Quadratic Assignment Problem[1]

Sourav Yadav

May 2021

[1]This project was undertaken under the guidance of Dr Pawan Kumar Aurora.

Contents

Abstract

Quadratic Assignment Problem(QAP) is one of the most challenging problems in combinatorial optimization. The number of life problems that QAP can model is huge and is constantly increasing. It finds applications in placement problems, scheduling, manufacturing, VLSI design, statistical data analysis, and parallel and distributed computing. Also, most combinatorial optimization problems turn out to be special cases of QAP. There are various approaches to find a solution; one such is the polyhedral cutting approach. The polyhedral cutting approach is mainly unused for QAP due to few known facets. Now, we have two more such facets. This project aims to devise an efficient separation procedure for the previously known and the new facets. We present a simple yet more effective separation procedure than in the literature. We present our results and compare them to the previously known results in the literature using polyhedral cutting methods.

1 Introduction

The quadratic assignment problem is a challenging classical problem in the NP-hard combinatorial problems. Koopmans and Beckmann[1] first introduced it in 1962, and they modelled it with the following real-life problem:

> Consider a set of n facilities and a set of n locations. For each pair of locations, a distance is specified, and for each pair of facilities, a weight/flow is given. The problem is to assign all facilities to different locations to minimize the sum of distances multiplied by the corresponding flows.

Intuitively, the cost function motivates facilities with high flows to be close.

1.1 Problem Statement

Consider the set {1,2.....n} and two n×n matrices A and B. Then we define QAP(A,B) as follows:

$$\min_{\pi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{\pi(i)\pi(j)} b_{ij} \tag{1}$$

where S_n is the set of permutation of {1,2.....n}. This resembles the assignment problem but with a quadratic objective function, hence the name Quadratic assignment problem.

1.1.1 Other Formulations

The above formulation is due to Koopmans and Beckmann[1]. There are other formulations such as:

Lawler Formulation[2]: Given the set {1,2,.....n} and a 4-D n×n×n×n

matrix D. We define QAP(A,B) as,

$$\min_{\pi \in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{\pi(i)i\pi(j)j} \tag{2}$$

where S_n is the set of permutation of {1,2,....n}.

Trace Formulation: Given the set {1,2,.....n} and two n×n matrices A and B. QAP(A,B) is defined as follows:

$$\min_{X \in \Pi_n} tr(AXBX^t) \tag{3}$$

where Π_n represents set of all the permutation matrices of order n×n. This was introduced by Edwards[3] and turned out very helpful in deriving eigenvalue related lower bounds.

2 Integer program for QAP

All the formulations defined above rely on the one-to-one correpondance between the set of permutation of the set {1,2.....n} and the set of the permutation matrices defined as,

Definition 2.1 (Permutation Matrix)**.** Let X = (x_{ij}) be a n × n matrix. If the entries x_{ij} fulfill the following conditions

$$\begin{aligned} \sum_{i=1}^{n} x_{ij} &= 1 \qquad \forall\, 1 \le i \le n \\ \sum_{j=1}^{n} x_{ij} &= 1 \qquad \forall\, 1 \le j \le n \\ x_{ij} &\in 0, 1 \qquad \forall\, 1 \le i, j \le n \end{aligned} \tag{4}$$

then X is called a permutation matrix.

A second-order permutation matrix $P_\sigma^{[2]}$ [4] is defined as follows:

Definition 2.2 (Permutation Matrix of second order)**.** Let X = x_{ijkl} be a n × n × n × n matrix. If the entries x_{ijkl} fulfill the following conditions

$$\begin{aligned}
&x_{ijkl} - x_{klij} = 0 \qquad \forall\, 1 \le i,j,k,l \le n \\
&x_{ijil} = x_{jili} = 0 \qquad \forall 1 \le i,j \ne l \le n \\
&\sum_{k=1}^{n} x_{ijkl} = \sum_{k=1}^{n} x_{ijlk} = x_{ijij} \qquad \forall\, 1 \le i,j,l \le n \\
&\sum_{j=1}^{n} x_{ijij} = \sum_{j=1}^{n} x_{jiji} = 1 \qquad \forall\, 1 \le i \le n \\
&x_{ijkl} \in 0,1 \qquad \forall\, 1 \le i,j,k,l \le n
\end{aligned} \tag{5}$$

then X is called a second-order permutation matrix.

Consider the following ILP where A,B are the matrices in QAP(A,B).

