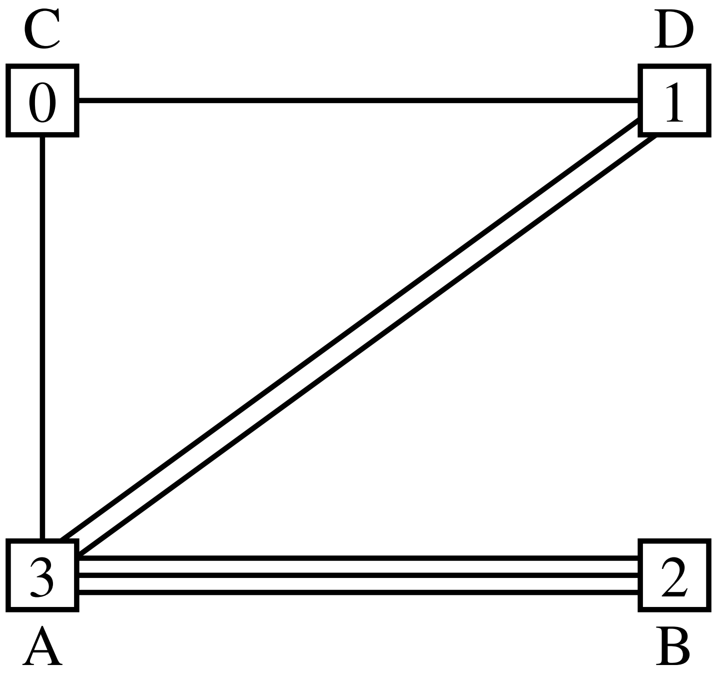
Metaheuristics for Optimization

Series 2: The Quadratic Assignment Problem

Quadratic Assignment Problem

- Definition: Combinatorial optimization problem
- Example: Find the best way to assign a **set of n facilities** (factories) to a **set of n locations** (cities) accordingly to **distances** and **flows** (amount of things that needs to be moved)
- Minimize the sum of products « distance-flow »



- Research space: Permutations —> Size $n! = n \times (n-1) \times \ldots \times 1$

$$\boxed{12! = 479'001'600}$$

in this TP

Quadratic Assignment Problem

- Example: Find the best location (A, B, C, D) for each facility (0, 1, 2, 3) in order to minimize

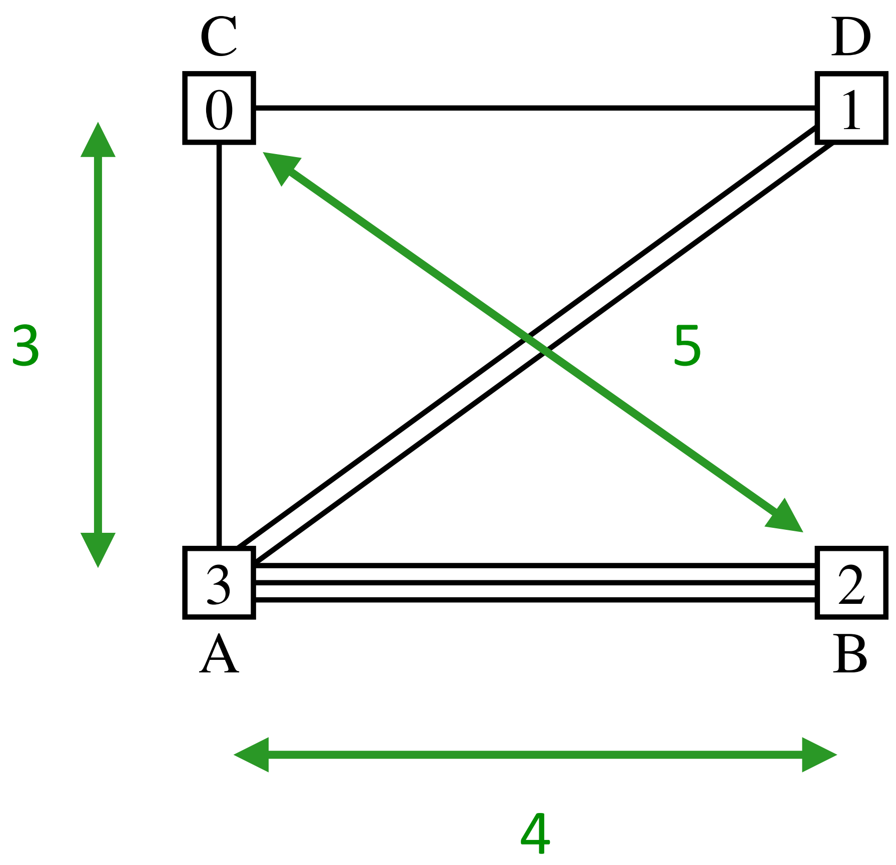
$$I(\psi) = \sum_{i,j=0}^{n-1} w_{ij} \times d_{\psi_i,\psi_j}$$

