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 nonmonotone 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:

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

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

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

in which $e \in \mathbb{R}^n$ 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:

$$\min_{X \in \Pi_n} \operatorname{tr}(A^T X B X^T),$$
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 \( \text{NP} \)-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:

\[
D_n = \{ X \in \mathbb{R}^{n \times n} \mid X e = X^T e = e, X \geq 0 \},
\]

where the symbol \( \geq \) denotes the componentwise ordering and \( 0 \in \mathbb{R}^{n \times n} \) is the all-zero matrix. By the Birkhoff-von Neumann theorem [10], we know that \( D_n \) is the convex hull of \( \Pi_n \) and the set of vertices of \( D_n \) is exactly \( \Pi_n \). 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 \( D_n \).

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

\[
\min_{X \in D_n} \text{tr}(A^T B X^T) + \mu_k \|X\|_F^2
\]

with dynamically decreasing parameters \( \{ \mu_k \mid k = 0, 1, 2, \ldots \} \). The initial value \( \mu_0 \) in subproblem (1.3) was chosen such that the problem is convex and the subproblem with fixed \( \mu_k \) 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 \( \mu_k \) is chosen such that the problem is strongly concave, but they only solved (1.3) once. In addition,
Huang [30] used the quartic term $\|X \odot (1 - X)\|_2^2$ to construct the regularization problem, where $\odot$ is the Hadamard product and $1 \in \mathbb{R}^{n \times n}$ is the matrix of all ones.

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

$$\min_{X \in \Pi_n} \|AX - XB\|_F^2,$$

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 $-2\text{tr}(A^TXBX^T) + \|A\|_F^2 + \|B\|_F^2$, 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:

$$\min_{X \in \Pi_n} \pi_I^T X^T L_A X \pi_I,$$

where $A$ is binary symmetric, $\pi_I = (1, 2, \ldots, n)^T$, and $L_A = \text{Diag}(Ae) - A$. Here, Diag$(Ae)$ represents a diagonal matrix whose diagonal elements are $Ae$. They used the regularization term $\|PX\|_F^2$ with $P = I_n - \frac{1}{n}11^T$, 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 $\Pi_n = \{X \mid X^TX = I_n, X \geq 0\}$, Wen and Yin [54] reformulated QAP (1.2) as

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

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 nonmonotone 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 $\langle M, N \rangle = \text{tr}(M^T N)$, where $\text{tr}(\cdot)$ is the trace operator. The Kronecker product of any two matrices $M$ and $N$ is denoted as $M \otimes N$. We define $\mathcal{D}_{[0,1]^n} = \{X \in \mathbb{R}^{n \times n} \mid 0 \leq X \leq 1\}$, 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 $\Lambda(X_i) = \{j \in \mathcal{N} \mid (i, j) \in \Lambda(X)\}$ and $\Lambda(X,j) = \{i \in \mathcal{N} \mid (i, j) \in \Lambda(X)\}$. 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 $\mathcal{N}_2(X) = \{Z \in \Pi_n : \|Z - X\|_F^2 \leq 4\} = \{Z \in \Pi_n \mid \langle Z, X \rangle \geq n/2\}$ 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,

\begin{equation}
\|\nabla f(X) - \nabla f(Y)\|_F \leq L\|X - Y\|_F, \quad \forall X, Y \in \mathcal{D}_{[0,1]^n}.
\end{equation}

Moreover, we assume

\begin{equation}
\frac{L^2}{2}\|Y - X\|_F^2 \leq f(Y) - f(X) - \langle \nabla f(X), Y - X \rangle \leq \frac{L^2}{2}\|Y - X\|_F^2, \quad \forall X, Y \in \mathcal{D}_n.
\end{equation}
Note that when $\varphi_f < 0$, (2.2) implies that $f(\cdot)$ is strongly concave. By some easy calculations, (2.2) indicates that

$$(3.3) \quad f(tx + (1-t)y) \geq tf(x) + (1-t)f(y) - \frac{\varphi_f}{2} t(1-t)\|X - Y\|_F^2$$

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 $L = \max\{||\varphi_f||, ||\varphi_f||\}$, where $\varphi_f$ and $\varphi_f$ are taken as the largest and smallest eigenvalues of $B^T \otimes A^T + B \otimes A \in \mathbb{R}^{n^2 \times n^2}$, 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

$$(2.4) \quad \varphi_h := p(1-p)(1+\epsilon)^{p-2}$$

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

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

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 $\varphi_h$ on $\mathcal{D}_{[0,1]^n}$, namely,

$$(2.6) \quad h(tX + (1-t)Y) \geq t h(X) + (1-t)h(Y) + \frac{\varphi_h}{2} t(1-t)\|Y - X\|_F^2$$

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 $\Pi_n$ can be equivalently characterized as

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

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:

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

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

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

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

**Lemma 3.1.** Suppose $\sigma > \frac{1}{2}(f^* - f)$, where $f^*$ and $f$ 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
\begin{equation}
\tag{3.4}
f^* + \sigma \|X^*\|_0 \geq f(X(\sigma)) + \sigma \|X(\sigma)\|_0.
\end{equation}
Moreover, by the definition of $f$, we have $f(X(\sigma)) + \sigma \|X(\sigma)\|_0 \geq f + \sigma \|X(\sigma)\|_0$. This inequality, together with (3.4) and the fact $\|X(\sigma)\|_0 \geq n + 2$, implies
\[ \sigma \leq (f^* - f)/ (\|X(\sigma)\|_0 - \|X^*\|_0) \leq (f^* - f)/2, \]
which is a contradiction to $\sigma > \frac{1}{2}(f^* - f)$.

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). \qed

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 $X \in D_n$, $\|X\|_1$ 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 $h(X) = \|X + \epsilon 1\|_p^p$ to approximate $\|X\|_0$ in (3.2) and obtain the corresponding $L_p$-norm regularization model:
\begin{equation}
\tag{3.5}
\min_{X \in D_n} F_{\sigma,p,\epsilon}(X) := f(X) + \sigma \|X + \epsilon 1\|_p^p.
\end{equation}

Note that the problem of minimizing $h(X)$ over $D_n$ and the problem of minimizing $\|X\|_0$ over $D_n$ 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 $\sigma$ in problem (3.5) mainly controls the sparsity of $X$ while the parameter $\epsilon$ affects the smoothness of the regularizer. Roughly speaking, problem (3.5) with a large $\sigma$ returns a sparse $X$ close to a permutation matrix, while problem (3.5) with a relatively large (not too small) $\epsilon$ 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
\begin{equation}
\tag{3.6}
\sigma > \sigma_{p,\epsilon}^* := \max \left\{ \frac{\partial f}{\partial h}, 0 \right\},
\end{equation}
where $\partial f$ and $\partial h$ 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
\begin{equation}
\tag{3.7}
\sigma h(tX + (1 - t)Y) \geq \sigma \left( t h(X) + (1 - t) h(Y) + \frac{\partial h}{\partial h} t (1 - t) \|Y - X\|_p^p \right)
\end{equation}
for any $X, Y \in D_n$ 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
We next claim that one can pick \( n \) elements from different columns and rows of \( X \) in \((3.13)\) such that their summation is equal to \((3.12)\) \( \bar{W} \) and \((3.11)\).

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

\[
f(X^*) + \sigma \|X^* + e\|_p^p \geq f(X_{\sigma,p,\epsilon}) + \sigma \|X_{\sigma,p,\epsilon} + e\|_p^p,
\]

which, together with \( \|X^*\|_p = \|X_{\sigma,p,\epsilon}\|_p^p = n(1 + \epsilon)^p \), 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 \( X \in D_n \) is called 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 \((i, j) \in N \times N\), there holds

\[
X_{ij} S_{ij} = 0, \quad S_{ij} \geq 0,
\]

\[
(W_{ij} - \lambda_i - \mu_j)(X_{ij} + \epsilon) + \sigma p(X_{ij} + \epsilon)^p = \epsilon S_{ij},
\]

where \( W_{ij} = (\nabla f(X))_{ij} \) and \( S, \lambda, \) and \( \mu \) are the Lagrange multipliers corresponding to the constraints \( X \geq 0 \), \( X e = e \), and \( X^T e = e \), 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

\[
\bar{X}_{ij} \geq \max(\bar{c} - \epsilon, 0),
\]

where \( \bar{c} = (\|\bar{X}\|_1 + \epsilon') (n + \|\bar{X}\|_0 + \epsilon) - (n - 1)(1 + \epsilon)^{p-1} + \sqrt{2n} \epsilon' \|\nabla f(0)\|_0 \). \( \sqrt{p} \).

