6) is also a solution of (1.1). Moreover, for any solution X^* of (1.1), we have

$$(5.8) \qquad\qquad \|X_\mu^* - \hat{X}\|_{\mathsf{F}} \geq \|X^* - \hat{X}\|_{\mathsf{F}}.$$

Proof. From the optimality of X_μ^*, we obtain

$$(5.9) \qquad f(X_\mu^*) - \mu \|X_\mu^* - \hat{X}\|_{\mathsf{F}}^2 \leq f(X^*) - \mu \|X^* - \hat{X}\|_{\mathsf{F}}^2,$$

which is equivalent to

$$f(X_\mu^*) \leq f(X^*) + \mu \left(\|X_\mu^*\|_{\mathsf{F}}^2 - \|X^*\|_{\mathsf{F}}^2 + 2\langle X^* - X_\mu^*, \hat{X} \rangle \right).$$

Combining the above inequality with the facts that $\|X_\mu^*\|_{\mathsf{F}} = \|X^*\|_{\mathsf{F}} = \sqrt{n}$, $\langle X^*, \hat{X} \rangle \leq n$, and $\langle X_\mu^*, \hat{X} \rangle \geq 0$ yields

$$(5.10) \qquad f(X_\mu^*) \leq f(X^*) + 2\mu n < f(X^*) + c_2,$$

where the last inequality is due to the choice of μ. Moreover, it follows from the optimality of X^* that $f(X^*) \leq f(X_\mu^*)$, which, together with (5.7) and (5.10), implies $f(X_\mu^*) = f(X^*)$. This shows that X_μ^* is a solution to (1.1). From (5.9) and $f(X_\mu^*) = f(X^*)$, we immediately obtain the desired result (5.8). □

By setting $\hat{X} = \frac{1}{K}\sum_{i=1}^{K} \hat{X}_i$ in Theorem 5.1, where $\hat{X}_i \in \mathcal{D}_n$ for all $i = 1, \dots, K$, and combining it with Theorem 3.2, we obtain the following theorem.

THEOREM 5.2. *Suppose* $0 < \mu < \frac{c_2}{2n}$ *and* $\sigma > \max((\bar{\nu}_f - 2\mu)/\bar{\nu}_h, 0)$. *Then any global solution of*

$$(5.11) \qquad \min_{X \in \mathcal{D}_n} \; f(X) + \sigma \|X + \epsilon \mathbf{1}\|_p^p - \mu \left\| X - \frac{1}{K}\sum_{i=1}^{K} \hat{X}_i \right\|_{\mathsf{F}}^2$$

is also a global solution of problem (1.1). *Moreover, it is one of the farthest solutions of* (1.1) *away from* $\frac{1}{K}\sum_{i=1}^{K} \hat{X}_i$.

Based on Theorem 5.2, we present the L_p regularization algorithm with the negative proximal point technique, which is dubbed as L_p-negProx.

Algorithm 4: An L_p-negProx algorithm for problem (1.1).

---

1 Set $\mu > 0$, $K_{\max} > 0$ and $K = 0$. Define $X_K^{\mathrm{best}} = \mathbf{0}$.

2 **while** $K < K_{\max}$ and $X_K^{\mathrm{best}} \notin \{X_1^{\mathrm{best}}, \dots, X_{K-1}^{\mathrm{best}}\}$ **do**

3 $\quad$ Solve (5.6) with $\hat{X} = \frac{1}{K}\sum_{i=1}^{K} X_i^{\mathrm{best}}$ by Algorithm 2 with subproblem (3.5) replaced by (5.11), and obtain X_K^{best}.

4 $\quad$ Set $\mu = \mu/2$, $X^{\mathrm{best}} = \arg\min_K\{f(X_K^{\mathrm{best}})\}$, and $K = K + 1$.

Consider the problem "chr20c" in QAPLIB again. By simply setting $\mu \equiv 0.1$, Algorithm 4 is able to find the global solution. More detailed numerical results are reported in Section 6.3 to demonstrate the effectiveness of the proposed negative proximal point technique.

We can also combine Algorithm 2 with both the cutting plane technique and the negative proximal point technique. The resulting algorithm is named as L_p-CP-negProx and it is described as follows.

Algorithm 5: An L_p-CP-negProx algorithm for problem (1.1).

---

1 Given $\mathcal{A} := \mathcal{D}_n$, set $\mu > 0$, $K_{\max} > 0$ and $K = 0$. Define $X_K^{\mathrm{best}} = \mathbf{0}$.

2 **while** $K < K_{\max}$ and $X_K^{\mathrm{best}} \notin \{X_1^{\mathrm{best}}, \dots, X_{K-1}^{\mathrm{best}}\}$ **do**

3 $\quad$ Solve (5.6) with $\hat{X} = \frac{1}{K}\sum_{i=1}^{K} \hat{X}_i^{\mathrm{best}}$ by Algorithm 2 with subproblem (3.5) replaced by (5.11), wherein $\mathcal{D}_n$ being replaced by $\mathcal{A}$, and obtain X_K^{best}.

4 $\quad$ Set $\mathcal{A} = \mathcal{A} \cap \mathrm{LC1}(X_K^{\mathrm{best}}) \cap \mathrm{LC2}(X_K^{\mathrm{best}})$.

5 $\quad$ Set $\mu = \mu/2$, $X^{\mathrm{best}} = \arg\min_K\{f(X_K^{\mathrm{best}})\}$, and $K = K + 1$.

We give some remarks on Algorithms 4 and 5 to conclude this section.

Firstly, the term $-\left\|X - \frac{1}{K}\sum_{i=1}^{K}\hat{X}_i\right\|_{\mathsf{F}}^2$ plays the role in pushing Algorithms 4 and 5 to find a new permutation matrix which is far away from the average $\frac{1}{K}\sum_{i=1}^{K}\hat{X}_i$ of $\{\hat{X}_i\}_{i=1}^{K}$. More generally, given nonnegative $\{w_i\}_{i=1}^{K}$ satisfying $\sum_{i=1}^{K}w_i = 1$, we can choose $-\|X - \sum_{i=1}^{K}w_i\hat{X}_i\|_{\mathsf{F}}^2$.

Secondly, Theorem 5.2 shows that problem (5.11) with any $\mu \in (0, \frac{c_2}{2n})$ shares the same global solution with problem (1.1). However, (i) since c_2 is generally unknown, it is hard to choose the parameter μ satisfying $\mu < \frac{c_2}{2n}$ and (ii) $\mu \in (0, \frac{c_2}{2n})$ might be too small to effectively improve the performance of Algorithms 4 and 5. Therefore, we initialize μ with a relatively large value and gradually decrease it by setting $\mu = \mu/2$ in the two algorithms. We terminate the two algorithms by checking that if the newly obtained permutation matrix $\hat{X}_K$ has already been explored or not; see line 2 of Algorithms 4 and 5.

6. Numerical results on QAPLIB. In this section, we report numerical results to demonstrate the efficiency and effectiveness of our proposed algorithms for solving QAP (1.2), which is a special and important case of problem (1.1).

We consider 134 instances from QAPLIB [13] except `esc16f` and `tai10b` since the elements of matrix A of `esc16f` are all zero and the best feasible solution of `tai10b` is not provided. All the experiments were performed in OS X 10.10 on an iMac with a 3.2GHz Intel Core i5 Processor with access to 8GB of RAM. We implemented our methods in MATLAB (Release 2014b).

For each problem instance, we scale the matrix A and B as $A := A/\rho_A$ and $B := B/\rho_B$ with $\rho_A = \max_{ij}|A_{ij}|$ and $\rho_B = \max_{ij}|B_{ij}|$. We use the relative gap ratio

$$\mathrm{gap} := \left(\frac{\mathrm{obj} - \mathrm{obj}^*}{\mathrm{obj}^*} \times 100\right)\%$$

to measure the quality of the solution returned by different algorithms, where obj^* is the optimal or best known function value provided by QAPLIB and obj is the function value obtained by each algorithm. Note that the values of obj^* and obj are computed based on the original A and B, and all obj^* are positive integers.