IP-QAP(A,B): Minimize $\sum_{1 \le i,j,k,l \le n} A_{ik} \times B_{jl} \times X_{ijkl}$

subject to, X is a second-order Permutation Matrix

Theorem 1. IP-QAP(A,B) returns an optimal solution for QAP(A,B).

It was proved in [4] that the QAP polytope corresponds to the convex hull of all the second-order Permutation Matrices.

2.1 Linear Programming Relaxation

Since it's NP-hard to solve IP-QAP, we study it's linear programming relaxation, where the variable X_{ijkl} is allowed to take values between 0 and 1.

3 Polyhedral Approach

In polyhedral combinatorics, the quadratic assignment polytope is the convex hull of all the feasible solutions of IP-QAP. The relaxed quadratic assignment polytope is the convex hull of all the feasible solutions of LP-QAP. Naturally, the quadratic assignment polytope is embedded inside the relaxed quadratic assignment polytope.

Polyhedral cutting plane methods use classes of valid inequalities, which are known to be satisfied by all the feasible solutions of the original problem but not its relaxation. The finest of such inequalities must be the facets of the original problem.

At each iteration, we find an optimal solution for the relaxed problem. If it is also a solution to the original problem, then we are done. Otherwise, some of the inequalities mentioned above must be violated. In this case, one or more violated inequalities are added to the LP relaxation of the problem. The latter is solved again, and the whole process is repeated.

procedure POLYHEDRAL CUTTING PLANE METHOD

Input : Linear program L, Family of inequalities I

Step 1 : Solve L and let x be the optimal solution.

Step 2 : Check for violated inequalities in I.

Step 3 : If none found, quit.

Step 4 : Otherwise, add the violated inequalities to L and go to step 1.

end procedure

3.1 Branch and Cut

Branch and cut is where the Branch and bound approach meets the polyhedral cutting plane methods. This method solves the relaxed linear program

using standard methods. When the optimal solution is found, cutting plane methods are employed to improve the solution. After repeating some such iterations, the cutting plane methods will fail to produce a significant difference.

At this point, the branch and bound procedure is initiated. The problem is then split into multiple(usually two) different problems with disjoint solution spaces. In the branch and bound method, the solution to LP-relaxations serves as a lower bound, and the solutions to the respective integer programs serve as an upper bound. Branch and bound is only better than naive methods due to pruning. A node can be pruned if its lower bound is greater than an existing upper bound.

procedure BRANCH AND CUT

Input : Linear program L

Step 1 : Form a queue Q of problems, add L to Q

Step 2 : xopt = NULL, globalup = inf

while Q is not empty **do**

Step a : De-queue a problem from Q, let it be P.

Step b : Solve P.

Step c : If solution is infeasible, go back to Step a. Otherwise, let x be the solution and o be its objective value.

Step d : If o $\geq$ globalup, go back to Step a.

Step e : With the help of a heuristic, form a corresponding integer program for P. Call it P-int.

Step f : Solve P-int. Let x* be its solution and o* be it's objective value.

Step g : If o* $<$ globalup then xopt = x* and globalup = o*.

Step h : Start the cutting plane procedure. If found, add the violated inequalities to P and go back to Step b.

Step i : Branch to partition the problem into two new problems. Call them L_1 and L_2.

Step j : En-queue L_1 and L_2 to Q.

end while

Step 4 : return xopt and globalup.

end procedure

4 Known Facets

- QAP 1, A Trivial Facet

$$Y_{i,j,k,l} \geq 0 \qquad \forall i,j,k,l \tag{6}$$

- QAP 2

$$\sum_{r=1}^{m} Y_{i_r j_r,kl} - Y_{kl,kl} - \sum_{r<s} Y_{i_r j_r,i_s j_s} \leq 0 \tag{7}$$

where $i_1, ..., i_m, k$ are all distinct and $j_1, ..., j_m, l$ are also distinct. In addition n $\geq$ 6, m $\geq$ 3. These inequalities were introduced by Aurora and Mehta[4] who also proved them to be facet defining.

- QAP 3

$$(\beta - 1) \sum_{(ij)\epsilon P \times Q} Y_{ij,ij} - \sum_{(ij),(kl)\epsilon P \times Q i<k} Y_{ij,kl} \leq \frac{\beta^2 - \beta}{2} \tag{8}$$

where P,Q $\subset$ [n]. In addition (i) β+1 $\leq$ |P|, |Q| $\leq$ n-3, (ii) |P|+|Q| $\leq$ n-3+β, (iii) $\beta \geq 2$. These inequlities were introduced by Kaibel[5] who also proved they are facet defining for the QAP-polytope. They called them 1-Box Inequalities.