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

\[
\sum_{i \in \Lambda(\bar{X})} \bar{X}_{ij} = 1, \quad \forall \ j \in N, \quad \sum_{j \in \Lambda(\bar{X}_i)} \bar{X}_{ij} = 1, \quad \forall \ i \in N
\]

and

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

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

\[
\sum_{(i, j) \in \Lambda(\bar{X})} (\bar{W}_{ij} \bar{X}_{ij} + \sigma p(\bar{X}_{ij} + \epsilon)^{p-1} \bar{X}_{ij}) = \sum_{(i, j) \in \Lambda(\bar{X})} (\lambda_i + \mu_j) \bar{X}_{ij} = (\lambda + \mu)^T e,
\]

where the second equality is due to (3.11) and

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

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

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

Combining (3.13) and (3.14) and then rearranging the obtained equation, we have

\[
(3.15) \quad \sum_{(i,j) \in \Lambda(\bar{X})} (\bar{X}_{ij} + \epsilon)^{p-1} \bar{X}_{ij} = \frac{1}{\sigma p} \sum_{(i,j) \in \Lambda(\bar{X})} \bar{W}_{ij} (\bar{X}_{ij} - \bar{X}_{ij}) + \sum_{(i,j) \in \Lambda(\bar{X})} (\bar{X}_{ij} + \epsilon)^{p-1} \bar{X}_{ij}
\]

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

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

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

It is easy to see that \( \|\bar{X} - \bar{X}\|_F \leq \sqrt{2n} \). Thus, we obtain

\[
(3.17) \quad \sum_{(i,j) \in \Lambda(\bar{X})} \bar{W}_{ij} (\bar{X}_{ij} - \bar{X}_{ij}) \leq \|\nabla f(\bar{X})\|_F \|\bar{X} - \bar{X}\|_F \leq \sqrt{2n}(L \sqrt{n} + \|\nabla f(0)\|_F).
\]

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

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

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 extention 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

\[
(3.19) \quad \sigma > \sigma_{p,\epsilon} := \frac{c}{p} \frac{L(2 + \sqrt{n}) + \|\nabla f(0)\|_F}{\epsilon^{p-1} - (1/2 + \epsilon)^{p-1}},
\]
where \( c > 1 \) is a constant. On the other hand, any local minimizer of (3.5) with \( \sigma > \sigma^*_{p,c} \) is a permutation matrix, where \( \sigma^*_{p,c} \) 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

\[
\Lambda_1^l = \{(i, j) | \bar{X}_{ij} = 0, X_{ij}^l = 0\}, \quad \Lambda_2^l = \{(i, j) | \bar{X}_{ij} = 0, X_{ij}^l = 1\}, \\
\Lambda_3^l = \{(i, j) | \bar{X}_{ij} = 1, X_{ij}^l = 0\}, \quad \Lambda_4^l = \{(i, j) | \bar{X}_{ij} = 1, X_{ij}^l = 1\}.
\]

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

\[
f(\bar{X} + tD^l) - f(\bar{X}) = t\langle \nabla f(\bar{X} + \xi tD^l), D^l \rangle \geq -t\|D^l\|_F \|\nabla f(\bar{X} + \xi tD^l)\|_F \\
\geq -t\|D^l\|_F \cdot (L\|\bar{X} + \xi tD^l\|_F + \|\nabla f(0)\|_F) \\
\geq -t \left(L\|D^l\|_F^2 + \|D^l\|_F \sqrt{n}\right) + \|D^l\|_F \|\nabla f(0)\|_F),
\]

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

\[
h(\bar{X} + tD^l) - h(\bar{X}) = \sum_{(i, j) \in \bigcup_{l=1}^{n!} \Lambda_i^l} ((\bar{X}_{ij} + tD_{ij}^l + \epsilon)^p - (\bar{X}_{ij} + \epsilon)^p) \\
= |\Lambda_2^l| ((t + \epsilon)^p - \epsilon^p + |\Lambda_3^l| ((1 - t + \epsilon)^p - (1 + \epsilon)^p) \\
> \frac{p}{2} \|D^l\|_F^2 \left((t + \epsilon)^{p-1} - (1 - t + \epsilon)^{p-1}\right) t \\
> \frac{p}{2} \|D^l\|_F^2 \left((t + \epsilon)^{p-1} - (1/2 + \epsilon)^{p-1}\right) t,
\]

where the second equality is due to (3.20); the first inequality uses \(|\Lambda_2^l| = |\Lambda_3^l| = \frac{\|D^l\|_F^2}{2}\); 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 \( \hat{t} := \left(c^{-1} \epsilon^{p-1} + (1 - c^{-1})(1/2 + \epsilon)^{p-1}\right)^{1/\epsilon} - 1 \). Using the fact that the function \((\xi_1 + \xi_2(1/2 + z)^{p-1})^{1/\xi_2} - 1\) with \( \xi_1, \xi_2 > 0 \) is increasing with \( z \) in \((0, \infty)\), we have \( \hat{t} > (c^{-1} \epsilon^{p-1} + (1 - c^{-1})(1/2 + \epsilon)^{p-1})^{1/\epsilon} - 1 = 0 \). Furthermore, if we restrict \( t \in (0, \hat{t}) \) with \( \hat{t} = \min(1/2, \hat{t}) \), we obtain from (3.22) that

\[
h(\bar{X} + tD^l) - h(\bar{X}) > \frac{p}{2} \|D^l\|_F^{p-1} \left(\frac{\epsilon^{p-1}}{c} - \frac{(1/2 + \epsilon)^{p-1}}{c}\right)
\]

which together with (3.19) and \( \|D^l\|_F \geq 2 \) implies that

\[
(3.23) \quad \sigma(h(\bar{X} + tD^l) - h(\bar{X})) > t \left(L\|D^l\|_F^2 + \|D^l\|_F \sqrt{n}\right) + \|D^l\|_F \|\nabla f(0)\|_F).
\]

Combining (3.21) and (3.23), we have

\[
(3.24) \quad F_{\sigma, p, c}(\bar{X} + tD^l) - F_{\sigma, p, c}(\bar{X}) > 0
\]

for any \( t \in (0, \hat{t}) \). This fact means that \( D^l, l = 1, \ldots, n! - 1 \), are strictly increasing feasible directions. Let \( \text{Conv}(\bar{X}, \hat{t}) \) denote the convex hull spanned by points \( \bar{X} \) and
\( \hat{X} + tD', l = 1, \ldots, n! - 1 \). Thus for any \( X \in \text{Conv}(\hat{X}, \hat{t}) \), we have \( F_{\sigma_p, \epsilon}(X) > F_{\sigma_p, \epsilon}(\hat{X}) \) by (3.24) and the strict concavity of \( h(X) \). Moreover, one can always choose a sufficiently small but fixed \( t > 0 \) such that \( B(\hat{X}, t) \cap D_n \subset \text{Conv}(\hat{X}, \hat{t}) \), where \( B(\hat{X}, t) = \{ X \in \mathbb{R}^{n \times n} | \| X - \hat{X} \|_F \leq t \} \). Consequently, \( \hat{X} \) 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 \( D_n \) 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

\[
\min (F_{\sigma_p, \epsilon}(\bar{X} - t'D), F_{\sigma_p, \epsilon}(\bar{X} + t'D)) < F_{\sigma_p, \epsilon}(\bar{X}),
\]

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 \( \sigma \) 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 \( \sigma \) from an initial value \( \sigma_0 \) since a large \( \sigma \) may result in the ill-conditioned subproblem. Since \( \sigma_{p, \epsilon}^* \) is increasing with respect to \( \epsilon \) if \( \sigma_f > 0 \) (see Theorem 3.2), dynamically decreasing \( \epsilon \) from a relatively large value \( \epsilon_0 \) is helpful in finding a permutation matrix relatively faster.