6.1. Implementation details. For Algorithm 2, we choose the initial point $X_0 = \frac{1}{n}\mathbf{1}$, the initial guess $\epsilon_0 = 0.1$, the safeguards $\epsilon_{\min} = 10^{-3}$, $\sigma_{\max} = 10^6$, and the shrinkage parameter for updating ϵ_k as $\gamma = 0.9$. Our experience shows that Algorithm 2 also works on an ϵ_0 larger than 1, smaller shrinkage and safeguard parameters, but the choices of these parameter might affect its efficiency. We set the stopping tolerance $\mathrm{tol} = 10^{-3}$, which guarantees that the returned X_k is sufficiently close to some permutation matrix. For the projected gradient method, we choose the initial stepsize $\alpha_0 = 10^{-3}$, the parameters of nonmonotone linesearch condition (4.8) as $\theta = 10^{-4}$, $\delta = 0.5$, $\eta = 0.85$, the initial guess of the tolerance for solving the subproblems (3.5) as $\tau_0^x = 10^{-3}, \tau_0^f = 10^{-6}$ and the corresponding safeguards as $\tau_{\min}^x = 10^{-5}, \tau_{\min}^f = 10^{-8}$. For problems with $n > 50$, we use MOSEK 7.1[1] to compute the projection onto $\mathcal{D}_n$ if $\max_{ij}|X_{ij}| > 8$, and use dualBB otherwise. For problems with $n \le 50$, we always use MOSEK 7.1 to compute the projection onto $\mathcal{D}_n$. For Algorithms 3 – 5, we choose $K_{\max} = 10$. As for computing projections onto the restricted domain $\mathcal{A}$ in Algorithms 3 and 5, MOSEK 7.1 is always employed no matter how large n is. For Algorithms 4 and 5, we choose $\mu = \min\{0.5, (\overline{\nu}_f - \underline{\nu}_f)/100\}$,

---

[1]Downloadable from `https://www.mosek.com/resources/downloads`.

where $\overline{\nu}_f$ and $\underline{\nu}_f$ are taken as the largest and smallest eigenvalues of $B^\mathsf{T} \otimes A^\mathsf{T} + B \otimes A \in \mathbb{R}^{n^2 \times n^2}$, respectively. The parameter ω in (5.1), which is used in Algorithms 3 and 5, is chosen to be $1 - 0.5\underline{\nu}_f$.

For a given $X \in \mathcal{D}_n$, a locally 2-optimal permutation matrix can be generated by a procedure with two steps: (i) generating a permutation $\hat{X}^{(0)} \in \Pi_n$ by some greedy procedure; (ii) finding a permutation matrix $\hat{X}^{(k)} \in \mathcal{N}_2^{\text{best}}(\hat{X}^{(k)})$ starting from $\hat{X}^{(0)}$. Their details are outlined as follows.

Step (i). Given $X \in \mathcal{D}_n$, we use a fast greedy procedure in the code "LagSA" of [55] to generate $\hat{X}^{(0)} \in \Pi_n$. It computes an approximate solution of the linear assignment problem $\max_{Y \in \Pi_n} \langle X, Y \rangle$ in a greedy way as follows: (a) Calculate a vector $x \in \mathbb{R}^n$ with the i-th element $x_i = \max_{j \in \mathcal{N}} X_{ij}$. (b) Identity a permutation vector $\pi \in \mathbb{R}^n$ such that $x_{\pi_1} \leq x_{\pi_2} \leq \cdots \leq x_{\pi_n}$. (c) Let $\hat{X}^{(0)} := \mathbf{0}$. For each $i \in \mathcal{N}$, compute $k_{\pi_i} = \arg \max_{j \in \mathcal{N} \setminus \{k_{\pi_1}, \ldots, k_{\pi_{i-1}}\}} X_{\pi_i, j}$ with $\{k_{\pi_0}\} = \emptyset$ and set $\hat{X}^{(0)}_{\pi_i, k_{\pi_i}} = 1$. We can also apply the canonical Hungarian algorithm to solve the linear assignment problem. The complexity of the Hungarian algorithm and "LagSA" is $O(n^3)$ and $O(n^2)$, respectively.

Step (ii). Starting from $\hat{X}^{(0)}$, we perform an iterative local 2-neighborhood search to find a locally 2-optimal permutation matrix. More specifically, for $k \geq 0$, we compute $\hat{X}_s^{(k)} \in \arg \min_{Y \in \mathcal{N}_2(\hat{X}^{(k)})} f(Y)$ and update

$$(6.1) \qquad \hat{X}^{(k+1)} = \begin{cases} \hat{X}_s^{(k)}, & \text{if } f(\hat{X}_s^{(k)}) < f(\hat{X}^{(k)}), \\ \hat{X}^{(k)}, & \text{otherwise.} \end{cases}$$

The above procedure is terminated until a locally 2-optimal permutation matrix is found, i.e., $\hat{X}^{(k+1)} = \hat{X}^{(k)}$ for some k.

The above iterative local 2-neighborhood search procedure terminates within $\lceil (f(\hat{X}^{(0)}) - f^*)/c_2 \rceil$ iterations. This is because of the facts $f(\hat{X}^{(k)}) \leq f(\hat{X}^{(k-1)}) - c_2 \leq f(\hat{X}^{(0)}) - kc_2$ (i.e., (5.7)) and $f(\hat{X}^{(k)}) \geq f^*$. The cost of computing $\hat{X}_s^{(k)}$ is cheap for QAP. Suppose the procedure terminates within k_0 iterations. Then the total computational cost is $4n^3 + \mathcal{O}(k_0 n^2)$; see [43, 50, 51] for detailed information.

6.2. Comparisons of L_p and L_2 regularization algorithms. In this subsection, we compare our proposed L_p regularization algorithms (Algorithm 2) using $p = 0.25, 0.5, 0.75$ with the L_2 regularization algorithm LagSA [55]. We also test a variant of LagSA, where subproblem (1.3) is solved by the projected gradient method, and name the variant as LagSA-BB. Note that LagSA and LagSA-BB do not use the local 2-neighborhood search or the rounding techniques[2]. Therefore, for fair of comparisons, we use a basic version of our proposed algorithm, named as L_p-bs, which does not use the local 2-neighborhood search technique neither.

We summarize the numerical results in Table 1, where the number denotes how many problem instances can be solved by the corresponding algorithm within the given gap. For instance, Table 1 shows that LagSA can solve 14 instances within gap 0.0 (to global optimality) and 99 instances within gap 5.0. It can be seen from Table 1 that our proposed L_p regularization algorithms generally perform better than LagSA and LagSA-BB and $L_{0.75}$-bs performs the best (among all the five algorithms). It can also be seen from Table 1 that LagSA-BB performs better than LagSA by using the

---

[2]The local 2-neighborhood search and the rounding techniques can be used to improve the performance of the two algorithms.

projected gradient method to solve L_2 regularization subproblem (1.3) instead of the (truncated) Frank-Wolfe method.

Table 1

Comparison of gap levels of the five algorithms on 134 instances in QAPLIB

gap $\leq$ %	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	2.0	3.0	4.0	5.0
LagSA	14	21	29	41	42	44	48	51	53	57	61	80	91	95	99
LagSA-BB	26	39	50	56	58	62	65	68	73	78	80	96	105	111	116
$L_{0.25}$-bs	22	37	46	57	58	59	62	68	71	74	75	90	102	105	109
$L_{0.5}$-bs	24	37	50	56	59	60	64	70	76	77	77	89	107	111	113
$L_{0.75}$-bs	**27**	**44**	**52**	**65**	**67**	**69**	**73**	**74**	**80**	**82**	**84**	**98**	**108**	**114**	**115**

To make the comparison clearer, we make a pairwise comparison of the five algorithms as used in [18]. Given a problem and a collection of the five algorithms, algorithm i is called the winner if the gap obtained by the algorithm is the smallest among all algorithms. We summarize the comparison results in Table 2. The second line of Table 2 reports the winners of each algorithm among all the five algorithms for solving the 134 instances. In particular, Table 2 shows that $L_{0.75}$-bs wins 62 of the 134 instances. Starting from the third line, we perform pairwise comparisons of the algorithms. It follows from the last line of this table that $L_{0.75}$-bs shows its superiority over the other algorithms. In summary, $L_{0.75}$-bs performs the best among all the five algorithms. Therefore, we set $p = 0.75$ in our proposed L_p regularization algorithms in the subsequent numerical experiments. It should be pointed out that the performance of $L_{0.75}$ is much better than that of $L_{0.75}$-bs due to the local 2-neighborhood search and rounding techniques. For instance, $L_{0.75}$ can solve 51 problems to a zero gap while $L_{0.75}$-bs can only solve 27 problems to a zero gap.

Table 2

Pairwise comparison of the five algorithms

methods	LagSA	LagSA-BB	$L_{0.25}$-bs	$L_{0.5}$-bs	$L_{0.75}$-bs
# of winners	28	53	47	49	**62**
LagSA : i	—	47 : 107	53 : 92	49 : 98	46 : **101**
LagSA-BB : i	107 : 47	—	88 : 66	79 : 76	71 : **88**
$L_{0.25}$-bs : i	92 : 53	66 : 88	—	85 : 96	69 : **99**
$L_{0.5}$-bs : i	98 : 49	76 : 79	96 : 85	—	86 : **93**
$L_{0.75}$-bs : i	**101** : 46	**88** : 71	**99** : 69	**93** : 86	—

6.3. Comparisons of $L_{0.75}$-negProx, $L_{0.75}$-CP, and $L_{0.75}$-CP-negProx. Numerical results of $L_{0.75}$-negProx, $L_{0.75}$-CP, and $L_{0.75}$-CP-negProx are presented in Tables 3 and 4. For the sake of saving space, we only present in Table 3 the results on problem instances such that the best gap returned by $L_{0.75}$-negProx, $L_{0.75}$-CP, and $L_{0.75}$-CP-negProx is zero.

In the tables in this and subsequent subsections, "Rgap" denotes the average relative gap of 100,000 random permutation matrices, which can illustrate the hardness of the corresponding problem to some extent; "gap" denotes the relative gap achieved by our proposed algorithm; and "time" denotes the running time in seconds of different algorithms. We put the best gaps among all the gaps returned by different algorithms in bold. In the last line of each table, we summarize the total number of instances N_{best} such that the gap returned by each algorithm is the best among the gaps returned by all algorithms.