distances

$$d_{AB} = d_{CD} = 4$$
$$d_{AC} = d_{BD} = 3$$
$$d_{AD} = d_{BC} = 5$$

flows

$$w_{13} = 2$$
$$w_{01} = w_{03} = 1$$
$$w_{23} = 3$$



Fitness $I(\psi) = w_{01} \times d_{\psi_0\psi_1} + w_{03} \times d_{\psi_0\psi_3} + w_{13} \times d_{\psi_1\psi_3} + w_{23} \times d_{\psi_2\psi_3}$

Here $\psi = (C, D, B, A)$

Hence

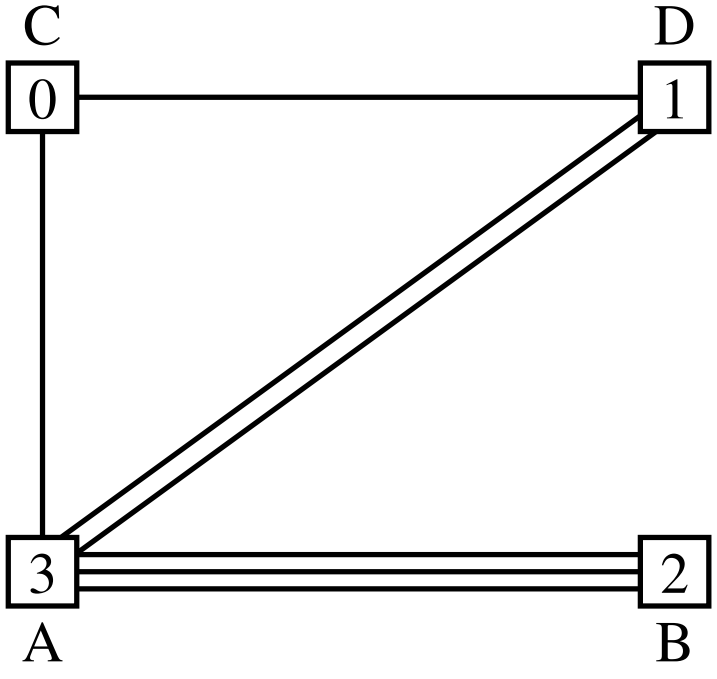
$$\boxed{I(\psi) = d_{CD} + d_{AC} + 2d_{AD} + 3d_{AB} = 29}$$

