620-362
Applied Operations Research

The Quadratic Assignment Problem (QAP)

Department of Mathematics and Statistics
The University of Melbourne

Some contents of this presentation are adapted from year 2005 course notes for 620-362 Applied Operations Research, Department of Mathematics and Statistics, The University of Melbourne (compiled by Prof Natashia Boland and Dr Renata Sotirov)

The Quadratic Assignment Problem (QAP)

…introduced in 1957 by Koopmans and Beckman as a model for location problems (assigning a set of economic activities to a set of locations)

…model for real-life problems:

- building layout (architectural design)
- board wiring (locating modules on the board to minimise total wire length)
- control panel and keyboard design (optimised for a language)
- machine scheduling (minimizing average job completion time)
- information retrieval (optimal ordering of data on tapes)
- analysis of chemical reactions for organic compounds
- ranking of a team in a relay race
- balancing turbine runners

QAP

QAP can be stated in the following way:

For given

$A = (a_{ij})$, $B = (b_{ij})$, and $C = (c_{ij})$

real n x n matrices, find a permutation π of the set $\{1,\ldots,n\}$ which minimises

$$\min_{\pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi(i)\pi(j)} + \sum_{i=1}^{n} c_{i,\pi(i)}$$

QAP

$$\min_{\pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi(i)\pi(j)} + \sum_{i=1}^{n} c_{i,\pi(i)} \qquad (1)$$

- permutation π_0 which minimise (1) is called an **optimal solution**.
- the quadratic part? $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi(i)\pi(j)}$
- the linear term? $\sum_{i=1}^{n} c_{i,\pi(i)}$
- size of QAP:= n
- QAP is a *combinatorial optimization* problem.
  – how many possibilities are there?

Example 1

Campus planning problem

- the University owns a piece of land on which new buildings are to be placed
- on the land *n* sites have been identified as *sites for the buildings*
- each building has a *special function* (library,…)

Let

a_{ij} : the walking distance between sites *i* and *j*

$A = (a_{ij})$: the *distance matrix*

b_{kl} : the number of people per week who circulate between buildings *k* and *l*

$B = (b_{kl})$: the flow matrix

Define

$\pi(i) = k$: if building *k* is built on site *i*

Example 1

- Minimising the walking distance at campus?

Now, if building *k* is built on site *i* and building *l* on site *j*, then the weekly walking distance of people who travel between building *k* and building *l* is given by

$$a_{ij} b_{\pi(i)\pi(j)}$$

Problem of assigning buildings to sites so that the walking distance between sites is minimised:

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

Example 1

- Minimising the total construction cost?

  c_{ik} : cost of constructing building *k* on site *i*

  The cost construction minimisation problem is:

$$\min_{\pi} \sum_{i=1}^{n} c_{i\pi(i)}$$

Example 1

The minimisation problem whose solution satisfies both requirements is:

$$\min_{\pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi(i)\pi(j)} + \sum_{i=1}^{n} c_{i,\pi(i)}$$

Remark: π is a permutation of the set {1,…,n}

$\pi(i) \neq \pi(j)$, forall i,j with $i \neq j$

Example 2 (Numerical)

$$\text{Distances between cites}: A = \begin{pmatrix} 0 & 50 & 70 & 100 \\ 50 & 0 & 45 & 60 \\ 70 & 45 & 0 & 95 \\ 100 & 60 & 95 & 0 \end{pmatrix}$$

$$\text{Expected \# people commuting}: B = \begin{pmatrix} 0 & 1234 & 1050 & 900 \\ 1234 & 0 & 500 & 300 \\ 1050 & 500 & 0 & 400 \\ 900 & 300 & 400 & 0 \end{pmatrix}$$

If building i assigned to site i, i.e. $\pi(i) = i$, total weekly walking distance

= 2 x (50 x 1234 + 70 x 1050 + 100 x 900 + 45 x 500+ 60 x 300 + 95 x 400)

= 607,400

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

Equivalent Formulation of QAP

Definition. Let $X = (x_{ij})$ be an n x n matrix. X is called a *permutation matrix* if entries x_{ij} fulfill the following conditions:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n$$

$$x_{ij} \in \{0,1\}, \quad 1 \le i \le n, 1 \le j \le n$$

Definition. Π = {set of all n x n permutation matrices}

Equivalent Formulation of QAP

- There is one-to-one correspondence between

the set of all permutations of the set {1,…,n}

and

the set of n x n permutation matrices

- For the entries of $X = (x_{ij}) \in \Pi$, we specify

$$x_{ik} = \begin{cases} 1, & \text{if and only if } \pi(i) = k \\ 0, & otherwise \end{cases}$$

Equivalent Formulation of QAP

- If $\pi(i) = k$ and $\pi(j) = l$ then

$$a_{ij}b_{\pi(i)\pi(j)} = \sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij}b_{kl}x_{ik}x_{jl}$$

- An equivalent formulation of QAP:

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

  This is the Koopmans Beckmann formulation of the QAP.