It can be seen from Tables 3 and 4 that the performance of $L_{0.75}$-negProx and $L_{0.75}$-CP are better than that of $L_{0.75}$ on most of problem instances. This shows that both the negative proximal point technique and the cutting plane technique are helpful in improving the performance of $L_{0.75}$. It can also be seen from Tables 3 and 4 that $L_{0.75}$-negProx (which employs the negative proximal point technique) performs much better than $L_{0.75}$-CP (which uses the cutting plane technique). In particular, $L_{0.75}$-negProx reduces the gaps of 17 problem instances to zero and $L_{0.75}$-CP reduces the gaps of only 5 problem instances to zero. This demonstrates that our proposed negative proximal point technique is more effective in improving the performance of $L_{0.75}$.

It can also be observed from the two tables that $L_{0.75}$-CP-negProx (which employs both the cutting plane technique and the negative proximal point technique) performs slightly better in terms of the solution quality than $L_{0.75}$-negProx. However, $L_{0.75}$-negProx is faster than $L_{0.75}$-CP-negProx since the projections in $L_{0.75}$-CP-negProx are more difficult to compute than those in $L_{0.75}$-negProx.

TABLE 3

Numerical results of $L_{0.75}$-negProx, $L_{0.75}$-CP, $L_{0.75}$-CP-negProx for 19 problem instances with $n < 80$ in QAPLIB

Problem		$L_{0.75}$		$L_{0.75}$-negProx		$L_{0.75}$-CP		$L_{0.75}$-CP-negProx	
name	Rgap	gap	time	gap	time	gap	time	gap	time
bur26a	9.6	0.1212	0.8	**0.0000**	2.6	0.0889	3.3	**0.0000**	3.0
bur26b	10.4	0.1769	1.0	**0.0000**	3.0	0.1716	4.6	**0.0000**	3.1
bur26d	10.2	0.0020	0.6	**0.0000**	3.6	0.0020	3.3	**0.0000**	4.3
bur26e	10.3	0.0091	0.7	**0.0000**	2.2	**0.0000**	4.0	**0.0000**	2.5
chr15c	546.6	30.9554	0.4	20.1178	3.9	30.9554	2.0	**0.0000**	3.6
chr20c	656.5	18.2152	0.4	**0.0000**	3.5	18.2152	1.7	**0.0000**	4.3
esc32b	171.0	9.5238	1.8	**0.0000**	10.5	**0.0000**	19.9	**0.0000**	10.6
had20	12.2	0.0289	0.5	**0.0000**	2.2	0.0867	2.5	**0.0000**	2.6
kra30a	51.5	0.8999	1.1	1.3498	10.2	1.5298	12.4	**0.0000**	10.8
lipa20a	7.0	2.0092	0.1	**0.0000**	2.3	1.8192	0.6	**0.0000**	4.6
nug14	34.5	0.1972	0.3	**0.0000**	1.7	0.1972	1.3	**0.0000**	1.9
nug18	32.9	0.4145	0.5	**0.0000**	4.3	0.4145	2.0	**0.0000**	5.6
nug28	34.2	0.1549	1.3	**0.0000**	5.5	**0.0000**	6.0	**0.0000**	6.7
scr12	89.2	2.3368	0.1	**0.0000**	1.0	**0.0000**	0.9	**0.0000**	1.4
scr20	105.7	0.6525	0.3	**0.0000**	4.0	0.0254	2.4	**0.0000**	2.5
ste36b	439.0	4.5672	0.7	**0.0000**	4.0	1.1734	3.1	1.5519	8.3
tai20b	163.6	0.4526	0.3	**0.0000**	2.3	0.4526	1.4	**0.0000**	2.8
tai30a	21.0	1.7887	2.2	**0.0000**	7.7	1.7887	10.6	**0.0000**	9.0
tai64c	59.2	0.0926	5.3	**0.0000**	24.2	**0.0000**	12.4	**0.0000**	21.1
N_{best}		0		17		5		18	

6.4. Comparisons of $L_{0.75}$, Ro-TS, and their hybrid $L_{0.75}$-Ro-TS. To illustrate the efficiency and effectiveness of our proposed $L_{0.75}$ regularization algorithm (Algorithm 2 with $p = 0.75$), we compare it with one of the state-of-the-art heuristics Ro-TS[3] in this subsection. Ro-TS implements the robust taboo search for solving QAP. For QAP problem instances with $n < 80$, we set the parameters in Ro-TS to be `nr_iterations = 1500*n; nr_resolutions = 10`; for instances with $n \geq 80$, we set `nr_iterations = 500*n; nr_resolutions = 10`. We also test a hybrid, named as $L_{0.75}$-Ro-TS, of $L_{0.75}$ and Ro-TS by using the permutation matrix returned by

---

[3]Downloadable from `http://mistic.heig-vd.ch/taillard/codes.dir/tabou_qap2.c`

Table 4

Numerical results of $L_{0.75}$-negProx, $L_{0.75}$-CP, $L_{0.75}$-CP-negProx for 14 problem instances with $n \geq 80$ in QAPLIB

Problem		$L_{0.75}$		$L_{0.75}$-negProx		$L_{0.75}$-CP		$L_{0.75}$-CP-negProx	
name	Rgap	gap	time	gap	time	gap	time	gap	time
lipa80a	2.1	0.7540	0.4	0.7255	3.6	**0.7168**	2.8	0.7255	6.3
lipa90a	1.9	0.7179	0.9	**0.6827**	7.7	0.6960	3.2	0.6893	4.3
sko100b	17.4	0.0858	13.9	0.0832	141.4	**0.0793**	177.9	0.0858	127.8
sko100d	17.5	0.1712	13.9	**0.1043**	97.0	0.1712	103.9	**0.1043**	157.7
sko100e	18.2	0.0134	14.0	**0.0000**	88.2	0.0094	95.7	**0.0000**	117.7
sko100f	17.3	0.0550	12.3	**0.0523**	199.9	0.0550	95.7	0.0550	92.6
sko81	19.2	0.1143	12.2	**0.0813**	71.0	0.1143	50.4	**0.0813**	94.1
tai100b	50.9	0.3800	13.7	**0.2237**	62.8	0.3789	76.9	**0.2237**	90.7
tai150b	30.6	0.5098	86.2	**0.1422**	221.5	0.4995	836.6	0.1462	330.2
tai256c	19.8	0.2610	24.6	0.1112	311.0	**0.1041**	493.2	0.1581	277.3
tai80a	15.6	0.6904	20.0	**0.4734**	111.8	0.6904	93.6	**0.4734**	129.0
tai80b	51.9	0.0378	6.7	0.0378	36.0	**0.0336**	53.4	0.0378	53.6
tho150	20.6	0.1500	53.4	0.1302	225.7	0.1302	377.5	**0.1223**	478.2
wil100	9.8	0.0146	21.0	0.0146	91.7	**0.0132**	177.9	**0.0132**	277.1
N_{best}		0		8		5		7	

$L_{0.75}$ as the input of Ro-TS. The combination of $L_{0.75}$-negProx, $L_{0.75}$-CP, and $L_{0.75}$-CP-negProx with Ro-TS can improve the performance similarly but their results are not shown for the simplicity of presentation.

In the tables in this subsection, the numbers "min gap", "mean gap", and "max gap" denote the minimum, mean, and maximum gap of Ro-TS among 10 runs, respectively; "nfe" denotes the total number of the objective function evaluations of (3.5) with $p = 0.75$; "time" denotes the mean running time in seconds of different algorithms. Note that the implementation of Ro-TS is in the C language, while our proposed algorithm is implemented in the MATLAB environment. Hence, the comparison of the running time is more favorable for Ro-TS.

In Table 5, we present numerical results on 51 problem instances in QAPLIB for which our proposed algorithm $L_{0.75}$ is able to achieve the optimal value or the best known upper bound. It can be observed from Table 5 that the time used by our algorithm is comparable to that of Ro-TS. Particularly, for the problem instance "esc128", our algorithm is twice faster than Ro-TS.