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

\[
X_{k+1}^{(0)} := X_k
\]

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

\[
X_{k+1}^{(0)} = \text{a perturbation of } X_k \text{ in } D_n.
\]

The following theorem shows that once we find a permutation matrix, which is a KKT point of problem (3.5) with some \( \sigma_k \) and \( \epsilon_k \), 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 \( \sigma \) and \( \epsilon \) satisfying \( \sigma \geq \hat{\sigma} \) and \( 0 < \epsilon \leq \hat{\epsilon} \).
Proof. Denote the corresponding permutation vector of $X$ as $\pi$. Let the corresponding multipliers be $\lambda^{\sigma, \hat{\epsilon}}, \mu^{\sigma, \hat{\epsilon}},$ and $S^{\sigma, \hat{\epsilon}}$. Then we know from (3.9) that for $i, j \in \mathcal{N}$, there holds
\[
S^{\sigma, \hat{\epsilon}}_{ij} = \begin{cases} W_{ij} - \lambda^{\sigma, \hat{\epsilon}}_i - \mu^{\sigma, \hat{\epsilon}}_j + \hat{\sigma}p(1 + \hat{\epsilon})^{p-1}, & i = \pi(j), \\ W_{ij} - \lambda^{\sigma, \hat{\epsilon}}_i - \mu^{\sigma, \hat{\epsilon}}_j + \hat{\sigma}p^{p-1}, & i \neq \pi(j), 
\end{cases}
\]
and $S^{\sigma, \hat{\epsilon}}_{ij} = 0$ if $i = \pi(j)$ and $S^{\sigma, \hat{\epsilon}}_{ij} \geq 0$ if $i \neq \pi(j)$. For any $\sigma \geq \hat{\sigma}$ and $0 < \epsilon \leq \hat{\epsilon}$, define
\[
\lambda^{\sigma, \epsilon} = \lambda^{\sigma, \hat{\epsilon}} + p \left( (1 + \epsilon)^{p-1} - (1 + \hat{\epsilon})^{p-1} \right) e, \quad \mu^{\sigma, \epsilon} = \mu^{\sigma, \hat{\epsilon}}
\]
and
\[
S^{\sigma, \epsilon}_{ij} = \begin{cases} W_{ij} - \lambda^{\sigma, \epsilon}_i - \mu^{\sigma, \epsilon}_j + \sigma p(1 + \epsilon)^{p-1}, & i = \pi(j), \\ W_{ij} - \lambda^{\sigma, \epsilon}_i - \mu^{\sigma, \epsilon}_j + \sigma p^{p-1}, & i \neq \pi(j). 
\end{cases}
\]
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
\[
S^{\sigma, \epsilon}_{ij} = W_{ij} - \lambda^{\sigma, \epsilon}_i - \mu^{\sigma, \epsilon}_j + \sigma p(1 + \epsilon)^{p-1} = S^{\sigma, \hat{\epsilon}}_{ij} = 0;
\]
for any $i \neq \pi(j)$, we have
\[
S^{\sigma, \epsilon}_{ij} = W_{ij} - \lambda^{\sigma, \epsilon}_i - \mu^{\sigma, \epsilon}_j + \sigma p^{p-1} \\
= S^{\sigma, \hat{\epsilon}}_{ij} + \sigma \left( (1 + \hat{\epsilon})^{p-1} - (1 + \epsilon)^{p-1} \right) - p\sigma \left( (1 + \epsilon)^{p-1} - \epsilon^{p-1} \right) \\
\geq S^{\sigma, \hat{\epsilon}}_{ij} + (1 + \hat{\epsilon})^{p-1} - \epsilon^{p-1} \geq S^{\sigma, \hat{\epsilon}}_{ij} \geq 0,
\]
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 $0 < \epsilon \leq \hat{\epsilon}$.

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 $k = 0$, $\epsilon_0 > 0$, $\sigma_0 > 0$, $\hat{\sigma} > 0$.
2. **while** $\|X_k\|_p > n$ **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 $\sigma_k$ and $\epsilon_k$ starting from $X_k^{(0)}$ such that $F_{\sigma_k, \epsilon_k}(X_k) \leq F_{\sigma_k, \epsilon_k}(X_k^{(0)})$.
5. **Choose** $\sigma_{k+1} > \sigma_k + \hat{\epsilon}$, $0 < \epsilon_{k+1} \leq \epsilon_k$ 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 $\sigma_k$ and $\epsilon_k$, we have $\sigma_k > \max\{\sigma^*_p, 0, \sigma_{p, c_0}\}$ for some finite $k$, where $\sigma^*_p$ and $\sigma_{p, c_0}$ are defined in (3.6) and (3.19), respectively. The update scheme of $\epsilon_k$ yields that $\sigma^*_p > \sigma^*_p + \hat{\epsilon}$ and $\sigma_{p, c_0} \geq \sigma_{p, c_0}$. It follows from the proof of Theorem 3.2 that $F_{\sigma_p, \epsilon_k}(X)$ with $\sigma_p > \max\{\sigma^*_p, 0, \sigma_{p, c_0}\}$ and $\epsilon_k = 0$ is strongly concave. Consequently, any KKT point $X_k$ of problem (3.5) with $\sigma > \max\{\sigma^*_p, 0, \sigma_{p, c_0}\}$
and $0 < \epsilon \leq \epsilon_0$ 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 $(\sigma, \epsilon) = (\kappa_{k+1}, \epsilon_{k+1})$. By the choice of $X_{k+1}^{(0)}$, we have $F_{\sigma_{k+1}, \epsilon_{k+1}}(X_{k+1}^{(0)}) < F_{\sigma_{k+1}, \epsilon_{k+1}}(X_k)$. This inequality together with the fact that $F_{\sigma_{k+1}, \epsilon_{k+1}}(X_{k+1}) \leq F_{\sigma_{k+1}, \epsilon_{k+1}}(X_{k+1}^{(0)})$ 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 $X_{k+1} = X_k$ 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 for solving problem (3.5). Specifically, starting from initial $X_k^{(0)}$, the projected gradient method with the BB step sizes [5, 9] to compute an approximate KKT point for any point $X$ for any point $X$ gradient method with the BB step sizes [5, 9] to compute an approximate KKT point for any point $X$.

\[
X_{k+1}^{(0)} = X_k^{(i)} + \delta \nabla F_{\sigma, \epsilon}(X_k^{(i)}), \quad \delta \in (0, 1),
\]

where the step size $\delta$ 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:

\[
F_{\sigma, \epsilon}(X_k^{(i)} + \delta \nabla F_{\sigma, \epsilon}(X_k^{(i)})) \leq F_{\sigma, \epsilon}(X_k^{(i)}), \quad \theta \in (0, 1),
\]

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

Any intermediate point $X \in D_n$ 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 $\epsilon$. Specifically, for any point $X_k^{(i)}$, we generate $\hat{X}_k^{(i)}$ from $X_k^{(i)}$ such that $X_k^{(i)} \in X_2^{\text{best}}(\hat{X}_k^{(i)})$ and update

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

Let $f_k^{\text{best}}$ be the best function value among $f_1^{\text{best}}, \ldots, f_k^{\text{best}}$. If $f_k^{\text{best}}$ is less than or equal to $f_k^{\text{best}}$, we set $\epsilon_k = \epsilon_{k-1}$; otherwise we set $\epsilon_k = \gamma \epsilon_{k-1}$ with $\gamma \in (0, 1)$. Moreover, recall that $\sigma_{p, \epsilon}$ in Theorem 3.5 is increasing with respect to $\epsilon$ and $\lim_{\epsilon \to 0} \sigma_{p, \epsilon} = 0$. Therefore, if $\epsilon_k$ 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 $\epsilon_{\text{min}}$ for $\epsilon_k$. We summarize the strategy of updating $\epsilon_k$ as follows

\[
\text{if } f_k^{\text{best}} < f_k^{\text{best}} \quad \text{then } \quad f_k^{\text{best}} = f_k^{\text{best}}, X_k^{\text{best}} = X_k^{\text{best}}, \epsilon_k = \epsilon_{k-1},
\]

\[
\text{else } \quad \epsilon_k = \max(\gamma \epsilon_{k-1}, \epsilon_{\text{min}}).
\]

end
Finally, we use the continuation scheme to dynamically update $\sigma$ such that problem \eqref{eq:lagrangian_dual} changes from being strongly convex to being strongly concave. To achieve this goal, we choose $\sigma_0 = \min \left\{ \frac{2}{\rho_1 \rho_2 p_0^p - p}, \sigma_- \right\}$, where $L_f$ is defined in \eqref{eq:lbfgs} and $\sigma_- < 0$ is given. Motivated by the strategy used in \cite{Bo2021}, we update $\sigma_{k+1}$ by

\begin{equation}
\sigma_{k+1} := \min \{ \tilde{\sigma}_{k+1}, \sigma_{\max} \}
\end{equation}

where $\tilde{\sigma}_{k+1}$ is given. To keep $\sigma$ safe for $\sigma_0$, we take $\sigma_{\max} = 2^{-l} \sigma_0$, where $l = \lceil \log_2(-\sigma_0) \rceil$ and $\sigma_{\max} > 0$ is a safeguard for $\sigma$.