Quadratic Assignment Problem

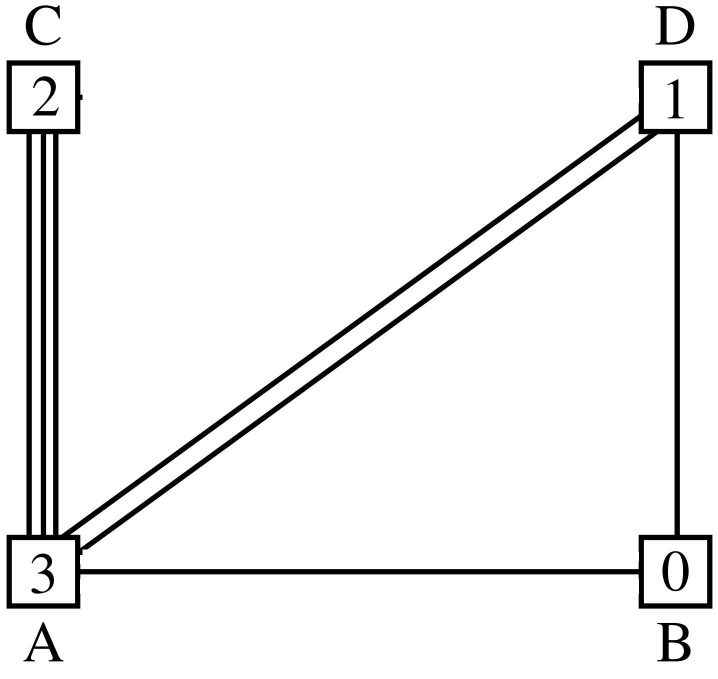
- Neighborhood: Permutations of two elements (2-swap)

$\longrightarrow$ $n(n-1)/2$ neighbors

$\psi = (C, D, B, A)$



$\psi' = (B, D, C, A)$

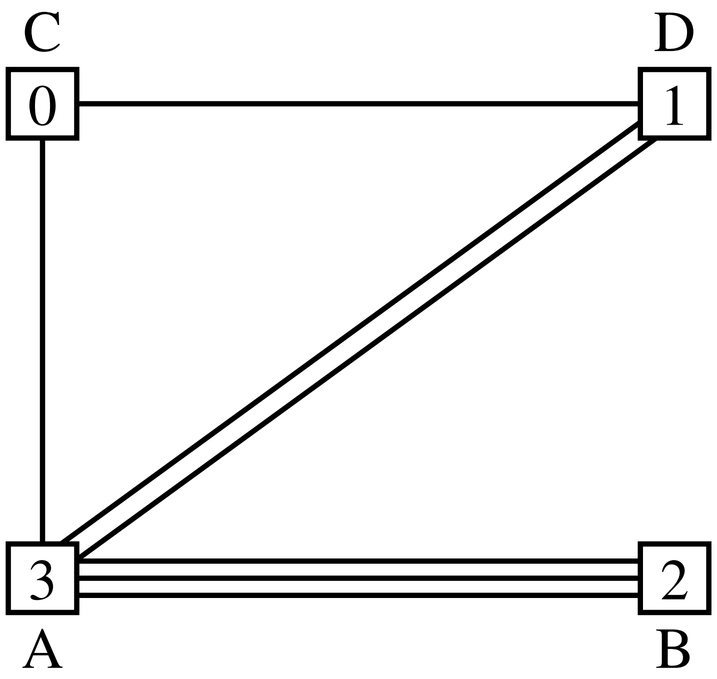


Quadratic Assignment Problem

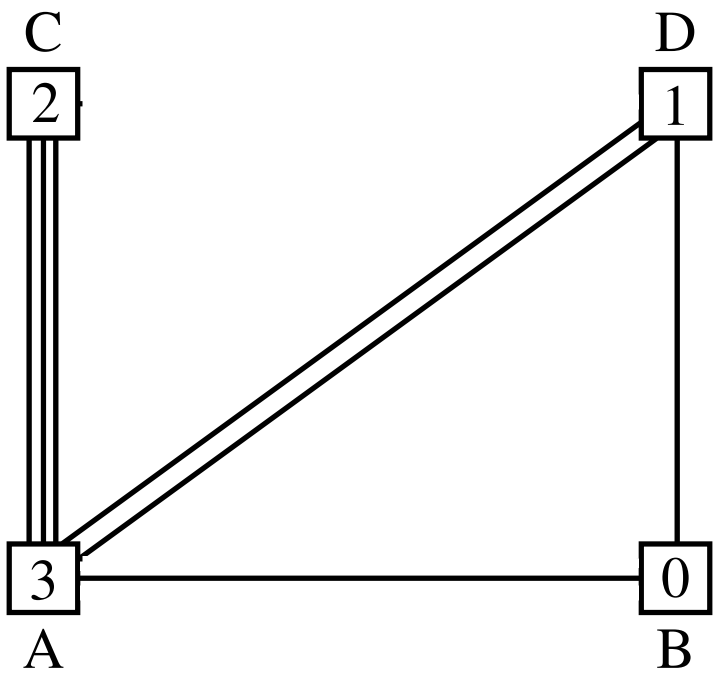
- Neighborhood: Permutations of two elements (2-swap)