Table 5

Numerical results of $L_{0.75}$ and Ro-TS on 51 problem instances in QAPLIB

Problem			$L_{0.75}$			Ro-TS, 10 runs	
name	obj*	Rgap	gap	time	nfe	gap	time
						(min, mean, max)	
bur26c	5426795	9.5	**0.0000**	0.7	161	(**0.0000**, 0.0004, 0.0036)	1.0
bur26f	3782044	11.3	**0.0000**	1.2	246	(**0.0000**, 0.0001, 0.0006)	1.0
bur26g	10117172	9.9	**0.0000**	0.6	147	(**0.0000**, 0.0000, 0.0000)	1.0
bur26h	7098658	10.9	**0.0000**	0.6	149	(**0.0000**, 0.0003, 0.0035)	1.0
chr12a	9552	372.5	**0.0000**	0.3	185	(**0.0000**, 0.0000, 0.0000)	0.1
chr12b	9742	363.4	**0.0000**	0.2	152	(**0.0000**, 0.0000, 0.0000)	0.1
chr18b	1534	199.9	**0.0000**	1.0	366	(**0.0000**, 0.0000, 0.0000)	0.3
esc16a	68	63.3	**0.0000**	0.4	199	(**0.0000**, 0.0000, 0.0000)	0.2
esc16b	292	7.9	**0.0000**	3.8	269	(**0.0000**, 0.0000, 0.0000)	0.2
esc16c	160	55.8	**0.0000**	0.4	195	(**0.0000**, 0.0000, 0.0000)	0.2
esc16d	16	226.0	**0.0000**	0.7	308	(**0.0000**, 0.0000, 0.0000)	0.2
						Continued on next page	

TABLE 5
continued from the previous page

Problem			$L_{0.75}$			Ro-TS, 10 runs	
name	obj*	Rgap	gap	time	nfe	gap	time
						(min, mean, max)	
esc16e	28	118.5	**0.0000**	0.3	158	(**0.0000**, 0.0000, 0.0000)	0.2
esc16g	26	152.6	**0.0000**	0.3	120	(**0.0000**, 0.0000, 0.0000)	0.2
esc16h	996	41.6	**0.0000**	0.4	187	(**0.0000**, 0.0000, 0.0000)	0.2
esc16i	14	288.7	**0.0000**	0.3	132	(**0.0000**, 0.0000, 0.0000)	0.2
esc16j	8	268.0	**0.0000**	0.3	160	(**0.0000**, 0.0000, 0.0000)	0.2
esc32c	642	45.3	**0.0000**	1.0	227	(**0.0000**, 0.0000, 0.0000)	1.8
esc32d	200	80.2	**0.0000**	1.0	237	(**0.0000**, 0.0000, 0.0000)	1.8
esc32e	2	2428.2	**0.0000**	0.8	187	(**0.0000**, 0.0000, 0.0000)	1.8
esc32g	6	637.6	**0.0000**	0.7	151	(**0.0000**, 0.0000, 0.0000)	1.8
esc32h	438	53.7	**0.0000**	1.1	242	(**0.0000**, 0.0000, 0.0000)	1.8
esc64a	116	140.1	**0.0000**	5.7	443	(**0.0000**, 0.0000, 0.0000)	14.9
had12	1652	14.3	**0.0000**	0.3	181	(**0.0000**, 0.0000, 0.0000)	0.1
had14	2724	15.7	**0.0000**	0.2	128	(**0.0000**, 0.0000, 0.0000)	0.1
had16	3720	13.6	**0.0000**	0.4	171	(**0.0000**, 0.0000, 0.0000)	0.2
had18	5358	11.8	**0.0000**	0.4	177	(**0.0000**, 0.0000, 0.0000)	0.3
kra32	88700	54.6	**0.0000**	1.0	192	(**0.0000**, 0.0000, 0.0000)	1.8
lipa20b	27076	31.3	**0.0000**	0.3	131	(**0.0000**, 0.0000, 0.0000)	0.4
lipa30b	151426	29.2	**0.0000**	0.4	108	(**0.0000**, 0.0000, 0.0000)	1.5
lipa40b	476581	30.4	**0.0000**	0.5	83	(**0.0000**, 0.0000, 0.0000)	3.6
lipa50b	1210244	28.9	**0.0000**	0.9	94	(**0.0000**, 0.0000, 0.0000)	7.1
lipa60b	2520135	29.9	**0.0000**	1.6	113	(**0.0000**, 0.0000, 0.0000)	12.3
lipa70b	4603200	30.1	**0.0000**	1.5	84	(**0.0000**, 0.0000, 0.0000)	19.6
nug12	578	40.5	**0.0000**	0.3	191	(**0.0000**, 0.0000, 0.0000)	0.1
nug15	1150	37.8	**0.0000**	0.4	220	(**0.0000**, 0.0000, 0.0000)	0.2
nug16b	1240	39.3	**0.0000**	0.3	152	(**0.0000**, 0.0000, 0.0000)	0.2
nug20	2570	32.6	**0.0000**	0.6	205	(**0.0000**, 0.0000, 0.0000)	0.4
nug21	2438	40.3	**0.0000**	0.5	162	(**0.0000**, 0.0000, 0.0000)	0.5
nug22	3596	43.2	**0.0000**	0.6	186	(**0.0000**, 0.0000, 0.0000)	0.6
nug24	3488	36.7	**0.0000**	0.7	191	(**0.0000**, 0.0000, 0.0000)	0.8
nug25	3744	33.7	**0.0000**	0.6	158	(**0.0000**, 0.0000, 0.0000)	0.9
nug27	5234	36.2	**0.0000**	0.6	147	(**0.0000**, 0.0000, 0.0000)	1.1
rou15	354210	32.2	**0.0000**	0.9	544	(**0.0000**, 0.0000, 0.0000)	0.2
scr15	51140	99.5	**0.0000**	0.3	149	(**0.0000**, 0.0000, 0.0000)	0.2
tai10a	135028	38.4	**0.0000**	0.3	177	(**0.0000**, 0.0000, 0.0000)	0.1
tai12a	224416	39.3	**0.0000**	0.3	159	(**0.0000**, 0.0000, 0.0000)	0.1
tai12b	39464925	111.8	**0.0000**	0.3	178	(**0.0000**, 0.0000, 0.0000)	0.1
tai15b	51765268	676.6	**0.0000**	0.2	70	(**0.0000**, 0.0000, 0.0000)	0.2
esc128	64	397.6	**0.0000**	19.8	390	(**0.0000**, 13.4375, 28.1250)	41.5
lipa80b	7763962	30.8	**0.0000**	3.1	163	(**0.0000**, 0.0000, 0.0000)	9.8
lipa90b	12490441	30.5	**0.0000**	2.2	84	(**0.0000**, 0.0000, 0.0000)	14.1
N_{best}			51			51	

There are still 65 problem instances with $n < 80$, for which $L_{0.75}$ cannot achieve the optimal value or the best known upper bound. For the sake of saving space, we only report in Table 6 the results on problem instances, for which the maximal gap returned by $L_{0.75}$-Ro-TS or Ro-TS in 10 runs is greater than zero. For these problems, although $L_{0.75}$ does not perform as well as Ro-TS, the hybrid $L_{0.75}$-Ro-TS can achieve a satisfactory performance (compared with Ro-TS). Note that the time of $L_{0.75}$-Ro-TS does not include the time of $L_{0.75}$.

Table 6

Numerical results of $L_{0.75}$, $L_{0.75}$-Ro-TS, and Ro-TS on 41 problem instances with $n < 80$ in QAPLIB