- QAP 4

$$\begin{aligned} -(\beta-1) \sum_{(ij)\epsilon P_1 \times Q} Y_{ij,ij} + \beta \sum_{(ij)\epsilon P_2 \times Q} Y_{ij,ij} + \sum_{i<k(ij),(kl)\epsilon P_1 \times Q} Y_{ij,kl} \\ + \sum_{i<k(ij),(kl)\epsilon P_2 \times Q} Y_{ij,kl} - \sum_{(ij)\epsilon P_1 \times Q (kl)\epsilon P_2 \times Q} Y_{ij,kl} - \frac{\beta^2 - \beta}{2} \geq 0 \end{aligned} \tag{9}$$

where P_1, P_2, Q $\subset$ [n], $P_1 \cap P_2 = \phi$. Futher, (i) $3 \leq$ |Q| $\leq$ n-1, (ii) $|P_1|$ + $|P_2| \leq$ n-1, (iii) $|P_1| \geq$ min2,β+1, (iv) $|P_2| \geq$ min1, -β +2, (v) $||P_1|$ - $|P_2|$ - $\beta| \leq$ n - |Q| - 4, (vi) if $|P_2| = 1$: |Q| $\geq$ min-β + 5, β + 2; if $|P_2|$ $\geq$ 2: |Q| $\geq$ min-β+5, β+3 or |Q| $\geq$ min-β+1, β+4. These inequlities were introduced by Kaibel[5] who also proved they are facet defining for the QAP-polytope. They called them 2-Box Inqualities.

- QAP 5 - 4-Box Inequalities. Introduced by Kaibel[5] and proved to be facet defining

- QAP 6

$$\sum_{r=1}^{m} Y_{i_r j_r, i_r j_r} - \sum_{r<s} Y_{i_r j_s, i_s j_s} \leq 1 \tag{10}$$

where $i_1, ..., i_m$ are all distinct and $j_1, ..., j_m$ are also distinct. In addition, m,n $\geq$ 7. These inequalitied were introduced by Aurora and Tiwary[6] and they proved them to be facet defining for the QAP-polytope.

QAP 2 and QAP 6 are new and we hope to use them in our separation procedure to find better bounds.

5 A Separation Heuristic for Kaibel Facets

The separation heuristic for QAP3 essentially involves choosing a random starting solution and then using a tabu search to optimize the solution for the given inequality(in this case, QAP3).

5.1 Tabu Search

Given a solution space S of solutions, tabu search works like any typical greedy search algorithm but with two significant changes. First, it can avoid stalling at a local optimum by moving even when no better neighbouring solution exists. Second, recent moves are forbidden in order to avoid cycling.

Let S be the solution space, with objective function (fitness) f : S $\rightarrow$ R. Further, there is a neighbourhood N : S $\rightarrow 2^S$ which, for each s $\in$ S defines a set N $\subset$ S called neighbours of s. A move is defined as a jump from the current solution to a neighbouring solution. Finally, there is a forbidden list of solutions to which, for the current solutions move is not allowed. This is called the tabu list. Note that the size of tabu list is critical for performance.

Tabu search starts with a random solution s and repeatedly moves to a neighbouring solution. At each iteration, the neighbouring set of the current solution is considered, and the move that improves the fitness (objective function) is chosen. If no moves improve the objective function, then the best among the neighbouring solutions is chosen. It effectively avoids stalling at the local optimum. Note that no moves to the tabu list will be considered.

5.2 Heuristic

The core of our heuristic is a tabu search that tries for a given value of beta and a set P to find a set Q that yield a large LHS value in QAP3 (equation 8). Since the right-hand side value of the equation is quadratic in beta, we restrict beta values to {2,3}.

Let k be the size of set Q. Then we define the follows:

Solution Space : Set of all the permutations of the set {1,2.....k}.

Left-hand side value of QAP2 :

$$L(P, Q, \beta, Y) = (\beta - 1) \sum_{ij \in P \times Q} Y_{ij,ij} - \sum_{(ij),(kl) \in P \times Q, i<k} Y_{ij,kl} \tag{11}$$

Fitness :

$$Fit(P, Q, \beta, Y) = L(P, Q, \beta, Y) \tag{12}$$

Neighbourhood : For a set s, N(s) consists of all such sets that differ from s at exactly one place. For Eg. s = {1,2} and n = 3 then N(s) = {{3,2}, {1,3}}.

Tabu list : We keep previous 5 solutions in the tabu list to avoid cycling.