$\longrightarrow$ $n(n-1)/2$ neighbors

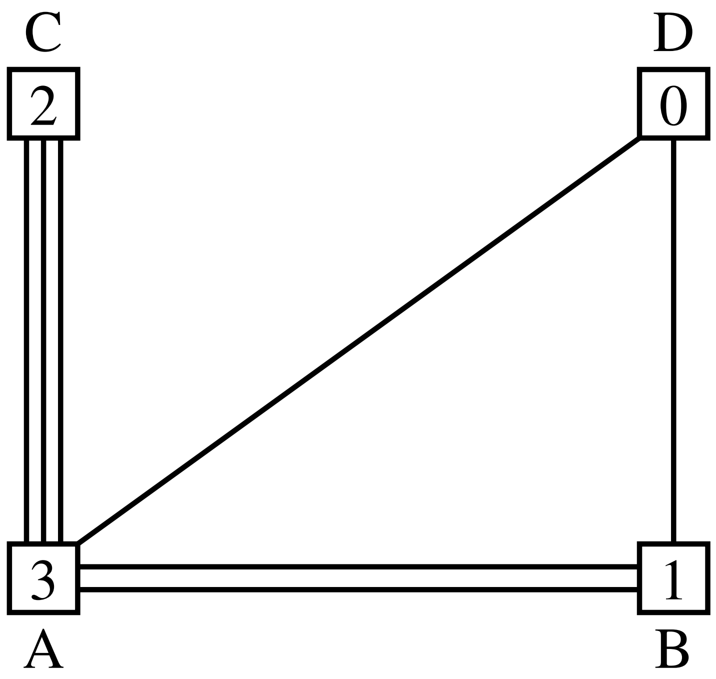
$\psi = (C, D, B, A)$



$\psi' = (B, D, C, A)$



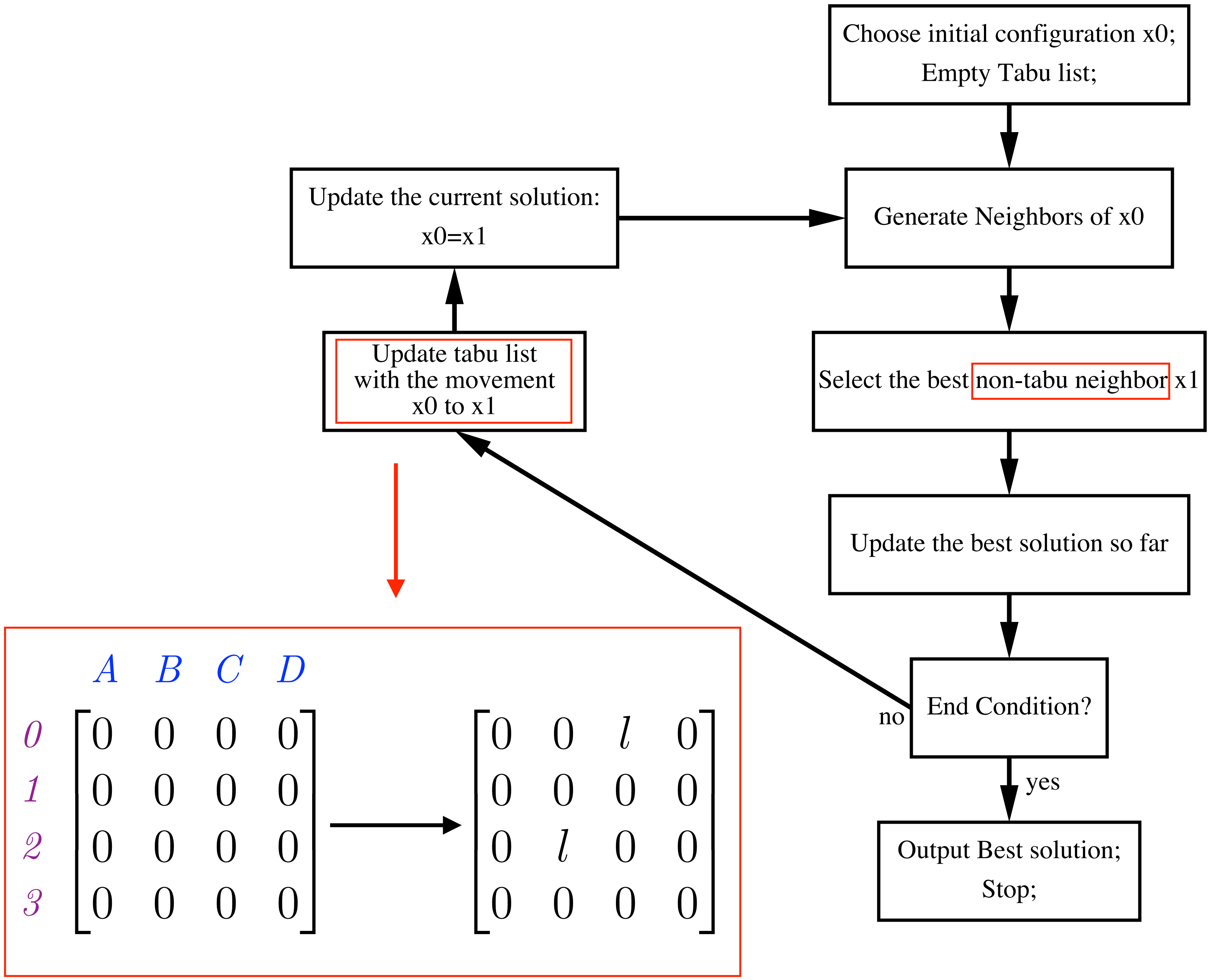
$\psi'' = (D, B, C, A)$



Optimal combination

Tabu search

- Purpose: Avoid already explored solutions by forbidding moves/permutations



Choose initial configuration x0;
Empty Tabu list;
Generate Neighbors of x0
Select the best non-tabu neighbor x1
Update the best solution so far
End Condition?
yes
Output Best solution;
Stop;
no
Update tabu list with the movement x0 to x1
Update the current solution:
x0=x1

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \end{array} \longrightarrow \begin{bmatrix} 0 & 0 & l & 0 \\ 0 & 0 & 0 & 0 \\ 0 & l & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\psi = (C, D, B, A)$$

$$\psi', \psi'', \psi''', etc$$

$$\psi' = (B, D, C, A)$$

Tabu search

- <u>Two types of memory</u>

→ Short term: Avoid solutions that were visited during the last **l** iterations (forbidden permutations)

→ Long term: Impose a certain move/permutation if it has not been chosen during the last u=n2 iterations

Tabu search

- Two types of memory

→ Short term: Avoid solutions that were visited during the last **l** iterations (forbidden permutations)

→ Long term: Impose a certain move/permutation if it has not been chosen during the last u=n2 iterations

Work to do (main steps)

- Use the tabu search to solve the QAP
- Quantify the impact of both memories (and the aspiration process) on the convergence of the tabu search
- To do so, run 10 (or more) simulations and return the (1) best, (2) mean, and (3) variance/std of the fitness