Problem		$L_{0.75}$			$L_{0.75}$-Ro-TS, 10 runs		Ro-TS, 10 runs	
name	Rgap	gap	time	nfe	gap	time	gap	time
					(min, mean, max)		(min, mean, max)	
bur26a	9.6	0.1212	0.8	188	(**0.0000**, 0.0137, 0.0845)	1.0	(**0.0000**, 0.0000, 0.0000)	1.0
bur26b	10.4	0.1769	1.0	259	(**0.0000**, 0.0022, 0.0189)	1.0	(**0.0000**, 0.0000, 0.0000)	1.0
bur26d	10.2	0.0020	0.6	152	(**0.0000**, 0.0001, 0.0008)	1.0	(**0.0000**, 0.0006, 0.0021)	1.0
chr15a	520.4	0.4042	0.5	276	(**0.0000**, 0.0404, 0.4042)	0.2	(**0.0000**, 0.0829, 0.8286)	0.2
chr15b	668.1	17.6971	0.3	136	(**0.0000**, 0.5507, 2.7534)	0.2	(**0.0000**, 0.0000, 0.0000)	0.2
chr15c	546.6	30.9554	0.4	243	(**0.0000**, 0.0000, 0.0000)	0.2	(**0.0000**, 0.6355, 6.3552)	0.2
chr18a	600.0	11.7679	0.5	221	(**0.0000**, 0.0000, 0.0000)	0.3	(**0.0000**, 0.0396, 0.3965)	0.3
chr20a	388.5	12.5000	0.8	274	(**0.0000**, 2.4453, 6.3869)	0.4	(**0.0000**, 1.8887, 5.4745)	0.4
chr20b	366.0	13.8381	0.7	271	(3.0461, 4.4822, 5.9182)	0.4	(**0.0000**, 4.3777, 6.6144)	0.4
chr20c	656.5	18.2152	0.4	141	(**0.0000**, 2.9359, 10.4370)	0.4	(**0.0000**, 1.8894, 4.7235)	0.4
chr22a	153.8	5.7180	0.5	163	(**0.0000**, 0.7797, 1.8194)	0.6	(**0.0000**, 0.6790, 2.1767)	0.6
chr22b	152.3	5.2954	0.6	192	(**0.0000**, 0.9106, 1.8405)	0.6	(0.7104, 1.3497, 1.8082)	0.6
chr25a	423.8	25.2371	0.9	279	(**0.0000**, 7.1286, 11.6965)	0.9	(2.3182, 5.8061, 10.6428)	0.9
esc32a	233.2	1.5385	3.1	440	(**0.0000**, 0.6154, 1.5385)	1.8	(**0.0000**, 0.0000, 0.0000)	1.8
kra30a	51.5	0.8999	1.1	222	(**0.0000**, 0.0000, 0.0000)	1.5	(**0.0000**, 0.3690, 1.3386)	1.5
kra30b	49.9	0.3172	1.3	274	(**0.0000**, 0.0077, 0.0766)	1.5	(**0.0000**, 0.0077, 0.0766)	1.5
lipa60a	2.7	0.9858	0.3	38	(**0.0000**, 0.0671, 0.6706)	12.3	(**0.0000**, 0.0000, 0.0000)	12.3
lipa70a	2.4	0.8807	0.3	36	(**0.0000**, 0.0596, 0.5956)	19.7	(**0.0000**, 0.0590, 0.5903)	19.6
nug30	32.8	0.0653	1.0	205	(**0.0000**, 0.0000, 0.0000)	1.5	(**0.0000**, 0.0131, 0.0653)	1.5
rou20	25.5	0.2021	0.9	362	(**0.0000**, 0.0233, 0.1778)	0.4	(**0.0000**, 0.0074, 0.0193)	0.4
sko42	26.9	0.2150	2.6	352	(**0.0000**, 0.0076, 0.0253)	4.2	(**0.0000**, 0.0051, 0.0253)	4.2
sko49	24.2	0.1197	2.9	284	(**0.0000**, 0.0735, 0.1197)	6.7	(0.0086, 0.0804, 0.1368)	6.7
sko56	23.8	0.0348	5.2	327	(0.0348, 0.0348, 0.0348)	10.0	(**0.0116**, 0.0818, 0.2264)	10.0
sko64	21.3	0.0206	5.7	321	(0.0124, 0.0198, 0.0206)	15.0	(**0.0041**, 0.0499, 0.1361)	15.0
sko72	20.3	0.0664	7.7	380	(0.0211, 0.0580, 0.0664)	21.3	(**0.0060**, 0.1210, 0.3139)	21.4
ste36a	138.8	2.2045	1.0	141	(**0.0000**, 0.0042, 0.0420)	2.6	(**0.0000**, 0.0945, 0.2939)	2.6
ste36c	127.7	2.3769	1.2	183	(**0.0000**, 0.0011, 0.0112)	2.6	(**0.0000**, 0.0188, 0.1883)	2.6
tai20a	27.5	0.9763	1.1	488	(**0.0000**, 0.1852, 0.4697)	0.4	(**0.0000**, 0.1825, 0.3042)	0.4
tai25a	23.9	2.1781	1.5	498	(**0.0000**, 0.0884, 0.3658)	0.9	(**0.0000**, 0.3898, 0.8721)	0.9
tai30a	21.0	1.7887	2.2	565	(**0.0000**, 0.3436, 0.7806)	1.5	(**0.0000**, 0.3254, 0.6595)	1.5
tai30b	106.7	2.5101	0.9	183	(**0.0000**, 0.0406, 0.1456)	1.5	(**0.0000**, 0.0782, 0.2637)	1.5
tai35a	21.1	2.3491	3.9	801	(0.1642, 0.5192, 0.8617)	2.4	(**0.0671**, 0.5541, 0.9723)	2.4
tai35b	82.1	4.7863	1.3	217	(**0.0000**, 0.0569, 0.2153)	2.4	(**0.0000**, 0.1596, 0.9763)	2.4
tai40a	20.6	1.6882	3.1	494	(**0.3336**, 0.6905, 0.9928)	3.6	(0.3753, 0.7424, 1.0622)	3.6
tai40b	77.8	0.0051	1.6	202	(**0.0000**, 0.0000, 0.0000)	3.6	(**0.0000**, 0.2639, 2.1123)	3.6
tai50a	19.5	1.1252	6.2	715	(**0.9638**, 1.0725, 1.1252)	7.1	(0.9764, 1.2008, 1.4282)	7.1
tai50b	72.0	0.5660	5.1	591	(0.0106, 0.3520, 0.5660)	7.1	(**0.0029**, 0.1101, 0.4161)	7.1
tai60a	18.2	1.0398	8.5	476	(**0.8591**, 0.9631, 1.0398)	12.3	(0.9338, 1.1456, 1.3461)	12.3
tai60b	66.0	0.0986	5.6	478	(**0.0000**, 0.0732, 0.0986)	12.3	(0.0010, 0.2838, 0.9002)	12.3
tho40	42.0	0.2545	2.2	292	(**0.0000**, 0.0523, 0.1172)	3.6	(**0.0000**, 0.0369, 0.0632)	3.6
wil50	13.7	0.1311	6.9	799	(**0.0000**, 0.0238, 0.0533)	7.1	(**0.0000**, 0.0315, 0.1024)	7.1
N_{best}		0			35		34	

In Table 7, we present numerical results on 21 problem instances with $n \geq 80$. Table 7 shows that our proposed $L_{0.75}$ regularization algorithm significantly outperforms Ro-TS in terms of both the solution quality and the speed for solving these problems. More specifically, our algorithm can find a permutation matrix whose gap is less than 0.8% for all 21 problem instances and whose gap is less than 0.1% for 11 problem instances. In contrast, the corresponding two numbers for Ro-TS (min) are 18 and 9 and for Ro-TS (mean) are only 15 and 2. For the largest problem instance "tai256c", our proposed algorithm is able to find a solution with gap 0.2610% while

the best gap returned by Ro-TS (among 10 runs) is 0.3169%, and our algorithm is 15 times faster than Ro-TS.

TABLE 7
Numerical results of $L_{0.75}$, $L_{0.75}$-Ro-TS, Ro-TS on 21 problem instances with $n \geq 80$ in QAPLIB

Problem		$L_{0.75}$			$L_{0.75}$-Ro-TS, 10 runs		Ro-TS, 10 runs	
name	Rgap	gap	time	nfe	gap	time	gap (%)	time
					(min, mean, max)		(min, mean, max)	
esc128	397.6	**0.0000**	19.8	390	(**0.0000**, 0.0000, 0.0000)	0.0	(**0.0000**, 13.4375, 28.1250)	41.5
lipa80a	2.1	0.7540	0.4	37	(**0.0000**, 0.4808, 0.5593)	9.8	(0.5016, 0.5351, 0.5644)	9.8
lipa80b	30.8	**0.0000**	3.1	163	(**0.0000**, 0.0000, 0.0000)	0.0	(**0.0000**, 0.0000, 0.0000)	9.8
lipa90a	1.9	0.7179	0.9	37	(0.4689, 0.4872, 0.5025)	14.0	(**0.4453**, 0.4734, 0.4980)	14.1
lipa90b	30.5	**0.0000**	2.2	84	(**0.0000**, 0.0000, 0.0000)	0.0	(**0.0000**, 0.0000, 0.0000)	14.1
sko100a	17.4	**0.0645**	14.5	396	(0.0645, 0.0645, 0.0645)	19.4	(0.0961, 0.1962, 0.5987)	19.4
sko100b	17.4	0.0858	13.9	371	(**0.0195**, 0.0489, 0.0819)	19.4	(0.0871, 0.2102, 0.3808)	19.4
sko100c	18.0	0.0203	16.8	464	(**0.0108**, 0.0185, 0.0203)	19.3	(0.0555, 0.2252, 0.4355)	19.4
sko100d	17.5	0.1712	13.9	381	(0.0949, 0.1296, 0.1645)	19.4	(**0.0388**, 0.2434, 0.4279)	19.5
sko100e	18.2	**0.0134**	14.0	298	(0.0134, 0.0134, 0.0134)	19.3	(0.1824, 0.4008, 0.6195)	19.4
sko100f	17.3	0.0550	12.3	328	(**0.0550**, 0.0550, 0.0550)	19.5	(0.1342, 0.3073, 0.5569)	19.4
sko81	19.2	0.1143	12.2	535	(**0.0440**, 0.0963, 0.1143)	10.2	(0.1165, 0.2024, 0.3143)	10.2
sko90	18.5	0.0554	14.7	526	(0.0554, 0.0554, 0.0554)	14.1	(**0.0450**, 0.2612, 0.3653)	14.1
tai100a	14.4	**0.4131**	33.8	968	(0.4131, 0.4131, 0.4131)	19.4	(0.9459, 1.1252, 1.3004)	19.4
tai100b	50.9	0.3800	13.7	400	(**0.3701**, 0.3786, 0.3800)	19.4	(0.4035, 1.1940, 2.8654)	19.4
tai150b	30.6	0.5098	86.2	1220	(**0.4858**, 0.4899, 0.4926)	67.1	(1.8733, 3.4037, 4.4771)	66.9
tai256c	19.8	**0.2610**	24.6	191	(0.2610, 0.2610, 0.2610)	390.3	(0.3169, 0.4097, 0.5283)	387.2
tai80a	15.6	0.6904	20.0	894	(**0.5827**, 0.6571, 0.6904)	9.8	(1.0769, 1.2615, 1.4734)	9.8
tai80b	51.9	0.0378	6.7	329	(**0.0283**, 0.0283, 0.0283)	9.8	(0.0379, 0.9246, 2.0504)	9.8
tho150	20.6	**0.1500**	53.4	679	(0.1500, 0.1500, 0.1500)	67.0	(0.2551, 0.6223, 1.0213)	67.1
wil100	9.8	0.0146	21.0	542	(**0.0110**, 0.0134, 0.0139)	19.5	(0.1267, 0.1762, 0.2740)	19.4
N_{best}		8			13		6	

