The Quadratic Assignment Problem

1 The Quadratic Assignment Problem (QAP)

The quadratic assignment problem (QAP) is a standard problem in facility location. We have \( n \) facilities, each to be placed at one of \( n \) locations. There is a cost \( c_{ik} \) for placing facility \( i \) at location \( k \). Goods need to be moved between the locations, and we let \( a_{ij} \) denote the flow from facility \( i \) to facility \( j \). There is a cost to move goods between locations; we let \( b_{kl} \) denote the cost to move one unit from location \( k \) to location \( l \).

The 1-to-1 map from facilities to locations can be regarded as a permutation of the integers \( 1, \ldots, n \). We define variables \( x_{ik} \) to represent the mapping, so

\[
x_{ik} = \begin{cases} 
1 & \text{if facility } i \text{ is placed in location } k \\
0 & \text{otherwise}
\end{cases} \quad \text{for } i = 1, \ldots, n, \ k = 1, \ldots, n.
\]

The variables \( x_{ik} \) constitute an \( n \times n \) matrix \( X \). In feasible solutions, this matrix \( X \) is a permutation matrix, with exactly one “1” in each row and in each column. The objective function can then be written as

\[
\min_{X} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij} b_{kl} x_{ik} x_{jl} + \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} x_{ik} + \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} x_{ik}
\]

We let \( A, B, \) and \( C \) denote the matrices of the cost coefficients. In matrix terms, the objective function can also be expressed as follows:

\[
\min_{X} \text{trace}((AXB + C)X^T).
\]

The QAP is \( \mathcal{NP} \)-hard and very difficult to solve to global optimality. Problems with \( n = 30 \) are extremely challenging.

2 Solution approach

Currently, the most effective method for solving quadratic assignment problems is to use branch-and-bound, with quadratic programming subproblems [2]. The entries in the \( n \times n \) matrix \( X \) can be arranged as a vector \( x \) with \( n^2 \) entries; we write \( x = \text{vec}(X) \). The objective function can then be written as

\[
\min_{x} \frac{1}{2} x^T Q x + c^T x
\]

for appropriately defined matrix \( Q \) and vector \( c \). Let \( e \) denote the vector of ones, dimensioned appropriately. The QAP can be written equivalently as

\[
\min_{x,X} \frac{1}{2} x^T Q x + c^T x \quad \text{subject to } \quad X e = e, \ X^T e = e, \ x = \text{vec}(X), \ X_{ik} \text{ binary } \forall i, k
\]
The constraints \( Xe = e \) and \( X^Te = e \) capture the requirements that each row and each column of \( X \) contain exactly one “1”, respectively.

**Proposition 1** \([1]\) The matrix \( Q \) is positive semidefinite on the null space of the set of solutions to \( Xe = e, X^Te = e \).

Thus, if we relax the binary requirement, we have a convex quadratic programming problem. This observation is the basis for a branch-and-bound algorithm.

An SDP relaxation can also be constructed by replacing the outer product \( xx^T \) by a \( n^2 \times n^2 \) positive semidefinite matrix \( Z \). The objective function is linear in this matrix, since \( x^TQx = \text{trace}(QZ) \).

### 3 Formulating the TSP as a QAP

The traveling salesman problem can be formulated as QAP:

let \( x_{ik} = \begin{cases} 1 & \text{if city } i \text{ appears in position } k \text{ on the tour} \\ 0 & \text{otherwise} \end{cases} \) for \( i = 1, \ldots, n, k = 1, \ldots, n \).

The entries in the matrix \( A \) are the distances between the cities. Define the parameter matrix \( B \) by

\[
b_{ik} = \begin{cases} 1 & \text{if } k - i = 1 \text{ or } i - k = n - 1 \\ 0 & \text{otherwise} \end{cases} \text{ for } i = 1, \ldots, n, k = 1, \ldots, n.
\]

Then the value of the tour is

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij}b_{kl}x_{ik}x_{jl} = \text{trace}(AXBX^T).
\]

### References
