# Linearization of the Binary Quadratic Assignment Problem (QAP)

The Quadratic Assignment Problem (QAP) is a classic optimization problem that arises in various real-world scenarios, such as facility location and layout design. The objective of QAP is to minimize the total cost associated with assigning n facilities to n locations. The cost is modeled as a function of the flow between facilities and the distance between locations. Mathematically, this involves a quadratic objective function where the decision variables represent the assignments of facilities to locations.

Due to the quadratic nature of the problem, QAP is NP-hard and challenging to solve for large instances. This complexity motivates the use of the Reformulation Linearization Technique (RLT), which transforms the quadratic problem into a linear one, making it more tractable for linear programming solvers.

The QAP can be defined by two matrices:

Flow Matrix (f): An ( $n \times n$ ) matrix where ($\mathrm{f_{ij}}$) represents the flow from facility ( i ) to facility ( j).

Distance Matrix ($d$): An ( $n \times n$ ) matrix where ($d_{pq}$) denotes the distance from location ( $p$ ) to location ( $q$).

$c_{ip}$: Cost of installing department $i$ to department $j$,

$$x_{ip} = \begin{cases} 1, & \text{If department } i \text{ is located in location } p \\ 0, & \text{Others} \end{cases}$$

The objective is to minimize the total cost:

$$[\,Min \sum_{q=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} d_{pq} f_{ij} x_{ip}\, x_{jq} + \sum_{p=1}^{n}\sum_{i=1}^{n} C_{ip} x_{ip}]$$

where$x_{ij}$ is a binary variable that is 1 if facility ( $i$ ) is assigned to location( $j$ ) and 0 otherwise.

**Linearization**

Quadratic terms ($x_{ip}x_{jq}$) by introducing auxiliary variables ($y_{ijkl}$), which are intended to represent these products. The new linear objective function becomes:

$$[Min \sum_{q=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} d_{pq} f_{ij} y_{ipjq} + \sum_{p=1}^{n}\sum_{i=1}^{n} C_{ip} x_{ip}]$$

To ensure that ($y_{ipjq}$) accurately represents the product ($\mathrm{x_{ip}x_{kl}}$), the following constraints are added:

$$\sum_{i=1}^{n} x_{ip} = 1 \qquad \forall p$$

$$\sum_{i=1}^{p} x_{ip} = 1 \qquad \forall n$$

$$\sum_{q\neq p} y_{ipjq} = x_{ip} \qquad \forall i \neq j, p$$

$$\sum_{j\neq i} y_{ipjq} = x_{ip} \qquad \forall i, p \neq q$$

$$y_{ipjq} = y_{jqip} \qquad \forall i \neq j, p \neq q$$

Zhnag et al. (2010) proposed a modification in Adam and Jhonsn (2010) linearization and they proved that the modification improve computational time in benchmark of QAP. I calculated the time using tic toc in MATLAB 10 times and computed the average. The first layout took 0.5341 s and the second took 0.4292 s.

**Layout linearization 2:**

$$Min \sum_{q=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{i=1}^{n} d_{pq} f_{ij} y_{ipjq} + \sum_{p=1}^{n}\sum_{i=1}^{n} C_{ip} x_{ip} \qquad (3.1)$$

$$\sum_{i=1}^{n} x_{ip} = 1 \qquad \forall p \qquad (3.2)$$

$$\sum_{i=1}^{p} x_{ip} = 1 \qquad \forall n \qquad (3.3)$$

$$\sum_{q\neq p} y_{ipjq} = x_{ip} \qquad \forall i > j, p \qquad (3.4)$$

$$\sum_{j>i} y_{ipjq} \leq x_{ip} \qquad \forall i, p \neq q \qquad (3.5)$$

$$y_{ipjq} = y_{jqip} \qquad \forall i \neq j, p \neq q \qquad (3.6)$$

**Code explanation**

The MATLAB script implements a quadratic assignment problem (QAP) where $x(i,p)$ are binary decision variables representing whether facility ii is assigned to location pp, and $y(i,p,j,q)$ are auxiliary variables representing the product $x(i,p) \cdot x(j,q)$ The objective is to minimize the total cost, which includes flows between facilities and distances between locations, multiplied by the corresponding yy variables. The objective function is constructed to account for these costs, ensuring an optimal assignment that minimizes the overall expense.

The constraints ensure that each facility is assigned to exactly one location and each location is assigned exactly one facility. The linking constraints ensure that the y variables correctly represent the product of the xx variables, maintaining the integrity of the solution. Specifically, the constraints enforce that $y(i, p, j, q) = x(i, p) \cdot x(j, q)$ The script uses the Gurobi solver through YALMIP to solve the problem, with solver output set to be displayed for verification. The script executes the optimization process over ten trials to account for variability in execution times and calculates the average execution time for a reliable performance measure. The optimal assignment matrix xx is displayed along with the average execution time, providing a comprehensive solution to the QAP.

**Result**

1: Explored 1 nodes (6 simplex iterations) in 0.02 seconds (0.00 work units)

Thread count was 8 (of 8 available processors)

Solution count 1: 16

Optimal solution found (tolerance 1.00e-04)

Best objective 1.600000000000e+01, best bound 1.600000000000e+01, gap 0.0000%

Optimal assignment matrix X:

```
    0     0     1     0

    1     0     0     0

    0     1     0     0

    0     0     0     1
```

Average execution time over 10 trials: 0.5341 seconds

2: Explored 1 nodes (6 simplex iterations) in 0.02 seconds (0.00 work units)

Thread count was 8 (of 8 available processors)

Solution count 1: 16

Optimal solution found (tolerance 1.00e-04)

Best objective 1.600000000000e+01, best bound 1.600000000000e+01, gap 0.0000%

Optimal assignment matrix X:

```
    0     0     1     0

    1     0     0     0

    0     0     0     1

    0     1     0     0
```

Average execution time over 10 trials: 0.4292 seconds

3- Explored 0 nodes (0 simplex iterations) in 0.02 seconds (0.00 work units)

Thread count was 1 (of 8 available processors)

Solution count 1: 16

Optimal solution found (tolerance 1.00e-04)

Best objective 1.600000000000e+01, best bound 1.600000000000e+01, gap 0.0000%

Optimal assignment matrix X:

```
1  0  0  0
0  1  0  0
0  0  1  0
0  0  0  1
```

Average execution time over 10 trials: 0.6125 seconds