7. Application in the bandwidth minimization problem. In this section, we apply Algorithm 2 to solve the bandwidth minimization (BM) problem. It is NP-hard [46] and has many applications in different fields such as sparse matrix computations, circuit design, and VLSI layout [16, 34]. For a real symmetric matrix $A \in \mathbb{R}^{n \times n}$, its bandwidth is defined as $b(A) = \max_{A_{ij} \neq 0} |i - j|$. The BM problem is to find a permutation matrix X such that the matrix $X^{\mathsf{T}}AX$ takes the minimum bandwidth. Define $\bar{A}_{ij} = 1$ if $A_{ij} \neq 0$ and $\bar{A}_{ij} = 0$ otherwise. Clearly, $X^{\mathsf{T}}\bar{A}X$ has the same bandwidth with $X^{\mathsf{T}}AX$. Let $B_m \in \mathbb{R}^{n \times n}$ be the symmetric Toeplitz matrix with $(B_m)_{ij} = \max\{|i - j| - m, 0\}$ and define

$$(7.1) \qquad h(m) = \min_{X \in \Pi_n} \operatorname{tr}(X^{\mathsf{T}}\bar{A}XB_m).$$

Then, $h(m) = 0$ if and only if the bandwidth of $X^{\mathsf{T}}\bar{A}X$ is at most m. Therefore, the above BM problem for a given matrix A can be formulated as the problem of finding the smallest nonnegative integer root of the equation $h(m) = 0$, namely,

$$(7.2) \qquad \min \ m, \quad \text{s.t.} \quad h(m) = 0.$$

The BM problem can also be defined for a given graph. More specifically, for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \ldots, v_n\}$ being the vertices and $\mathcal{E}$ being the edges, the BM problem is to find a bijection $\phi : \mathcal{V} \to \mathcal{N}$ such that the number $\max_{(i,j) \in E} |\phi(v_i) - \phi(v_j)|$ is minimal. Denote the adjacency matrix of $\mathcal{G}$ by $A_{\mathcal{G}}$. Then the BM problem for

the graph $\mathcal{G}$ is equivalent to the BM problem for the matrix $A_{\mathcal{G}}$. For more discussions on the BM problem for graphs and matrices, please refer to [16, 53].

Let m^* be the unique solution of (7.2). By the definition of $h(m)$, we have that $h(i) > h(j)$ for $0 \leq i < j \leq m^*$ and $h(i) = 0$ for $i \geq m^*$. Moreover, it is trivial that $h(n-1) = 0$. Based on the above properties of $h(m)$, we invoke the bisection procedure to solve (7.2), where the subproblem (7.1) is solved by Algorithm 2. The resulting method is named as Bi-L_p and described in Algorithm 6.

Algorithm 6: An Bi-L_p algorithm for BM problem (7.2).

1 Set $\underline{m} = 0, \overline{m} = n - 1$ and the integer $m_0 \in (1, n-1)$, $k = 0$.
2 **while** $\overline{m} - \underline{m} > 1$ **do**
3   Compute $h(m_k)$ by Algorithm 2.
4   **if** $h(m_k) > 0$ **then** $\underline{m} = m_k$;
5   **if** $h(m_k) = 0$ **then** $\overline{m} = m_k$;
6   Set $m_k = \frac{1}{2}\lceil \underline{m} + \overline{m} \rceil$ and $k = k + 1$.

Theoretically, if problem (7.1) is solved exactly, the above bisection procedure will also solve the BM problem globally. Even if problem (7.1) is only approximately solved (i.e., the $h(m_k)$ obtained is only an upper bound), Algorithm 6 would still yield an upper bound on the minimum bandwidth. We can also invoke Algorithms 3 – 5 or other solvers for QAP to compute $h(m_k)$. For simplicity, we only use Algorithm 2 with $p = 0.75$ in Algorithm 6.

We compare the performance of Algorithm 6 with that of three other solvers, including the MATLAB built-in function "symrcm" which implements the reverse Cuthill-McKee (rCM) algorithm [28], the code "imprRevCMcK" (irCM) which is an improved rCM algorithm proposed in [53], and "Bi-RoTS" which utilizes the same algorithm framework as Algorithm 6 but uses Ro-TS to compute $h(m_k)$. The initial bandwidth guess m_0 in Bi-$L_{0.75}$ and Bi-RoTS is set to be the minimal bandwidth returned by symrcm. We replace the implementation of rCM in irCM by symrcm, which can improve the performance of rCM based on our experience. The method irCM consists of a number of independent runs, which is set to be 2000 in our tests. Each run of irCM first generates a random ordering of the vertices of the graph, then performs the rCM algorithm and the improvement procedure developed in [53]. In order to investigate the impact of this number, we run irCM until it takes twice as much CPU time as Bi-$L_{0.75}$ on each test instance. We denote the resulting method as irCM+. For some other methods, one can refer to [44] and references therein.

The numerical results for three different kinds of bandwidth minimization problems are reported in Tables 8 – 10. In these tables, "bw$^{\mathrm{r}}$" denotes the upper bound of the minimal bandwidth returned by symrcm and "bw" denotes the upper bound of the minimal bandwidth returned by the other methods. The running time is counted in seconds. Since symrcm is fast, we do not report its running time in these tables. The term "nrun" denotes the total number of independent runs of irCM+. For each algorithm, the term "N_{best}" denotes the total number of problem instances when the upper bound of the minimal bandwidth provided by this algorithm is the smallest among the bounds provided by all the algorithms. In Table 9, "k" denotes the minimal bandwidth, and we use "$\overline{}$" to denote the average values of each term over 10 instances.

Table 8 reports the results on several symmetric graphs, including the Hamming graph, the 3-dimensional generalized Hamming graph, the Johnson graph and the Kneser graph. For details on these graphs, one can refer to [53]. Table 9 reports

TABLE 8

Numerical results on the upper bound for the bandwidth of a few symmetric graphs

Problem			Bi-$L_{0.75}$		irCM		irCM+		Bi-RoTS	
name	n	bw^r	bw	time	bw	time	bw	nrun	bw	time
$H(3,5)$	125	76	**60**	49.4	**60**	6.0	**60**	32795	64	197.7
$H(3,6)$	216	145	**101**	281.9	**101**	72.4	**101**	15533	113	860.3
$H(4,3)$	81	**35**	**35**	13.6	**35**	0.3	**35**	209742	**35**	70.3
$H(4,4)$	256	114	**113**	329.0	**113**	2.0	**113**	655738	114	1569.8
$GH(3,4,5)$	60	33	30	17.0	**29**	0.4	**29**	176368	**29**	41.8
$GH(4,5,6)$	120	68	59	77.2	**57**	4.1	**57**	72972	60	203.8
$GH(5,6,7)$	210	135	**99**	251.9	**99**	58.1	**99**	17100	114	783.6
$J(9,3)$	84	**49**	**49**	36.2	**49**	0.9	**49**	164372	**49**	77.5
$J(10,3)$	120	71	**68**	98.1	**68**	3.2	**68**	124717	**68**	239.9
$J(11,3)$	165	97	**92**	241.1	**92**	10.6	**92**	90684	97	638.8
$J(12,3)$	220	127	**120**	601.1	**120**	36.2	**120**	65944	127	457.1
$J(8,4)$	70	41	**40**	26.1	**40**	0.5	**40**	215414	**40**	107.4
$J(9,4)$	126	**70**	**70**	100.1	**70**	0.9	**70**	462981	**70**	161.3
$J(10,4)$	210	**110**	**110**	297.6	**110**	3.7	**110**	325865	**110**	588.8
$J(11,4)$	330	**170**	**170**	865.2	171	15.3	**170**	228949	**170**	3603.8
$K(9,3)$	84	66	**56**	52.2	59	3.6	58	58243	**56**	62.2
$K(10,3)$	120	96	**86**	90.4	90	7.5	90	49212	88	204.5
$K(11,3)$	165	135	**125**	164.3	131	15.0	131	43851	129	548.4
$K(12,3)$	220	183	**173**	315.6	181	27.9	180	44361	183	228.1
$K(9,4)$	126	61	**51**	40.4	52	5.2	52	31173	52	165.1
$K(10,4)$	210	169	**106**	377.9	118	72.1	117	21061	132	588.3
$K(11,4)$	330	294	**197**	1289.7	223	229.5	219	21995	221	1787.7
N_{best}		5	20		14		15		9	

TABLE 9

Numerical results on the upper bound for the bandwidth of random symmetric sparse matrices

