# CONTINUOUS METHODS QUADRATIC ASSIGNMENT PROBLEM

JOHN M. CONROY, STEVEN G. KRATZER
AND LOUIS J. PODRAZIK

Abstract. In this note we propose and analyze continuous methods to solve quadratic assignment problems. We formulate the problem as a quadratic programming problem with non-negativity and linear constraints. We investigate the Frank-Wolfe algorithm to solve this quadratic programming problem. To demonstrate that this simple approach is competitive we present the results of using this method to solve standard test cases of the quadratic assignment problem, an NP-hard problem. As the problem is global optimization a multiple start strategy is presented. Using the QAP library test set the proposed method finds solutions with a median performance of withing 5% of the best solution with 2 random starts.

## 1. Introduction

Linear assignment (LAP) and quadratic assignment problems (QAP) arise frequently in facility location problems in operations research (OR). The algorithm given in this paper are motivated by a matrix formulation of assignment problems. This formulation, we will see later are equivalent to the common formulations found in OR literature [?]. Informally, a linear assignment problem is to find a permutation such that when the columns of $A$ are permuted the resulting matrix is as near as possible to $B$, given two $m \times n$ matrices $A$ and $B$. We now define some notation, which will allow us to formally define the LAP and QAP and expose some of the structure of these problems.

**Definition 1.1** (Notation).

(1) $A_F = \left(\sum_{i,j} A_{i,j}^2\right)^{\frac{1}{2}}$
(2) *Hadamard product of two matrices* $A, B$: $(A \circ B)_{ij} = a_{ij} b_{ij}$.
(3) $\Pi_n \subsetneq E^{n\times n}$ *space of permutation matrices, the set of zero one matrices with exactly one 1 in each row and column.*
(4) $\Pi_{n,m} \subsetneq E^{n\times m}$ *space of partial permutation matrices*
(5) $\mathcal{D}_n \subsetneq E^{n\times n}$ *space of doubly stochastic matrices*

We now give the definition of the assignment problem, which is a special type of linear program.

**Definition 1.2** (Assignment Problem (AP)). *Given* $C \in E^{n\times n}$

$$\max_{X\in\mathcal{D}_n} \sum_{i,j=1}^{n} c_{ij} x_{ij} = \max_{X\in\mathcal{D}_n} \sum_{i,j=1}^{n} (C \circ X)_{ij}$$

We note that the LAP as defined is to find a doubly stochastic matrix which maximizes the above sum. Since the objective function is linear we know that the optimum will occur on a vertex of constraints set. The set vertices of the doubly stochastic matrices, $\mathcal{D}_n$, is the set of permutation matrices, $\Pi_n$.

Conversely, the constraint set for the quadratic assignment problem (QAP) will be the $\Pi_{n,m}$. Later in the paper we will considered a relaxed version of the QAP which has $\mathcal{D}_{n,m}$ as the constraint set.

**Definition 1.3** (Quadratic Assignment Problem (QAP))**.** *Given $A \in E^{n\times n}$, $B \in E^{m\times m}$*

$$\min_{X \subsetneq \Pi_{n,m}} \sum_{i,j=1}^{n} \sum_{k,l=1}^{m} a_{ij} b_{kl} x_{ik} x_{jl} = \min_{X \subsetneq \Pi_{n,m}} \sum_{i=1}^{n} \sum_{j=1}^{m} (AX \circ XB)_{ij}$$

In the next section we reformulate the LAP and the QAP as matrix optimization problems.

## 2. The Matrix Procrustes Problem & Preliminaries

We now need to define a bit more notation which we will need to help reformulate the LAP and QAP.

**Definition 2.1** (Notation)**.**

(1) *Given a matrix $A \in E^{n\times m}$, the stacking operator $s(.)$ creates an $mn$-vector $s(A)$ obtained by stacking the columns of $A$ on top of each other.*
(2) *The $m$-vector of all ones is $e_m$.*
(3) *$I_m$ denotes an identity matrix of size $m$.*
(4) *Given $x \in E^n$, the diagonal matrix whose diagonal elements are $x$ is denoted $diag(x)$.*

We now consider the one-sided permutation constrained matrix Procrustes problem and observe the well-known result that it is equivalent to solving a linear assignment problem in the case of permutation matrix constraints.