- The formulation above is a formulation with a **quadratic objective function**.

Equivalent Formulation of QAP

$$\min_{X \in \Pi} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij} b_{kl} x_{ik} x_{jl} + \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} x_{ik} \quad \equiv$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij} b_{kl} x_{ik} x_{jl} + \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ik} x_{ik}$$

$s.t.$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n$$

$$x_{ij} \in \{0,1\}, \quad 1 \le i \le n, 1 \le j \le n$$

Trace formulation of QAP

Definition. The trace of the n x n matrix A is

$$tr(A) = \sum_{i=1}^{n} a_{ii}$$

– for m x k matrices A and B it follows:

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix}, \quad B^T = \begin{bmatrix} b_{11} & b_{21} & b_{31} & b_{41} \\ b_{12} & b_{22} & b_{32} & b_{42} \\ b_{13} & b_{23} & b_{33} & b_{43} \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix}$$

$$tr\left(B^T A\right) = \sum_{i=1}^{m} \sum_{j=1}^{k} a_{ij} b_{ij}$$

– note that the ik-entry of AXB^T is

$$\left(AXB^T\right)_{ik} = \sum_{j,l} a_{ij} x_{jl} b_{kl}$$

– and i^{th} diagonal entry of CX^T is

$$\left(CX^T\right)_{ii} = \sum_{k} c_{ik} x_{ik}$$

Trace formulation of QAP

It follows that

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ij} b_{kl} x_{ik} x_{jl} + \sum_{i=1}^{n}\sum_{k=1}^{n} c_{ik} x_{ik}$$

$$= \min_{X \in \Pi} \sum_{i=1}^{n}\sum_{k=1}^{n} \left(AXB^{T}\right)_{ik} x_{ik} + \sum_{i=1}^{n} \left(CX^{T}\right)_{ii}$$

$$= \min_{X \in \Pi} tr\left(AXB^{T} + C\right)X^{T}$$

This is the trace formulation of QAP introduced by Edwards in 1977.

Why use trace formulation?

Allows flexible algebraic manipulation -
lower bound calculation.

Why is lower bound needed?

Example 3

Airport gate assignment:

- airport has gates
- flights arriving and departing from gates
- known expected number of passengers transferring between flights
- known distances between gates

Objective:

Which flight is to be assigned to which gate, so as to minimise the total distance travelled by transiting passengers

Let

F = set of flights

G = set of gates

t_{ij} = expected number of passengers transferring from arriving flight *i* to departing flight *j*

d_{kl} = distance between gates *k* and *l*

Example 3

$$x_{ij} = \begin{cases} 1, & \text{flight } i \text{ is assigned to gate } k \\ 0, & \textit{otherwise} \end{cases}$$

$$\min \sum_{i \in F} \sum_{j \in F} \sum_{k \in G} \sum_{l \in G} t_{ij} d_{kl} x_{ik} x_{jl}$$

$$s.t.$$

$$\sum_{k \in G} x_{ik} = 1, \quad \forall i \in F$$

$$\sum_{k \in F} x_{ik} = 1, \quad \forall k \in G$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in F, k \in G$$

Computational Complexity of QAP

- QAP is one of the most difficult combinatorial optimization problems.
- Finding an ε-approximate solution for QAP is also a hard problem.
- Practice show that QAP is extremely hard to solve to optimality.
- The computational effort to solve QAP is very likely to grow exponentially with the problem size.
- Problems of size $n \geq 20$ are currently considered as huge problems.

Linearising the QAP Model

$$y_{ijkl} = \begin{cases} 1, & i \text{ assigned to } k \text{ and } j \text{ to } l \\ 0, & \textit{otherwise} \end{cases}$$

$$\min \sum_i \sum_j \sum_k \sum_l t_{ij} d_{kl} y_{ijkl}$$

$$s.t.$$

$$\sum_i \sum_j \sum_k \sum_l y_{ijkl} = n^2$$

$$x_{ik} + x_{jl} \geq 2 y_{ijkl}, \quad \forall i, j, k, l$$

$$y_{ijkl} \geq x_{ik} + x_{jl} - 1$$

...

- Linearised model has
  $n^4 + n^2$ variables and $O(n^4)$ constraints

- Solving linearised QAP is impossible (due to hardware restrictions) for larger *n*

- Other solution methods:
  - (meta)heuristics
  - dynamic programming
  - branch-and-bound

- Breakthroughs:
  - E.g. in 2001, a number of previously unsolved large QAPs ($20 \le n \le 36$) were solved to optimality on supercomputer. The problem of $n = 30$ would have took an equivalent of 7 years of computational time on a single HP9000 C3000 workstation!
  - For more updates, see http://www.seas.upenn.edu/qaplib/