We summarize the practical $L_p$ regularization algorithm for solving problem (1.1) in Algorithm 2. Specifically, the stopping rules on line 5 are $\text{tol}_i^f = \|X_i^{(i)} - X_i^{(i-1)}\|_F / \sqrt{n}$, $\text{tol}_i^d = \left| \frac{F_{\alpha_k}^{(i)}(X_k^{(i)}) - F_{\alpha_k}^{(i)}(X_k^{(i-1)})}{1 + |F_{\alpha_k}^{(i)}(X_k^{(i-1)})|} \right|$ and $\tau_k^\alpha = \max \{ \tau_0^\alpha / k^3, \tau_0^\beta \}$. As described after the proof of Theorem 3.4, the lower bound \eqref{eq:lbfgs} may be helpful in Line 9.

\begin{algorithm}
\begin{algorithmic}[1]
\State Given $X_0 \in \mathcal{D}_n$, set $\epsilon_0, \tau_0^\alpha, \tau_0^\beta, \tau_0^\gamma > 0$, $\sigma_0 \leq 0$, $\theta, \delta, \gamma, \text{tol} \in (0, 1)$.
\State Set $k = 0$, $f_{\text{best}} = \infty$.
\While {$\|X_k\|_F / n - 1 > \text{tol}$}
\hspace{1em} // Lines 4 - 9: compute an approximate KKT point of (3.5)
\hspace{1em} Choose $X_k^{(0)}$ by \eqref{eq:proj_update} and \eqref{eq:proj_update2}. Set $i = 0$, $\alpha_i > 0$, $\text{tol}_i^f = \text{tol}_i^d = \infty$.
\hspace{1em} \While {$\text{tol}_i^f > \tau_k^\alpha$ or $\text{tol}_i^d > \tau_k^\beta$}
\hspace{2em} Compute $D^{(i)} = \mathcal{P}_{\mathcal{D}_n}(X_k^{(i)}) - \alpha_i \nabla F_{\alpha_k}^{(i)}(X_k^{(i)}) - X_k^{(i)}$.
\hspace{2em} Find the smallest $j$ such that $\delta_j$ satisfies \eqref{eq:delta_j}.
\hspace{2em} Set $X_k^{(i+1)} = X_k^{(i)} + \delta_j D_k^{(i)}$ and $i = i + 1$.
\hspace{1em} \EndWhile
\hspace{1em} Compute $X_k^{(i+1)} = X_k^{(i)} + \delta_i D_k^{(i)}$ and update $f_k^{\text{best}}$.
\hspace{1em} Update $\epsilon_{k+1}$, $\sigma_{k+1}$, $f_{\text{best}}$, and $X_{\text{best}}$ by \eqref{eq:delta_i} and \eqref{eq:sigma_update}.
\hspace{1em} Set $X_{k+1} = X_k^{(i)}$ and $k = k + 1$.
\EndWhile
\end{algorithmic}
\end{algorithm}

\subsection{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

\begin{equation}
\min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2} \|X - C\|^2_F \quad \text{s.t.} \quad X e = e, \ X^T e = e, \ X \geq 0,
\end{equation}

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

The Lagrangian dual problem of \eqref{eq:proj_dual} is

\begin{equation}
\max_{y, z} \min_{X \geq 0} \mathcal{L}(X, y, z),
\end{equation}

where $\mathcal{L}(X, y, z) = \frac{1}{2} \|X - C\|^2_F - \langle y, X e - e \rangle - \langle z, X^T e - e \rangle$, in which $y, z \in \mathbb{R}^n$ are the Lagrange multipliers associated with the linear constraints $X e = e$ and $X^T e = e$, respectively.
respectively. Let $P_+ (\cdot)$ denote the projection onto the nonnegative orthant. Then the dual problem (4.12) can be equivalently rewritten as

$$\min_{y,z} \theta(y, z) := \frac{1}{2} \left\| P_+ (C + ye^T + e z^T) e - e \right\|_F^2 - \langle y + z, e \rangle.$$  

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

$$\nabla \theta(y, z) = \begin{bmatrix} P_+ (C + ye^T + e z^T) e - e \\ P_+ (C + ye^T + e z^T)^T e - e \end{bmatrix}.$$  

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 $P_{D_n}(C) = P_+ (C + y^* e^T + e (z^*)^T).$ 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 $\Pi_n$ of problem (1.1) based on the information of $\tilde{X}$.

Denote $F_\omega(X) := f(X) + \omega \|X\|_F^2,$ where the parameter $\omega$ is chosen such that $F_\omega$ is strongly convex. The ideal cut

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

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

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

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

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

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

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

$$\min_{X \in A_{\Pi_n}} f(X).$$
By replacing \( D_n \) by \( A := D_n \cap \text{LC1}(\tilde{X}) \cap \text{LC2}(\tilde{X}) \), Algorithm 2 can be employed to solve problem (5.4), where the \( L_p \) regularization subproblem becomes

\[
\min_{X \in A} F_{\sigma,p,e}(X).
\]

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

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

1. Given \( A = D_n \), set \( K_{\text{max}} > 0 \) and \( K = 0 \).
2. while \( K < K_{\text{max}} \) do
3. Solve (5.4) by Algorithm 2 with subproblem (3.5) there replaced by (5.5), and obtain \( X^K_{\text{best}} \).
4. Set \( A = A \cap (\text{LC1}(X^K_{\text{best}}) \cap \text{LC2}(X^K_{\text{best}})) \).
5. Set \( X^* = \arg \min_{K} \{ f(X^K_{\text{best}}) \} \) 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 \( A \) in Algorithm 3 is slightly time consuming than computing the projections onto the set \( D_n \) and computational cost of the projections is the dominant part in both algorithms. Note that Algorithm 3 with \( K_{\text{max}} = 1 \) 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 \( K_{\text{max}} = 4 \) 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 \( \hat{X} \in D_n \), consider problem

\[
\min_{X \in \Pi} f(X) - \mu \| X - \hat{X} \|^2_F.
\]

The next theorem shows that the solution of (5.6) with an appropriate choice of \( \mu \) 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

\[
|f(X) - f(Y)| \geq c_2, \quad \forall X, Y \in \Pi \text{ with } f(X) \neq f(Y).
\]

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

\[
\| X^*_\mu - \hat{X} \|_F \geq \| X^* - \hat{X} \|_F.
\]

**Proof.** From the optimality of \( X^*_\mu \), we obtain

\[
f(X^*_\mu) - \mu \| X^*_\mu - \hat{X} \|_F^2 \leq f(X^*) - \mu \| X^* - \hat{X} \|_F^2,
\]
which is equivalent to
\[ f(X^*_\mu) \leq f(X^*) + \mu \left( \|X^*_\mu\|_F^2 - \|X^*\|_F^2 + 2\langle X^* - X^*_\mu, \hat{X} \rangle \right). \]

Combining the above inequality with the facts that \( \|X^*_\mu\|_F = \|X^*\|_F = \sqrt{n} \), \( \langle X^*, \hat{X} \rangle \leq n \), and \( \langle X^*_\mu, \hat{X} \rangle \geq 0 \) yields
\[ (5.10) \quad f(X^*_\mu) \leq f(X^*) + 2\mu n < f(X^*) + c_2, \]

where the last inequality is due to the choice of \( \mu \). 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 \( X^*_\mu \) 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 \( i = 1, \ldots, K \), and combining it with Theorem 3.2, we obtain the following theorem.