**Definition 2.2** (One-sided Matrix Procrustes Problem)**.** *Given $A, B \in E^{m\times n}$, find the permutation matrix $X \in E^{m\times m}$ to*

$$\min_{X \in \Pi_m} A - XB_F.$$

**Lemma 2.1.** *The one-sided matrix permutation constrained Procrustes problem is equivalent to solving a linear assignment problem (LAP) with cost assignment matrix $AB^t$, that is,*

$$\min_{X \in \Pi_m} A - XB_F \equiv \max_{X \in \Pi_m} tr(AB^t X^t) = \max_{X \in \Pi_m} \sum_{i,j=1}^{m} \left(AB^t \circ X\right)_{ij}.$$

*Proof: Since $X^t X = I$, we have*

$$\begin{aligned} A - XB_F^2 &= tr\left((A - XB)^t (A - XB)\right) \\ &= A_F^2 + B_F^2 - 2tr(AB^t X^t). \end{aligned}$$

*Furthermore,*

$$\begin{aligned} tr(AB^tX^t) &= tr(I_n(AB^t)I_nX^t) \\ &= \langle e_m, (AB^t \circ X)e_m\rangle = \sum_{i,j=1}^{m} \left(AB^t \circ X\right)_{ij}. \quad \square \end{aligned}$$

Similarly, we have

**Lemma 2.2.** *The quadratic assignment problem is equivalent to the following minimization problem*

$$\min_{X\in\Pi_m} A - XBX^t_F.$$

In order to formulate the QAP as a constrained optimization we and to efficiently compute the gradient $\nabla f(P)$, we further exploit the problem structure.

**Lemma 2.3.** *Given $A, B$, and $P$, the QAP objective function can be written as*

$$\begin{aligned} f(P) &= \sum_{i=1}^{m}\sum_{j=1}^{n} \left(A \circ PBP^t\right)_{ij} \\ &= \sum_{i,j=1}^{m} \left(APB^t \circ P\right)_{ij} = \sum_{i,j=1}^{n} \left(A^tPB \circ P\right)_{ij} \\ &= \langle s(P), (B \otimes A)s(P)\rangle. \end{aligned}$$

*Consequently, the gradient $\nabla f = (\nabla_P f) \in (E^{m\times m}, E^{n\times n})$ can be computed as*

$$\nabla_P f(P) = APB^t + A^tPB.$$

*Proof: From Lemma 2.6,*

$$\begin{aligned} f(P) &= tr\left(\ APB^tP^t\ \right) \\ &= tr\left(I_nAI_n(PBP^t)^t\right) \\ &= \langle e_m, (A \circ PBP^t)e_n\rangle = \langle s(PBP^t), s(A)\rangle \\ &= \langle (P \otimes P)s(B), s(A)\rangle = \langle (P \otimes I_n)(I_n \otimes P)s(B), s(A)\rangle \\ &= \langle (I_n \otimes P)s(B), (P^t \otimes I_n)s(A)\rangle = \langle s(PBI_n), s(I_nAP)\rangle \\ &= \langle s(PB), s(AP)\rangle \\ &= \langle s(I_nPB), s(API_n)\rangle = \langle (B^t \otimes I_n)s(P), (I_n \otimes A)s(P)\rangle \\ &= \langle s(P), (B \otimes I_n)(I_n \otimes A)s(P)\rangle \\ &= \langle s(P), (B \otimes A)s(P)\rangle = \langle s(P), s(APB^t)\rangle \\ &= \langle (B^t \otimes A^t)s(P), s(P)\rangle = \langle s(A^tPB), s(P)\rangle. \quad \square \end{aligned}$$

The Frank-Wolfe method (FW) is an iterative method to approximately solve a quadratic programming problem. We note in passing that the Hessian of $f(P)$ is given by

$$\nabla^2 f(P) = B \otimes A + B^t \otimes A^t$$

and that the quadratic programming problem is in general not convex. FW takes a starting point $X^{(0)} \in \mathcal{D}$ and then proceeds by solving the LAP derived from a first order Taylor series expansion about $X^{(0)}$. The solution of this LAP defines the direction to search for the next iterate. An exact line search is used to find the next iterate, $X^{(1)}$ which is also a doubly stochastic matrix. The procedure is iterated until a desired accuracy is obtained or convergence.

2.0.1. *Frank-Wolfe Approach.* In view of the fact the solution to Problem 2.8 is a vertex, we now consider the Frank-Wolfe algorithm [xx].