Problem			Bi-$L_{0.75}$		irCM		irCM+		Bi-RoTS	
n	k	$\overline{bw^r}$	$\overline{bw}$	$\overline{time}$	$\overline{bw}$	$\overline{time}$	$\overline{bw}$	$\overline{nrun}$	$\overline{bw}$	$\overline{time}$
80	15	20.8	**15.4**	5.7	16.4	1.1	16.4	21129	15.4	28.9
80	23	31.6	**23.0**	7.7	24.6	1.8	24.6	18648	23.5	26.9
80	31	39.9	**31.4**	10.1	33.5	2.2	33.5	19925	31.4	36.0
80	39	51.0	**39.0**	13.4	41.7	2.3	41.7	24668	39.4	32.4
100	19	26.4	**19.4**	10.5	20.2	1.7	20.1	26017	19.5	63.7
100	29	38.7	**29.4**	13.2	31.1	2.9	31.1	21398	29.6	65.1
100	39	51.3	**39.4**	16.3	41.9	3.8	41.9	18292	39.9	73.5
100	49	61.6	**49.4**	24.6	53.0	4.7	53.0	23037	49.5	71.1
200	39	52.5	**39.5**	73.4	41.4	13.8	41.4	23358	40.3	555.2
200	59	75.9	**59.5**	73.2	62.9	23.7	62.9	12882	60.8	718.7
200	79	99.3	**79.4**	91.7	84.1	35.6	84.1	11786	80.8	657.8
200	99	130.3	**99.4**	134.6	105.6	51.6	105.6	11191	101.3	635.5
300	59	75.1	**59.5**	235.0	63.3	37.3	63.3	28214	60.9	2733.1
300	89	112.4	**89.6**	279.2	94.0	131.3	94.0	8948	90.9	2679.4
300	119	154.9	**119.6**	334.0	125.9	203.5	125.9	7050	120.9	3142.0
300	149	190.4	**149.5**	592.5	157.1	280.9	157.1	8847	151.4	2748.5
N_{best}		0	16		0		0		0	

the results on some random symmetric sparse matrices. For each n and the minimal bandwidth k, we generate 10 random matrices by the MATLAB commands "A = toeplitz([ones(k,1); zeros(n-k,1)]); T = rand(n) > 0.4; T = (T + T'); AS = A.*T; ps = randperm(n); A = AS(ps,ps); " In Table 10, we present the results on 49 sparse matrices from the UF Sparse Matrix Collection [19]. From these three tables, we can observe that Bi-$L_{0.75}$ performs the best in terms of the solution quality while Bi-RoTS performs the worst. In particular, for the Kneser graphs and the random

sparse matrices, Bi-$L_{0.75}$ returns much better bounds than those of irCM and irCM+.

Table 10

Numerical results on the upper bound for the bandwidth of 49 symmetric sparse matrices with $80 \leq n \leq 300$ from the UF Sparse Matrix Collection [19]

Problem			Bi-$L_{0.75}$		irCM		irCM+		Bi-RoTS	
name	n	bw^r	bw	time	bw	time	bw	nrun	bw	time
GD06_theory	101	57	**34**	39.8	35	15.5	35	10266	**34**	65.1
GD97_a	84	23	18	9.8	18	1.1	**17**	36174	**17**	47.6
GD98_c	112	50	**26**	34.1	32	10.9	32	12569	28	111.0
Journals	124	116	**98**	77.1	102	4.2	102	73658	103	151.5
Sandi_authors	86	22	**12**	17.3	**12**	9.6	**12**	7177	13	36.0
Trefethen_150	150	79	**67**	91.4	71	1.3	70	289009	79	203.2
Trefethen_200	200	99	**87**	167.7	88	1.5	88	461919	99	678.5
Trefethen_200b	199	99	**86**	243.1	87	1.5	87	643423	99	667.1
Trefethen_300	300	152	**132**	652.5	**132**	3.2	**132**	817720	150	3973.7
adjnoun	112	67	**38**	32.5	42	20.8	42	6225	42	55.9
ash292	292	32	23	316.3	**21**	26.6	**21**	47741	32	3718.3
ash85	85	13	10	10.0	10	0.6	10	70416	**9**	47.3
bcspwr03	118	17	12	19.6	**11**	11.8	**11**	6459	17	193.5
bcspwr04	274	49	31	304.1	**29**	207.1	**29**	5876	49	2509.4
bcsstk03	112	**3**	**3**	3.8	**3**	0.3	**3**	54309	**3**	162.2
bcsstk04	132	54	**37**	52.4	40	3.5	40	58609	38	184.2
bcsstk05	153	24	**20**	41.8	**20**	4.7	**20**	35781	24	294.0
bcsstk22	138	12	11	63.5	**10**	2.4	**10**	102888	12	158.5
can_144	144	18	**13**	25.9	**13**	8.5	**13**	12159	18	247.4
can_161	161	**18**	**18**	30.1	19	1.3	**18**	90297	**18**	258.9
can_187	187	23	**14**	62.0	15	2.0	15	123730	23	546.6
can_229	229	37	**32**	155.4	**32**	16.1	**32**	39029	37	1041.5
can_256	256	123	67	405.7	**62**	249.2	**62**	6561	85	1179.5
can_268	268	98	57	359.7	**53**	334.7	**53**	4256	82	1836.2
can_292	292	76	**40**	716.8	50	50.2	50	56878	76	628.3
can_96	96	23	**13**	16.0	21	0.6	21	99598	**13**	39.5
dwt_162	162	16	**13**	72.0	14	2.8	14	102152	16	174.3
dwt_193	193	54	**32**	145.3	34	56.0	34	10505	42	891.7
dwt_198	198	13	**8**	70.5	10	3.4	10	83262	13	488.2
dwt_209	209	33	**26**	97.9	29	12.8	29	30822	33	773.0
dwt_221	221	15	**14**	97.8	**14**	2.8	**14**	139580	15	1152.0
dwt_234	234	24	**12**	110.3	**12**	13.5	**12**	32277	24	990.3
dwt_245	245	55	**28**	267.1	31	132.3	31	8100	53	1609.1
dwt_87	87	18	12	9.1	12	4.0	12	9072	**11**	38.6
football	115	76	**37**	43.8	42	10.6	42	16866	39	157.3
grid1	252	22	22	409.5	**19**	0.7	**19**	2200545	22	1799.5
grid1_dual	224	**17**	**17**	162.2	**17**	0.7	**17**	984859	**17**	1195.5
jazz	198	141	**69**	162.2	81	105.1	81	6188	88	324.7
lshp_265	265	18	18	129.2	**17**	0.8	**17**	612015	18	951.9
lund_a	147	**23**	**23**	42.4	**23**	0.4	**23**	396257	**23**	263.5
lund_b	147	**23**	**23**	49.5	**23**	0.4	**23**	462959	**23**	260.3
mesh3e1	289	**17**	**17**	192.7	**17**	0.9	**17**	839178	**17**	2368.1
mesh3em5	289	**17**	**17**	189.4	**17**	0.9	**17**	824889	**17**	2367.0
nos1	237	4	**3**	43.8	**3**	1.1	**3**	156608	4	1706.8
nos4	100	12	**10**	9.9	**10**	4.2	**10**	9422	11	97.1
polbooks	105	30	**20**	17.7	21	12.2	21	5883	22	112.4
spaceStation_1	99	55	**28**	24.1	33	3.9	33	25099	30	90.5
sphere3	258	32	30	328.0	28	1.7	**27**	744388	32	1649.3
tumorAntiAngiogenesis_1	205	199	**100**	170.6	**100**	67.0	**100**	10212	101	542.7
N_{best}		7	37		24		27		12	

8. Concluding remarks. In this paper, we considered optimization problem (1.1) over permutation matrices. We proposed an L_p regularization model for problem (1.1). We studied theoretical properties of the proposed model, including its exactness, the connection between its local minimizers and the permutation matrices, and the lower bound theory of nonzero elements of its KKT points. Based on the L_p regularization model, we proposed an efficient L_p regularization algorithm for solving problem (1.1). To further improve the performance of the proposed L_p regularization algorithm, we combined it with the classical cutting plane technique and/or a novel negative proximal point technique. Our numerical results showed that the proposed L_p regularization algorithm and its variants performed quite well for QAP instances from QAPLIB. For the bandwidth minimization problem which has many applications in different fields, our proposed algorithms also exhibit satisfactory performance.

Notice that the main computational cost of our proposed algorithms arises from many projections onto $\mathcal{D}_n$ (or the restricted domain $\mathcal{A}$). It would be helpful to develop faster algorithms for computing these projection especially when n is large. Moreover, how to take full advantage of the low rank and/or sparse structure of the matrices A and B in our algorithms remains further investigation.

Our L_p regularization model/algorithm for solving problem (1.1) can be directly extended to solve a general nonlinear binary programming with a fixed number of nonzero elements as:

$$\min_{x \in \Omega} f(x) \quad \text{s.t.} \quad x_i \in \{0, 1\}, \ i = 1, 2, \ldots, n, \ \|x\|_0 = N,$$

where $\Omega \subset \mathbb{R}^n$ is convex and N is given. The L_p regularization model for the above problem is

$$\min_{x \in \Omega} f(x) + \sigma \|x + \epsilon \mathbf{e}\|_p^p \quad \text{s.t.} \quad 0 \le x_i \le 1, \ i = 1, 2, \ldots, n, \mathbf{e}^\mathsf{T} x = N,$$

and it can be solved efficiently by the projected gradient method. It would be interesting to extend the L_p-norm regularization to solve some general nonlinear integer programming problems.

REFERENCES

[1] W. P. ADAMS, M. GUIGNARD, P. M. HAHN, AND W. L. HIGHTOWER, *A level-2 reformulation–linearization technique bound for the quadratic assignment problem*, European Journal of Operational Research, 180 (2007), pp. 983–996.