**Theorem 5.2.** Suppose \( 0 < \mu < \frac{c_2}{2n} \) and \( \sigma > \max((\nu_f - 2\mu)/\nu_h, 0) \). Then any global solution of
\[ (5.11) \quad \min_{X \in \mathcal{D}_n} f(X) + \sigma \|X + \epsilon 1\|_p^p - \mu \left\| X - \frac{1}{K} \sum_{i=1}^{K} \hat{X}_i \right\|_F^2 \]

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 \( X^\text{best}_K = 0 \).
2. While \( K < K_{\max} \) and \( X^\text{best}_K \notin \{X_1^\text{best}, \ldots, X_{K-1}^\text{best}\} \) do
   3. Solve (5.6) with \( \hat{X} = \frac{1}{K} \sum_{i=1}^{K} X_i^\text{best} \) by Algorithm 2 with subproblem (3.5) replaced by (5.11), and obtain \( X^\text{best}_K \).
   4. Set \( \mu = \mu/2, X^\text{best} = \arg \min_K \{f(X^\text{best}_K)\} \), 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 \( A := \mathcal{D}_n \), set \( \mu > 0, K_{\max} > 0 \) and \( K = 0 \). Define \( X^\text{best}_K = 0 \).
2. While \( K < K_{\max} \) and \( X^\text{best}_K \notin \{X_1^\text{best}, \ldots, X_{K-1}^\text{best}\} \) do
   3. Solve (5.6) with \( \hat{X} = \frac{1}{K} \sum_{i=1}^{K} X_i^\text{best} \) by Algorithm 2 with subproblem (3.5) replaced by (5.11), wherein \( \mathcal{D}_n \) being replaced by \( A \), and obtain \( X^\text{best}_K \).
   4. Set \( A = A \cap \text{LC1}(X^\text{best}_K) \cap \text{LC2}(X^\text{best}_K) \).
   5. Set \( \mu = \mu/2, X^\text{best} = \arg \min_K \{f(X^\text{best}_K)\} \), and \( K = K + 1 \).

We give some remarks on Algorithms 4 and 5 to conclude this section.
Firstly, the term $-\|X - \frac{1}{K}\sum_{i=1}^{K} \hat{X}_i\|_F^2$ 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 $-\|X - \sum_{i=1}^{K} w_i \hat{X}_i\|_F^2$.

Secondly, Theorem 5.2 shows that problem (5.11) with any $\mu \in (0, \frac{2\pi}{n})$ shares the same global solution with problem (1.1). However, (i) since $c_2$ is generally unknown, it is hard to choose the parameter $\mu$ satisfying $\mu < \frac{2\pi}{n}$ and (ii) $\mu \in (0, \frac{2\pi}{n})$ might be too small to effectively improve the performance of Algorithms 4 and 5. Therefore, we initialize $\mu$ 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

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

to measure the quality of the solution returned by different algorithms, where $\text{obj}^*$ 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 $\text{obj}^*$ and obj are computed based on the original $A$ and $B$, and all $\text{obj}^*$ are positive integers.

6.1. Implementation details. For Algorithm 2, we choose the initial point $X_0 = \frac{1}{n}1$, the initial guess $c_0 = 0.1$, the safeguards $\epsilon_\min = 10^{-3}$, $\sigma_{\max} = 10^6$, and the shrinkage parameter for updating $\epsilon_k$ as $\gamma = 0.9$. Our experience shows that Algorithm 2 also works on an $c_0$ larger than 1, smaller shrinkage and safeguard parameters, but the choices of these parameter might affect its efficiency. We set the stopping tolerance $\text{tol} = 10^{-3}$, 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 $\tau_{\min} = 10^{-3}$, $\tau_{\max} = 10^{-6}$ and the corresponding safeguards as $\tau_{\min} = 10^{-5}$, $\tau_{\max} = 10^{-8}$. For problems with $n > 50$, we use MOSEK 7.1 to compute the projection onto $D_n$ if $\max_{ij} |X_{ij}| > 8$, and use dualBB otherwise. For problems with $n \leq 50$, we always use MOSEK 7.1 to compute the projection onto $D_n$.

For Algorithms 3 – 5, we choose $K_{\max} = 10$. As for computing projections onto the restricted domain $A$ in Algorithms 3 and 5, MOSEK 7.1 is always employed no matter how large $n$ is. For Algorithms 4 and 5, we choose $\mu = \min \{0.5, (\overline{\nu} - \overline{\nu}_f)/100\}$.

\footnote{Downloadable from https://www.mosek.com/resources/downloads.}
where $\pi_f$ and $\nu_f$ are taken as the largest and smallest eigenvalues of $B^T \otimes A^T + B \otimes A \in \mathbb{R}^{n^2 \times n^2}$, respectively. The parameter $\omega$ in (5.1), which is used in Algorithms 3 and 5, is chosen to be $1 - 0.5\nu_f$.

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 $\hat{X}^{(k)} \in \mathcal{X}_2^{\text{best}}(\hat{X}^{(k)})$ 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 $\pi \in \mathcal{N}^n$ such that $x_{\pi_1} \leq x_{\pi_2} \leq \cdots \leq x_{\pi_n}$. (c) Let $\hat{X}^{(0)} := 0$. 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 $\{k_0\} = \emptyset$ and set $\hat{X}^{(0)}_{k_i,k_{\pi_i}} = 1$. 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 $\hat{X}^{(k)} \in \arg \min_{Y \in \mathcal{X}_2(\hat{X}^{(k)})} f(Y)$ and update

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

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 $\hat{X}^{(k)}$ is cheap for QAP. Suppose the procedure terminates within $k_0$ iterations. Then the total computational cost is $4n^3 + O(k_0 n^2)$; 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

<table>
<thead>
<tr>
<th>gap ≤ %</th>
<th>0.0</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
<th>2.0</th>
<th>3.0</th>
<th>4.0</th>
<th>5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>LagSA</td>
<td>14</td>
<td>21</td>
<td>29</td>
<td>41</td>
<td>42</td>
<td>44</td>
<td>48</td>
<td>51</td>
<td>53</td>
<td>57</td>
<td>61</td>
<td>80</td>
<td>91</td>
<td>95</td>
<td>99</td>
</tr>
<tr>
<td>LagSA-BB</td>
<td>26</td>
<td>39</td>
<td>50</td>
<td>56</td>
<td>58</td>
<td>62</td>
<td>65</td>
<td>68</td>
<td>73</td>
<td>78</td>
<td>80</td>
<td>96</td>
<td>105</td>
<td>111</td>
<td>116</td>
</tr>
<tr>
<td>$L_0.25$-bs</td>
<td>22</td>
<td>37</td>
<td>46</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>62</td>
<td>68</td>
<td>71</td>
<td>74</td>
<td>75</td>
<td>90</td>
<td>102</td>
<td>105</td>
<td>109</td>
</tr>
<tr>
<td>$L_0.5$-bs</td>
<td>24</td>
<td>37</td>
<td>50</td>
<td>59</td>
<td>60</td>
<td>64</td>
<td>70</td>
<td>76</td>
<td>77</td>
<td>77</td>
<td>89</td>
<td>107</td>
<td>111</td>
<td>113</td>
<td></td>
</tr>
<tr>
<td>$L_0.75$-bs</td>
<td>27</td>
<td>44</td>
<td>52</td>
<td>65</td>
<td>69</td>
<td>73</td>
<td>74</td>
<td>80</td>
<td>82</td>
<td>98</td>
<td>108</td>
<td>114</td>
<td>115</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>methods</th>
<th>LagSA</th>
<th>LagSA-BB</th>
<th>$L_0.25$-bs</th>
<th>$L_0.5$-bs</th>
<th>$L_0.75$-bs</th>
</tr>
</thead>
<tbody>
<tr>
<td># of winners</td>
<td>28</td>
<td>33</td>
<td>47</td>
<td>49</td>
<td>62</td>
</tr>
<tr>
<td>LagSA : i</td>
<td>107</td>
<td>47</td>
<td>—</td>
<td>88</td>
<td>66</td>
</tr>
<tr>
<td>LagSA-BB : i</td>
<td>107</td>
<td>47</td>
<td>—</td>
<td>88</td>
<td>66</td>
</tr>
<tr>
<td>$L_0.25$-bs : i</td>
<td>92</td>
<td>53</td>
<td>66</td>
<td>88</td>
<td>—</td>
</tr>
<tr>
<td>$L_0.5$-bs : i</td>
<td>98</td>
<td>49</td>
<td>76</td>
<td>79</td>
<td>96</td>
</tr>
<tr>
<td>$L_0.75$-bs : i</td>
<td>101</td>
<td>46</td>
<td>88</td>
<td>71</td>
<td>99</td>
</tr>
</tbody>
</table>