Our heuristic first guesses some starting sets P and Q; it then keeps P the same and employs a tabu search with the above-defined attributes to optimize for Q. Maximum iterations and tabu list size are kept at 10 and 5, respectively. On average, it successfully finds a violated inequality on every other run.

The number of iterations that we run the separation procedure varies with the problem size. As mentioned already, QAP3, QAP4 and QAP5 were introduced by Volker Kaibel in his PhD thesis, and they noted that restricting to 1-box inequalities (QAP3) produces the best results. We, in our testing, found the same. Hence, the separation procedure is restricted to 1-box inequalities.

```
procedure Tabu Search
    Input: Linear problem L, Set P, Set Q, Int β, Solution Y
                                                    ▷ Y is a 4-D array
    bestCandidate ← Q
    List tabuList, Set best
    maxitr ← 10
    maxTabuSize ← 5
    i ← 0
    while i ≤ maxitr do
        Create a list neighbours
        neighbours ← getNeighbours(bestCandidate)
                              ▷ returns neighbourhood of the bestCandidate
        bestCandidate ← neighbours.front
        for candidate in neighbours do
            if candidate ∉ tabuList  and  Fit(L, P, candidate, β, Y) >
            Fit(L, P, bestCandidate, β, Y) then
                bestCandidate = candidate
            end if
            if Fit(L, P, bestCandidate, β, Y) > Fit(L, P, best, β, Y) then
                best = bestCandidate
            end if
        end for
    end while
    return  best
end procedure
```

procedure SEPARATION PROCEDURE1

Input : Linear Problem L, Solution Y, Int β ▷ Y is a 4-D array

cntLimit

limit $\leftarrow \frac{\beta^2-\beta}{2}$

for $i \leftarrow 1, cntLimit$ **do**

Generate two random sets P, Q, both of size $\leq$ n.

Assert $\beta + 1 \leq |P|, |Q| \leq n - 3$ and $|P| + |Q| \leq n - 3 + \beta$

Q $\leftarrow$ TabuSearch(L, P, Q,β)

if Fitness(L, P, Q, β, Y) $>$ limit **then**

Add the inequality to L

end if

end for

return L

end procedure

6 Heuristic for the other Facets

Now, we know two more facets of the QAP-polytope, QAP2 and QAP6. In our project, we were unable to find a suitable separation procedure for QAP6. This is because the minimum value of m,n must be greater than 7 making the negative valued terms quite large.

We have implemented a straightforward separation procedure for QAP2. Note that, larger the m, more is the number of negative valued terms on the left-hand side. Thus, we choose to extract violated inequalities with m=3 only. Though quite naive, it generally finds thousands of violated inequalities in a single run in relatively less time. However, the quality of those cuts does not seem on par with those provided by QAP2.

procedure SEPARATION PROCEDURE2

Input : Linear Problem P, Solution Y, Int n ▷ Y is a 4-D array.

for $k \leftarrow 1, n$ **do**

for $l \leftarrow 1, n$ **do**

1. Create a list L and add all $Y_{ij,kl}$ to L. ▷ $\forall i, j \in [1, n]$
2. Sort L.
3. Partition L into sets of 3 in increasing order. Test if they're distinct i.e. i_1, i_2 and i_3 are all distinct and j_1. j_2 and j_2 are all distinct.
4. If not distinct go to the next set. Otherwise, Let val be the left-hand side value of QAP6.
5. If val > 0, add the violated inequality to P. Otherwise, go to next such set.

end for

end for

return L.

7 Results

We have used instances of size n ≤ 20 from qaplib as the test set. We implemented the cutting plane procedure using the above separation algorithm. We used the Gurobi[7] barrier optimization method to solve the linear program; simplex took too much time to be practical.

Table 1, 2, and 3 show the results. Linear Bound stands for solution of the LP-relaxation, while the column 1-box represents the results presented by Kaibel using facets QAP3. Tabu-cuts represents the results from our separation procedure using QAP2 and QAP3.

name	Linear Bound	1-box	Tabu-cuts	Optimal
chr12a	9552.0	9552.0	9552.0	9552.0
chr12b	9742.0	9742.0	9742.0	9742.0
chr12c	11156.0	11156.0	11156.0	11156.0
had12	1621.5	1652.0	1652.0	1652.0
scr12	29827.0	31410.0	31410.0	31410.0
tai12a	222190.0	224416.0	224416.0	224416.0
tai12b	31697000.0	39464925.0	39464925.0	39464925.0
had14	2666.1	2724.0	2724.0	2724.0
chr15a	9513.1	9896.0	9896.0	9896.0
chr15b	7990.0	7990.0	7990.0	7990.0
chr15c	9504.0	9504.0	9504.0	9504.0
nug15	1041.0	1129.4	1137.5	1150.0
scr15	49265.0	51140.0	51140.0	51140.0
tai15a	352890.0	366465.9	368125.1	388214.0
tai15b	51559000.0	51765268.0	51765268.0	51765268.0