**Algorithm 2.1** (Frank-Wolfe Algorithm (FW))**.** *Given* $X^{(1)} \in \mathcal{D}_m$.
*Step 0: Let* $k = 1$.
*Step 1: Compute* $W^{(k)}$ *by solving the linear assignment problem:*

$$W^{(k)} = \arg \min_{W^{(k)}} \{ \sum_{i,j=1}^{m} \left( \nabla_X f(X^{(k)}) \circ W^{(k)} \right)_{ij} \mid W^{(k)} \in \mathcal{D}_m \}$$

*Step 2: Let* $d^{(k)} = W^{(k)} - X^{(k)}$.
*Step 3: Find* $\alpha^k \in [0,1]$ *such that*

$$\alpha^{(k)} = \arg \min_{\alpha^{(k)}} f(X^{(k)} + \alpha^{(k)} d^{(k)}).$$

*Step 4: Let*

$$X^{(k+1)} = X^{(k)} + \alpha^{(k)} d_X^{(k)}.$$

*Step 5: If* $d_P^{(k)} \le \epsilon$ *or* $\alpha^{(k)} = 0$ *or* $k = k_{\max}$ *then stop; else* $k = k + 1$ *and go to step 1.*

**Remark 2.1.**
*At each iteration,* $W^{(k)}$ *is an extreme point of the constraint set, a permutation matrix. However, in general, the iterates* $X^{(k)}$ *remain in the interior of the constraint set.*
*Step 3 is a "line search;" however, as the objective function is quadratic the optimum* $\alpha$ *can be found exactly.*
*Once FW has terminated the nearest permutation to the double stochastic matrix,* $X^{(k)}$ *can be found as given by the following Lemma.*

**Lemma 2.4.** *Given* $Y \in E^{n \times n}$*, the projection of* $Y$ *onto* $\Pi_n$ *can be obtained by solving the LAP*

$$W = \arg \min_W \{ W - Y_F \mid W \in \Pi_n \} \equiv \arg \max_W \{ tr(YW^t) \mid W \in \mathcal{D}_n \}.$$

*Proof: Since* $W \in \Pi_n \subset \mathcal{O}$*, we have*

$$\begin{aligned} W - Y_F^2 &= tr((W-Y)^t(W-Y)) \\ &= (Y_F^2 + I_F^2) - 2tr(YW^t). \quad \square \end{aligned}$$

2.1. **General QAP Test Problems.** To illustrate the effectiveness of FW we give the performance of the algorithm on 16 sample problems. The below table gives the optimum value found by FW with 1, 2, 3, and 100 random starting points and compared to the best known solution as given in column 2 and the Path algorithm [**?**], column 7. The starting point for FW, $X^{(0)}$ is chosen to be

$$X^{(0)} = \frac{1}{2n} + \frac{1}{2} S$$

where $S$ is a doubly stochastic matrix as computed by running Sinkhorn balancing on a uniform $[0,1]$ matrix.

Table 1. Comparison of Frank-Wolfe with Minimum Solution and Path Algorithm

| Problem | Min | $FW_{100}$ | $FW_3$ | $FW_2$ | $FW_1$ | Path |
|---:|---:|---:|---:|---:|---:|---:|
| chr12c | 11156 | 12176 | 13072 | 13072 | 13072 | 18048 |
| chr15a | 9896 | 9896 | 17272 | 17272 | 27584 | 19086 |
| chr15c | 9504 | 10960 | 14274 | 14274 | 17324 | 16206 |
| chr20b | 2298 | 2786 | 3068 | 3068 | 3068 | 5560 |
| chr22b | 6194 | 7218 | 7876 | 7876 | 8482 | 8500 |
| esc16b | 292 | 292 | 294 | 294 | 320 | 300 |
| rou12 | 235528 | 235528 | 238134 | 253684 | 253684 | 256320 |
| rou15 | 354210 | 356654 | 371458 | 371458 | 371458 | 391270 |
| rou20 | 725522 | 730614 | 743884 | 743884 | 743884 | 778284 |
| tai10a | 135028 | 135828 | 148970 | 157954 | 157954 | 152534 |
| tai15a | 388214 | 391522 | 397376 | 397376 | 397376 | 419224 |
| tai17a | 491812 | 496598 | 511574 | 511574 | 529134 | 530978 |
| tai20a | 703482 | 711840 | 721540 | 721540 | 734276 | 753712 |
| tai30a | 1818146 | 1844636 | 1890738 | 1894640 | 1894640 | 1903872 |
| tai35a | 2422002 | 2454292 | 2460940 | 2460940 | 2460940 | 2555110 |
| tai40a | 3139370 | 3187738 | 3194826 | 3194826 | 3227612 | 3281830 |