### 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 $N_{\text{best}}$ 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

<table>
<thead>
<tr>
<th>Problem</th>
<th>$L_{0.75}$</th>
<th>$L_{0.75}$-negProx</th>
<th>$L_{0.75}$-CP</th>
<th>$L_{0.75}$-CP-negProx</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Rgap</td>
<td>gap</td>
<td>time</td>
<td>gap</td>
</tr>
<tr>
<td>bur26a</td>
<td>9.6</td>
<td>0.1212</td>
<td>2.6</td>
<td>0.0889</td>
</tr>
<tr>
<td>bur26b</td>
<td>10.4</td>
<td>0.1769</td>
<td>3.0</td>
<td>0.1716</td>
</tr>
<tr>
<td>bur26d</td>
<td>10.2</td>
<td>0.0020</td>
<td>3.6</td>
<td>0.0020</td>
</tr>
<tr>
<td>bur26e</td>
<td>10.3</td>
<td>0.0091</td>
<td>2.2</td>
<td>0.0000</td>
</tr>
<tr>
<td>chr15c</td>
<td>546.6</td>
<td>30.9554</td>
<td>3.9</td>
<td>30.9554</td>
</tr>
<tr>
<td>chr20c</td>
<td>656.5</td>
<td>18.2152</td>
<td>1.7</td>
<td>18.2152</td>
</tr>
<tr>
<td>esc32b</td>
<td>171.0</td>
<td>9.5238</td>
<td>1.8</td>
<td>19.9</td>
</tr>
<tr>
<td>had20</td>
<td>12.2</td>
<td>0.0289</td>
<td>0.5</td>
<td>0.0000</td>
</tr>
<tr>
<td>kro30a</td>
<td>51.5</td>
<td>0.8999</td>
<td>1.1</td>
<td>1.3498</td>
</tr>
<tr>
<td>lip20a</td>
<td>7.0</td>
<td>2.0092</td>
<td>0.1</td>
<td>0.0000</td>
</tr>
<tr>
<td>mg14</td>
<td>34.5</td>
<td>0.1972</td>
<td>0.3</td>
<td>0.0000</td>
</tr>
<tr>
<td>mg18</td>
<td>32.9</td>
<td>0.1415</td>
<td>0.5</td>
<td>0.0000</td>
</tr>
<tr>
<td>mg28</td>
<td>34.2</td>
<td>0.1549</td>
<td>1.3</td>
<td>0.0000</td>
</tr>
<tr>
<td>scr12</td>
<td>89.2</td>
<td>2.3368</td>
<td>0.1</td>
<td>0.0000</td>
</tr>
<tr>
<td>scr20</td>
<td>105.7</td>
<td>0.6525</td>
<td>0.3</td>
<td>0.0000</td>
</tr>
<tr>
<td>ste36b</td>
<td>439.0</td>
<td>4.5672</td>
<td>0.7</td>
<td>1.1734</td>
</tr>
<tr>
<td>tai20b</td>
<td>163.6</td>
<td>0.4526</td>
<td>0.3</td>
<td>0.4526</td>
</tr>
<tr>
<td>tai30a</td>
<td>21.0</td>
<td>1.7887</td>
<td>2.2</td>
<td>1.7887</td>
</tr>
<tr>
<td>tai64c</td>
<td>59.2</td>
<td>0.0926</td>
<td>5.3</td>
<td>0.0000</td>
</tr>
<tr>
<td>Nbest</td>
<td>0</td>
<td>17</td>
<td>5</td>
<td>18</td>
</tr>
</tbody>
</table>

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 $\text{nr\_iterations} = 1500*n; \text{nr\_resolutions} = 10$; for instances with $n \geq 80$, we set $\text{nr\_iterations} = 500*n; \text{nr\_resolutions} = 10$. 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

<table>
<thead>
<tr>
<th>Problem</th>
<th>$L_{0.75}$</th>
<th>$L_{0.75}$-negProx</th>
<th>$L_{0.75}$-CP</th>
<th>$L_{0.75}$-CP-negProx</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>gap</td>
<td>time</td>
<td>gap</td>
<td>time</td>
</tr>
<tr>
<td>lipa80a</td>
<td>2.1</td>
<td>0.7540</td>
<td>3.6</td>
<td>0.7168</td>
</tr>
<tr>
<td>lipa90a</td>
<td>1.9</td>
<td>0.7179</td>
<td>7.7</td>
<td>0.6827</td>
</tr>
<tr>
<td>sko100b</td>
<td>17.4</td>
<td>0.0858</td>
<td>13.9</td>
<td>0.0832</td>
</tr>
<tr>
<td>sko100d</td>
<td>17.5</td>
<td>0.1712</td>
<td>97.0</td>
<td>0.1712</td>
</tr>
<tr>
<td>sko100e</td>
<td>18.2</td>
<td>0.0134</td>
<td>14.0</td>
<td>0.0094</td>
</tr>
<tr>
<td>sko100f</td>
<td>17.3</td>
<td>0.0550</td>
<td>12.3</td>
<td>0.0550</td>
</tr>
<tr>
<td>sko81</td>
<td>19.2</td>
<td>0.1143</td>
<td>12.2</td>
<td>0.1043</td>
</tr>
<tr>
<td>tai100b</td>
<td>50.9</td>
<td>0.3800</td>
<td>13.7</td>
<td>0.2237</td>
</tr>
<tr>
<td>tai150b</td>
<td>30.6</td>
<td>0.5908</td>
<td>86.2</td>
<td>0.1422</td>
</tr>
<tr>
<td>tai256c</td>
<td>19.8</td>
<td>0.2610</td>
<td>24.6</td>
<td>0.1122</td>
</tr>
<tr>
<td>tai80a</td>
<td>15.6</td>
<td>0.6904</td>
<td>20.0</td>
<td>0.4734</td>
</tr>
<tr>
<td>tai80b</td>
<td>51.9</td>
<td>0.0378</td>
<td>6.7</td>
<td>0.0380</td>
</tr>
<tr>
<td>tho150</td>
<td>20.6</td>
<td>0.1500</td>
<td>53.4</td>
<td>0.1302</td>
</tr>
<tr>
<td>wil100</td>
<td>9.8</td>
<td>0.0146</td>
<td>21.0</td>
<td>0.0146</td>
</tr>
</tbody>
</table>

$L_{0.75}$ 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

<table>
<thead>
<tr>
<th>Problem</th>
<th>$L_{0.75}$</th>
<th>Ro-TS, 10 runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>gap time</td>
<td>gap time</td>
</tr>
<tr>
<td>bur26c</td>
<td>9.5</td>
<td>0.7</td>
</tr>
<tr>
<td>bur26f</td>
<td>11.3</td>
<td>1.2</td>
</tr>
<tr>
<td>bur26g</td>
<td>9.9</td>
<td>0.6</td>
</tr>
<tr>
<td>bur26h</td>
<td>10.9</td>
<td>0.6</td>
</tr>
<tr>
<td>chr12a</td>
<td>372.5</td>
<td>0.3</td>
</tr>
<tr>
<td>chr12b</td>
<td>363.4</td>
<td>0.2</td>
</tr>
<tr>
<td>chr18b</td>
<td>199.9</td>
<td>1.0</td>
</tr>
<tr>
<td>esc16a</td>
<td>63.3</td>
<td>0.4</td>
</tr>
<tr>
<td>esc16b</td>
<td>7.9</td>
<td>3.8</td>
</tr>
<tr>
<td>esc16c</td>
<td>55.8</td>
<td>0.4</td>
</tr>
<tr>
<td>esc16d</td>
<td>226.0</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Continued on next page
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 \( L_{0.75} \).
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