[2] 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.

[3] K. M. ANSTREICHER, *Recent advances in the solution of quadratic assignment problems*, Mathematical Programming, 97 (2003), pp. 27–42.

[4] Z.-J. BAI, D. CHU, AND R. C. TAN, *Computing the nearest doubly stochastic matrix with a prescribed entry*, SIAM Journal on Scientific Computing, 29 (2007), pp. 635–655.

[5] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, IMA Journal of Numerical Analysis, 8 (1988), pp. 141–148.

[6] R. Battiti and G. Tecchiolli, *The reactive tabu search*, ORSA Journal on Computing, 6 (1994), pp. 126–140.

[7] M. S. Bazaraa and H. D. Sherali, *On the use of exact and heuristic cutting plane methods for the quadratic assignment problem*, Journal of the Operational Research Society, (1982), pp. 991–1003.

[8] W. Bian, X. Chen, and Y. Ye, *Complexity analysis of interior point algorithms for non-Lipschitz and nonconvex minimization*, Mathematical Programming, 149 (2015), pp. 301–327.

[9] E. G. Birgin, J. M. Martínez, and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization, 10 (2000), pp. 1196–1211.

[10] G. Birkhoff, *Tres observaciones sobre el algebra lineal*, Univ. Nac. Tucumán Rev. Ser. A, 5 (1946), pp. 147–151.

[11] R. Burkard, *Quadratic assignment problems*, in Handbook of Combinatorial Optimization, P. M. Pardalos, D.-Z. Du, and R. L. Graham, eds., Springer New York, 2013, pp. 2741–2814.

[12] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis, *The quadratic assignment problem*, in Handbook of Combinatorial Optimization, Springer, 1999, pp. 1713–1809.

[13] R. E. Burkard, S. E. Karisch, and F. Rendl, *QAPLIB–a quadratic assignment problem library*, Journal of Global Optimization, 10 (1997), pp. 391–403.

[14] E. J. Candes, J. K. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on pure and applied mathematics, 59 (2006), pp. 1207–1223.

[15] X. Chen, F. Xu, and Y. Ye, *Lower bound theory of nonzero entries in solutions of ℓ_2-ℓ_p minimization*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2832–2852.

[16] P. Z. Chinn, J. Chvátalová, A. K. Dewdney, and N. E. Gibbs, *The bandwidth problem for graphs and matrices-a survey*, Journal of Graph Theory, 6 (1982), pp. 223–254.

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

[18] Y.-H. Dai and R. Fletcher, *Projected Barzilai–Borwein methods for large-scale box-constrained quadratic programming*, Numerische Mathematik, 100 (2005), pp. 21–47.

[19] T. A. Davis and Y. Hu, *The university of Florida sparse matrix collection*, ACM Transactions on Mathematical Software (TOMS), 38 (2011), p. 1.

[20] D. L. Donoho, *Compressed sensing*, Information Theory, IEEE Transactions on, 52 (2006), pp. 1289–1306.

[21] Z. Drezner, *The quadratic assignment problem*, in Location Science, Springer, 2015, pp. 345–363.

[22] T. A. Feo and M. G. Resende, *Greedy randomized adaptive search procedures*, Journal of Global Optimization, 6 (1995), pp. 109–133.

[23] M. Fischetti, M. Monaci, and D. Salvagnin, *Three ideas for the quadratic assignment problem*, Operations Research, 60 (2012), pp. 954–964.

[24] G. Flnke, R. E. Burkard, and F. Rendl, *Quadratic assignment problems*, Surveys in Combinatorial Optimization, (2011), pp. 61–82.

[25] F. Fogel, R. Jenatton, F. Bach, and A. d'Aspremont, *Convex relaxations for permutation problems*, arXiv preprint arXiv:1306.4805, (2013).

[26] 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.

[27] D. Ge, X. Jiang, and Y. Ye, *A note on the complexity of L_p minimization*, Mathematical Programming, Series B, 129 (2011), pp. 285–299.

[28] A. George and J. W. Liu, *Computer solution of large sparse positive definite*, Prentice Hall Professional Technical Reference, 1981.

[29] M. Goemans, *Smallest compact formulation for the permutahedron*, Mathematical Programming, 153 (2015), pp. 5–11.

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

[31] S. Ji, K.-F. Sze, Z. Zhou, A. M.-C. So, and Y. Ye, *Beyond convex relaxation: A polynomial-time non-convex optimization approach to network localization*, in INFOCOM, 2013 Proceedings IEEE, IEEE, 2013, pp. 2499–2507.

[32] S. Kim, M. Kojima, and K.-C. Toh, *A Lagrangian-DNN relaxation: a fast method for computing tight lower bounds for a class of quadratic optimization problems*, Mathematical Programming, 156 (2015), pp. 161–187.

[33] T. C. Koopmans and M. Beckmann, *Assignment problems and the location of economic activities*, Econometrica: Journal of the Econometric Society, (1957), pp. 53–76.

[34] Y.-L. LAI AND K. WILLIAMS, *A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs*, Journal of graph theory, 31 (1999), pp. 75–94.

[35] E. L. LAWLER, *The quadratic assignment problem*, Management Science, 9 (1963), pp. 586–599.

[36] C. H. LIM AND S. J. WRIGHT, *Beyond the Birkhoff polytope: convex relaxations for vector permutation problems*, arXiv preprint arXiv:1407.6609, (2014).

[37] Y.-F. LIU, Y.-H. DAI, AND S. MA, *Joint power and admission control: Non-convex l_q approximation and an effective polynomial time deflation approach*, IEEE Transactions on Signal Processing, 63 (2015), pp. 3641–3656.

[38] Y.-F. LIU, S. MA, Y.-H. DAI, AND S. ZHANG, *A smoothing SQP framework for a class of composite L_q minimization over polyhedron*, Mathematical Programming, 158 (2016), pp. 467–500.

[39] Z.-Y. LIU, H. QIAO, AND L. XU, *An extended path following algorithm for graph-matching problem*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 1451–1456.

[40] E. M. LOIOLA, N. M. M. DE ABREU, P. O. BOAVENTURA-NETTO, P. HAHN, AND T. QUERIDO, *A survey for the quadratic assignment problem*, European Journal of Operational Research, 176 (2007), pp. 657–690.

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

[42] V. LYZINSKI, D. FISHKIND, M. FIORI, J. T. VOGELSTEIN, C. E. PRIEBE, AND G. SAPIRO, *Graph matching: relax at your own risk*, arXiv preprint arXiv:1405.3133, (2014).

[43] A. MISEVICIUS, *An implementation of the iterated tabu search algorithm for the quadratic assignment problem*, OR Spectrum, 34 (2012), pp. 665–690.

[44] N. MLADENOVIC, D. UROSEVIC, D. PÉREZ-BRITO, AND C. G. GARCÍA-GONZÁLEZ, *Variable neighbourhood search for bandwidth reduction*, European Journal of Operational Research, 200 (2010), pp. 14–27.

[45] K. A. MURTHY, Y. LI, AND P. M. PARDALOS, *A local search algorithm for the quadratic assignment problem*, Informatica, 3 (1992).

[46] C. H. PAPADIMITRIOU, *The NP-completeness of the bandwidth minimization problem*, Computing, 16 (1976), pp. 263–270.

[47] L. PARDALOS AND M. RESENDE, *A greedy randomized adaptive search procedure for the quadratic assignment problem*, Quadratic Assignment and Related Problems, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, 16 (1994), pp. 237–261.

[48] R. E. B. E. QELA AND B. KLINZ, *On the biquadratic assignment problem*, in Quadratic Assignment and Related Problems: DIMACS Workshop, May 20-21, 1993, vol. 16, American Mathematical Society, 1994, pp. 117–146.

[49] R. J. SAMWORTH, M. YUAN, ET AL., *Independent component analysis via nonparametric maximum likelihood estimation*, The Annals of Statistics, 40 (2012), pp. 2973–3002.

[50] E. TAILLARD, *Robust taboo search for the quadratic assignment problem*, Parallel Computing, 17 (1991), pp. 443–455.

[51] E. D. TAILLARD, *Comparison of iterative searches for the quadratic assignment problem*, Location Science, 3 (1995), pp. 87–105.

[52] D. M. TATE AND A. E. SMITH, *A genetic approach to the quadratic assignment problem*, Computers & Operations Research, 22 (1995), pp. 73–83.

[53] E. R. VAN DAM AND R. SOTIROV, *On bounding the bandwidth of graphs with symmetry*, INFORMS Journal on Computing, 27 (2014), pp. 75–88.

[54] Z. WEN AND W. YIN, *A feasible method for optimization with orthogonality constraints*, Mathematical Programming, 142 (2013), pp. 397–434.

[55] Y. XIA, *An efficient continuation method for quadratic assignment problems*, Computers & Operations Research, 37 (2010), pp. 1027–1032.

[56] Y. XIA AND Y.-X. YUAN, *A new linearization method for quadratic assignment problems*, Optimization Methods and Software, 21 (2006), pp. 805–818.

[57] M. ZASLAVSKIY, F. BACH, AND J.-P. VERT, *A path following algorithm for the graph matching problem*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31 (2009), pp. 2227–2242.

[58] H.-C. 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.