Table 1: Table for small instances

name	Linear Bound	1-box	Tabu-cuts	Optimal
esc16a	48.0	66.0	61.8*	68.0
esc16b	278.0	292.0	292.0	292.0
esc16c	118.0	160.0	160.0	160.0
esc16d	4.0	16.0	16.0	16.0
esc16e	14.0	28.0	28.0	28.0
esc16g	14.0	26.0	26.0	26.0
esc16h	704.0	996.0	996.0	996.0
esc16i	0.0	14.0	14.0	14.0
esc16j	2.0	8.0	8.0	8.0
had16	3560.2	3716.8	3720.0	3720.0
nug16a	1425.6	1567.0	1594.2	1610.0
nug16b	1088.2	1208.2	1224.4	1240.0
nug17	1505.8	1643.5	1677.6	1732.0
tai17a	442700.0	454625.1	460377.2	491812.0

Table 2: Table for medium instances

name	Linear Bound	1-box	Tabu-cuts	Optimal
chr18a	10738.5	10947.1	11098.0	11098.0
chr18b	1534.0	1534.0	1534.0	1534.0
had18	5087.9	5299.1	5330.3	5358.0
nug18	1663	1809.3	1825.8	1930.0
chr20a	2175.4	2172.4	2192.0	2192.0
chr20b	2287.0	2294.8	2298.0	2298.0
chr20c	14142.0	14033.2	14142.0	14142.0
had20	6578.8	6731.6	6867.7*	6922.0
lipa20a	3683.0	3683.0	3683.0	3683.0
lipa20b	27076.0	27076.0	27076.0	27076.0
nug20	2181.6	2313.5	2411.0	2570.0

Table 3: Table for large instances

*Denotes the instances which were stopped before terminating.

8 Conclusions

We set out to develop a branch and cut algorithm for the Quadratic Assignment Problem, in which we have failed. We have not presented our implementation of the Branch and Cut method due to the stability issues. We found that adding too many cuts sometimes introduces numerical instabilities, which may take tens of hours to solve if encountered at some node.

We mention the results presented by Volker Kaibel in his PhD thesis[8]. We have improved the results significantly for most instances. Though, in our case, we had enhanced computational powers, we still believe our separation procedure to be more versatile. In general, it can be adjusted for any other

inequality/cuts as well. For E.g. We implemented it for QAP2 but found it to be more costly than the naive one.

The project aimed to comprehend the effectiveness of the newfound facets of the QAP polytope. In our opinion, the new cuts do not improve the solution considerably well though they are much faster to implement. We again stand at the same problem with polyhedral cutting approaches, inadequate knowledge of the facets of the QAP polytope.

Another problem that we faced was with LP solvers. Although the state of the art LP solvers have improved considerably, they still fall short while solving for large instances (n>35). They are pretty sensitive to numerics and can take a tremendous amount of time in the crossover. Not to mention, we had our machines crash just creating the model files of some very large instances ($n \leq 60$); solving was even more challenging.

References

[1] T.C. Koopmans and M.J. Beckmann. Assignment problems and the location of economic activities. *Econometrica 25*, 1957.

[2] Eugene L. Lawler. The quadratic assignment problem. *Management Science*, 9(4):586–599, 1963.

[3] C.S. Edwards. The derivation of a greedy approximator for the koopmans-beckmann quadratic assignment problem. *Proceedings of the 77-th Combi- natorial Programming Conference (CP77)*, 1977.

[4] Pawan Aurora and Shashank Mehta. The qap-polytope and the graph isomorphism problem. *Journal of Combinatorial Optimization*, 36, 10 2018.

[5] Volker Kaibel. *Polyhedral Methods for the QAP*, pages 109–141. Springer US, Boston, MA, 2000.

[6] Pawan Aurora and Hans Raj Tiwary. On the complexity of some facet-defining inequalities of the qap-polytope. *CoRR*, abs/2010.06401, 2020.

[7] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021.

[8] V Kaibel. Polyhedral combinatorics of the quadratic assignment problem, ph.d. thesis, faculty of mathematics and natural sciences, university of cologne, 1997.