<table>
<thead>
<tr>
<th>Problem name</th>
<th>L_{0.75}</th>
<th>L_{0.75}-Ro-TS, 10 runs</th>
<th>Ro-TS, 10 runs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time gap</td>
<td>gap time nfe</td>
<td>gap (%) time</td>
</tr>
<tr>
<td></td>
<td>(min, mean, max)</td>
<td>(min, mean, max)</td>
<td></td>
</tr>
<tr>
<td>esc128</td>
<td>397.6</td>
<td>0.0000 19.8 390</td>
<td></td>
</tr>
<tr>
<td>lipa80a</td>
<td>2.1</td>
<td>0.7540 0.4 37</td>
<td></td>
</tr>
<tr>
<td>lipa80b</td>
<td>30.8</td>
<td>0.0000 3.1 163</td>
<td></td>
</tr>
<tr>
<td>lipa90a</td>
<td>1.9</td>
<td>0.7179 0.9 37</td>
<td></td>
</tr>
<tr>
<td>lipa90b</td>
<td>30.5</td>
<td>0.0000 2.2 84</td>
<td></td>
</tr>
<tr>
<td>sko100a</td>
<td>17.4</td>
<td>0.0645 14.5 396</td>
<td></td>
</tr>
<tr>
<td>sko100b</td>
<td>17.4</td>
<td>0.0858 13.9 371</td>
<td></td>
</tr>
<tr>
<td>sko100c</td>
<td>18.0</td>
<td>0.0203 16.8 464</td>
<td></td>
</tr>
<tr>
<td>sko100d</td>
<td>17.5</td>
<td>0.1712 13.9 381</td>
<td></td>
</tr>
<tr>
<td>sko100e</td>
<td>18.2</td>
<td>0.0134 14.0 298</td>
<td></td>
</tr>
<tr>
<td>sko100f</td>
<td>17.3</td>
<td>0.0550 12.3 328</td>
<td></td>
</tr>
<tr>
<td>sko81</td>
<td>19.2</td>
<td>0.1143 12.2 535</td>
<td></td>
</tr>
<tr>
<td>sko90</td>
<td>18.5</td>
<td>0.0554 14.7 526</td>
<td></td>
</tr>
<tr>
<td>tai100a</td>
<td>14.4</td>
<td>0.4131 33.8 968</td>
<td></td>
</tr>
<tr>
<td>tai100b</td>
<td>59.9</td>
<td>0.3800 13.7 400</td>
<td></td>
</tr>
<tr>
<td>tai150b</td>
<td>30.6</td>
<td>0.5098 86.2 1220</td>
<td></td>
</tr>
<tr>
<td>tai256c</td>
<td>19.8</td>
<td>0.2610 24.6 191</td>
<td></td>
</tr>
<tr>
<td>tai80a</td>
<td>15.6</td>
<td>0.6904 20.0 894</td>
<td></td>
</tr>
<tr>
<td>tai80b</td>
<td>51.9</td>
<td>0.0378 6.7 329</td>
<td></td>
</tr>
<tr>
<td>thio150</td>
<td>20.6</td>
<td>0.1500 53.4 679</td>
<td></td>
</tr>
<tr>
<td>will100</td>
<td>9.8</td>
<td>0.0146 21.0 542</td>
<td></td>
</tr>
<tr>
<td>N_{best}</td>
<td>8</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

### 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 $b(A) = \max_{i,j \neq 0}|i - j|$. The BM problem is to find a permutation matrix $X$ such that the matrix $X^T A X$ has the minimum bandwidth. Define $A_{ij} = 1$ if $A_{ij} \neq 0$ and $A_{ij} = 0$ otherwise. Clearly, $X^T A X$ has the same bandwidth with $X^T A X$. 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

$$h(m) = \min_{X \in \mathbb{R}^{n \times n}} \text{tr}(X^T A X B_m).$$

Then, $h(m) = 0$ if and only if the bandwidth of $X^T A X$ 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,

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

The BM problem can also be defined for a given graph. More specifically, for a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ being the vertices and $E$ being the edges, the BM problem is to find a bijection $\phi : V \rightarrow N$ such that the number $\max_{(i,j) \in E} |\phi(v_i) - \phi(v_j)|$ is minimal. Denote the adjacency matrix of $G$ by $A_G$. Then the BM problem for
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 $m = 0$, $\overline{m} = n - 1$ and the integer $m_0 \in (1, n-1)$, $k = 0$.

2. while $\overline{m} - m > 1$ do
3.     Compute $h(m_k)$ by Algorithm 2.
4.     if $h(m_k) > 0$ then $m = m_k$;
5.     if $h(m_k) = 0$ then $\overline{m} = m_k$;
6.     Set $m_k = \frac{1}{2}(m + \overline{m})$ 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 “imprRevCMKcK” (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” 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 “Nbest” 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 “~” 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
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 \texttt{“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>
<thead>
<tr>
<th>Problem</th>
<th>n</th>
<th>bw</th>
<th>Bi-$L_{0.75}$</th>
<th>irCM</th>
<th>irCM+</th>
<th>Bi-RoTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bw</td>
<td></td>
<td>bw time</td>
<td>bw time</td>
<td>bw run</td>
<td>bw time</td>
</tr>
<tr>
<td>GD06 specially</td>
<td>101</td>
<td>57</td>
<td>34 39.8</td>
<td>35 15.3</td>
<td>35 10266</td>
<td>34 65.1</td>
</tr>
<tr>
<td>GD97 specially</td>
<td>84</td>
<td>23</td>
<td>18 9.8</td>
<td>18 1.1</td>
<td>17 36174</td>
<td>17 47.6</td>
</tr>
<tr>
<td>JD97 specially</td>
<td>20</td>
<td>88</td>
<td>34 12.2</td>
<td>32 12.2</td>
<td>31 12569</td>
<td>28 111.0</td>
</tr>
<tr>
<td>Journals</td>
<td>124</td>
<td>116</td>
<td>98 77.1</td>
<td>102 4.2</td>
<td>102 73658</td>
<td>103 151.5</td>
</tr>
<tr>
<td>Sandi specially</td>
<td>86</td>
<td>22</td>
<td>12 17.3</td>
<td>12 9.6</td>
<td>12 7177</td>
<td>13 36.0</td>
</tr>
<tr>
<td>Trefethen's150</td>
<td>150</td>
<td>79</td>
<td>67 91.4</td>
<td>71 1.3</td>
<td>70 289009</td>
<td>79 263.2</td>
</tr>
<tr>
<td>Trefethen's200</td>
<td>200</td>
<td>99</td>
<td>87 167.7</td>
<td>88 1.5</td>
<td>88 461919</td>
<td>99 678.5</td>
</tr>
<tr>
<td>Trefethen's300b</td>
<td>199</td>
<td>99</td>
<td>86 243.1</td>
<td>87 1.5</td>
<td>87 643423</td>
<td>99 667.1</td>
</tr>
<tr>
<td>Trefethen's300c</td>
<td>300</td>
<td>152</td>
<td>132 652.5</td>
<td>132 1.2</td>
<td>132 817720</td>
<td>150 3973.7</td>
</tr>
<tr>
<td>adjoun</td>
<td>310</td>
<td>67</td>
<td>38 32.9</td>
<td>42 2.8</td>
<td>42 9225</td>
<td>42 975.7</td>
</tr>
<tr>
<td>ash292</td>
<td>292</td>
<td>32</td>
<td>23 316.3</td>
<td>21 26.6</td>
<td>21 47741</td>
<td>32 3718.3</td>
</tr>
<tr>
<td>ash85</td>
<td>85</td>
<td>13</td>
<td>10 10.0</td>
<td>10 0.6</td>
<td>10 70416</td>
<td>9 47.3</td>
</tr>
<tr>
<td>bcspwr03</td>
<td>118</td>
<td>17</td>
<td>12 19.6</td>
<td>11 11.8</td>
<td>11 6459</td>
<td>17 193.5</td>
</tr>
<tr>
<td>bcspwr04</td>
<td>274</td>
<td>49</td>
<td>31 304.1</td>
<td>29 207.1</td>
<td>29 5876</td>
<td>49 2509.4</td>
</tr>
<tr>
<td>bcstsk03</td>
<td>112</td>
<td>3</td>
<td>3 3.8</td>
<td>3 0.3</td>
<td>3 54309</td>
<td>3 162.2</td>
</tr>
<tr>
<td>bcstsk04</td>
<td>132</td>
<td>54</td>
<td>37 52.4</td>
<td>40 3.5</td>
<td>40 58609</td>
<td>38 184.2</td>
</tr>
<tr>
<td>bcstsk05</td>
<td>153</td>
<td>24</td>
<td>20 41.8</td>
<td>20 4.7</td>
<td>20 35781</td>
<td>24 294.0</td>
</tr>
<tr>
<td>bcstsk22</td>
<td>138</td>
<td>12</td>
<td>11 63.5</td>
<td>10 2.4</td>
<td>10 102888</td>
<td>12 158.5</td>
</tr>
<tr>
<td>can'144</td>
<td>144</td>
<td>18</td>
<td>13 25.9</td>
<td>13 8.5</td>
<td>13 12159</td>
<td>18 247.4</td>
</tr>
<tr>
<td>can'161</td>
<td>161</td>
<td>18</td>
<td>18 30.1</td>
<td>19 1.3</td>
<td>18 90297</td>
<td>18 258.9</td>
</tr>
<tr>
<td>can'187</td>
<td>187</td>
<td>23</td>
<td>14 62.0</td>
<td>15 2.0</td>
<td>15 123730</td>
<td>23 546.6</td>
</tr>
<tr>
<td>can'229</td>
<td>229</td>
<td>37</td>
<td>32 155.4</td>
<td>32 16.1</td>
<td>32 39029</td>
<td>37 1041.5</td>
</tr>
<tr>
<td>can'256</td>
<td>256</td>
<td>123</td>
<td>67 405.7</td>
<td>62 249.2</td>
<td>62 6561</td>
<td>85 1179.5</td>
</tr>
<tr>
<td>can'268</td>
<td>268</td>
<td>98</td>
<td>57 359.7</td>
<td>53 334.7</td>
<td>53 4256</td>
<td>82 1836.2</td>
</tr>
<tr>
<td>can'292</td>
<td>292</td>
<td>76</td>
<td>40 716.8</td>
<td>50 50.2</td>
<td>50 56876</td>
<td>76 628.3</td>
</tr>
<tr>
<td>can'96</td>
<td>96</td>
<td>23</td>
<td>13 16.0</td>
<td>11 0.6</td>
<td>11 99598</td>
<td>13 39.5</td>
</tr>
<tr>
<td>dwt'162</td>
<td>162</td>
<td>16</td>
<td>13 72.9</td>
<td>14 2.8</td>
<td>14 102152</td>
<td>16 174.3</td>
</tr>
<tr>
<td>dwt'193</td>
<td>193</td>
<td>54</td>
<td>32 145.3</td>
<td>34 56.0</td>
<td>34 10505</td>
<td>42 891.7</td>
</tr>
<tr>
<td>dwt'198</td>
<td>198</td>
<td>13</td>
<td>8 70.5</td>
<td>10 3.4</td>
<td>10 83262</td>
<td>13 488.2</td>
</tr>
<tr>
<td>dwt'209</td>
<td>209</td>
<td>33</td>
<td>26 97.9</td>
<td>29 12.8</td>
<td>29 30822</td>
<td>33 773.0</td>
</tr>
<tr>
<td>dwt'221</td>
<td>221</td>
<td>15</td>
<td>14 97.8</td>
<td>14 2.8</td>
<td>14 139580</td>
<td>15 1152.0</td>
</tr>
<tr>
<td>dwt'234</td>
<td>234</td>
<td>24</td>
<td>12 110.3</td>
<td>12 13.5</td>
<td>12 32277</td>
<td>24 990.3</td>
</tr>
<tr>
<td>dwt'245</td>
<td>245</td>
<td>55</td>
<td>28 267.1</td>
<td>31 132.3</td>
<td>31 8100</td>
<td>53 1609.1</td>
</tr>
<tr>
<td>dwt'87</td>
<td>87</td>
<td>18</td>
<td>12 9.1</td>
<td>12 4.0</td>
<td>12 9072</td>
<td>11 38.6</td>
</tr>
<tr>
<td>football</td>
<td>115</td>
<td>76</td>
<td>37 43.8</td>
<td>42 10.6</td>
<td>42 16866</td>
<td>39 157.3</td>
</tr>
<tr>
<td>grid1</td>
<td>252</td>
<td>12</td>
<td>22 409.5</td>
<td>19 0.7</td>
<td>19 2206545</td>
<td>22 1799.5</td>
</tr>
<tr>
<td>grid1_dual</td>
<td>224</td>
<td>17</td>
<td>17 162.2</td>
<td>17 0.7</td>
<td>17 984859</td>
<td>17 1195.5</td>
</tr>
<tr>
<td>jazz</td>
<td>198</td>
<td>141</td>
<td>69 162.2</td>
<td>81 105.1</td>
<td>81 6188</td>
<td>88 324.7</td>
</tr>
<tr>
<td>lshp265</td>
<td>265</td>
<td>18</td>
<td>18 129.2</td>
<td>17 0.8</td>
<td>17 612015</td>
<td>18 951.9</td>
</tr>
<tr>
<td>lund_a</td>
<td>147</td>
<td>23</td>
<td>23 42.4</td>
<td>23 0.4</td>
<td>23 396257</td>
<td>23 263.5</td>
</tr>
<tr>
<td>lund_b</td>
<td>147</td>
<td>23</td>
<td>23 49.5</td>
<td>23 0.4</td>
<td>23 462959</td>
<td>23 260.3</td>
</tr>
<tr>
<td>mesh3d1</td>
<td>289</td>
<td>17</td>
<td>17 192.7</td>
<td>17 0.9</td>
<td>17 839178</td>
<td>17 2368.1</td>
</tr>
<tr>
<td>mesh3d5</td>
<td>289</td>
<td>17</td>
<td>17 189.4</td>
<td>17 0.9</td>
<td>17 824889</td>
<td>17 2367.0</td>
</tr>
<tr>
<td>nose1</td>
<td>237</td>
<td>9 3 74.8</td>
<td>3 1.1</td>
<td>3 156008</td>
<td>4 1706.8</td>
<td></td>
</tr>
<tr>
<td>nose2</td>
<td>100</td>
<td>12</td>
<td>10 9.9</td>
<td>10 4.2</td>
<td>10 9422</td>
<td>11 97.1</td>
</tr>
<tr>
<td>polbooks</td>
<td>105</td>
<td>30</td>
<td>20 17.7</td>
<td>21 12.2</td>
<td>21 5883</td>
<td>22 112.4</td>
</tr>
<tr>
<td>spaceStation</td>
<td>99</td>
<td>55</td>
<td>28 24.1</td>
<td>33 3.9</td>
<td>33 25099</td>
<td>30 90.5</td>
</tr>
<tr>
<td>sphere3</td>
<td>258</td>
<td>32</td>
<td>30 328.0</td>
<td>28 1.7</td>
<td>27 744388</td>
<td>32 1649.3</td>
</tr>
<tr>
<td>tumorAntiAngiogenesis_1</td>
<td>205</td>
<td>199</td>
<td>100 170.6</td>
<td>100 67.0</td>
<td>100 10212</td>
<td>101 542.7</td>
</tr>
</tbody>
</table>

| $N_{\text{best}}$ | 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 $D_n$ (or the restricted domain $A$). 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

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

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.

Acknowledgements. We thank Yong Xia for the valuable discussions on the quadratic assignment problem and for sharing his code LagSA. We also thank Renata Sotirov for the helpful discussions on the bandwidth minimization problem and for sharing her code imprRevCMcK. The authors are grateful to the Associate Editor Alexandre d’Aspremont and two anonymous referees for their detailed and valuable comments and suggestions.

REFERENCES


