ISEN 621: Homework 5
Tabu Search for the Qaudratic Assignment Problem

Feyi Olalotiti-Lawal

December 3, 2013

Problem

Given a set of $n = |V|$ facilities (vertices) and flows a_{ij} from facility (vertex) i to j such that $\{i, j\} \in E$ for a system represented by a graph $G = (V, E)$, there are n locations for the facilites and b_{uv} distances between locations u and v for $u, v \in E$.
The optimization problem is to find such permutation (facility arrangement), π^* that minimizes the objective function given as:

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

This is referred to as the Qaudratic Assignment Problem (QAP)

The Algorithm section describes the solution apporach. This algorithm was applied to 2 cases: First with a 14-vertex graph and second, with a 48 vertex problem. The flow matrix A and the distance matrix B are shown in Equations 2 and 3 respectively.

$$A = \begin{bmatrix} 0 & 2 & 13 & 12 & 14 & 8 & 7 & 9 & 15 & 20 & 15 & 12 & 2 & 14 \\ 2 & 0 & 7 & 1 & 11 & 6 & 11 & 7 & 13 & 7 & 18 & 4 & 10 & 8 \\ 13 & 7 & 0 & 19 & 17 & 13 & 19 & 5 & 16 & 11 & 19 & 15 & 12 & 11 \\ 12 & 1 & 19 & 0 & 1 & 17 & 1 & 8 & 18 & 4 & 5 & 8 & 10 & 11 \\ 14 & 11 & 17 & 1 & 0 & 14 & 6 & 1 & 11 & 18 & 6 & 5 & 1 & 11 \\ 8 & 6 & 13 & 17 & 14 & 0 & 5 & 3 & 1 & 12 & 7 & 7 & 12 & 3 \\ 7 & 11 & 19 & 1 & 6 & 5 & 0 & 11 & 19 & 16 & 15 & 16 & 15 & 4 \\ 9 & 7 & 5 & 8 & 1 & 3 & 11 & 0 & 12 & 5 & 8 & 16 & 6 & 13 \\ 15 & 13 & 16 & 18 & 11 & 1 & 19 & 12 & 0 & 18 & 2 & 17 & 17 & 2 \\ 20 & 7 & 11 & 4 & 18 & 12 & 16 & 5 & 18 & 0 & 18 & 14 & 13 & 1 \\ 15 & 18 & 19 & 5 & 6 & 7 & 15 & 8 & 2 & 18 & 0 & 4 & 1 & 14 \\ 12 & 4 & 15 & 8 & 5 & 7 & 16 & 16 & 17 & 14 & 4 & 0 & 13 & 10 \\ 2 & 10 & 12 & 10 & 1 & 12 & 15 & 6 & 17 & 13 & 1 & 13 & 0 & 16 \\ 14 & 8 & 11 & 11 & 11 & 3 & 4 & 13 & 2 & 1 & 14 & 10 & 16 & 0 \end{bmatrix} \tag{2}$$

Algorithm

Algorithm 1 A Tabu Search Algorithm for QAP

Input: Graph flow matrix A, Vertex Locations and Distance matrx B, Tabu length $|T|$, Aspiration length $|As|$, $maxitr$

Output: Optimal vartex permutation

Initilaize:
Create all false logical matrices $[Tabu]_{i,j}$ and $[Asp]_{i,j}\ \ \forall i,j \in \{1,\ldots,n\}$
Initialize tabu iteration counter matrix $[Z]_{ij} = 0$
Calculate initial objective function $f(\pi^0)$
for itr=1 **to** $maxitr$ **do**
  for i=1 **to** n **do**
    for j=1 **to** n **do**
      Perfrom local exchange of vertices and calculate improvement $\Delta f_{ij} = \Delta(\pi, i, j)$
    end for
  end for
  Select best permuation such that $\big(\Delta(\pi,u,v) \le \Delta(\pi,i,j)\ \ \forall i,j \in \{1,\ldots,n\}\big)\&\big(Tabu(u,v) = false$ or $Asp(u,v) = true\big)$
  Update Tabu iteration matrix: $Z(u,v) \leftarrow itr$; $Z(v,u) \leftarrow itr$
  Update Tabu logical matrix: $Tabu(i,j) = \big(Z(i,j) > 0\big)\&\big(Z(i,j) + |T| \ge itr\big)\&\big(Z(j,i) + |T| \ge itr\big)\ \ \forall i,j \in \{1,\ldots,n\}$
  Update Aspiration logical matrix: $Asp(i,j) = \big(\Delta(\pi,i,j) \le (\pi,i,j)_{Best}\big)$ or $\big(max([Z(i,j) + |As|, Z(j,i) + |As|]) \le itr\big)\ \ \forall i,j \in \{1,\ldots,n\}$
  Store optimal improvement. Update B matrix and vertex locations
end for
Return optimal permutation

$$B = \begin{bmatrix}
0.00 & 1.66 & 5.08 & 6.52 & 8.83 & 5.53 & 4.10 & 0.75 & 1.29 & 3.15 & 1.28 & 5.08 & 3.12 & 3.94 \\
1.66 & 0.00 & 4.09 & 6.02 & 9.20 & 5.76 & 4.76 & 1.99 & 2.94 & 4.40 & 2.94 & 5.18 & 3.98 & 3.62 \\
5.08 & 4.09 & 0.00 & 2.45 & 6.96 & 4.00 & 4.50 & 4.73 & 6.15 & 8.22 & 6.01 & 3.37 & 4.64 & 2.01 \\
6.52 & 6.02 & 2.45 & 0.00 & 4.80 & 2.71 & 4.12 & 5.96 & 7.29 & 9.60 & 7.10 & 2.38 & 4.80 & 2.59 \\
8.83 & 9.20 & 6.96 & 4.80 & 0.00 & 3.44 & 4.77 & 8.09 & 8.93 & 11.21 & 8.70 & 4.06 & 5.82 & 5.80 \\
5.53 & 5.76 & 4.00 & 2.71 & 3.44 & 0.00 & 1.81 & 4.81 & 5.85 & 8.22 & 5.63 & 0.66 & 2.81 & 2.43 \\
4.10 & 4.76 & 4.50 & 4.12 & 4.77 & 1.81 & 0.00 & 3.35 & 4.19 & 6.51 & 3.96 & 1.77 & 1.07 & 2.50 \\
0.75 & 1.99 & 4.73 & 5.96 & 8.09 & 4.81 & 3.35 & 0.00 & 1.41 & 3.64 & 1.28 & 4.38 & 2.36 & 3.37 \\
1.29 & 2.94 & 6.15 & 7.29 & 8.93 & 5.85 & 4.19 & 1.41 & 0.00 & 2.37 & 0.23 & 5.52 & 3.12 & 4.73 \\
3.15 & 4.40 & 8.22 & 9.60 & 11.21 & 8.22 & 6.51 & 3.64 & 2.37 & 0.00 & 2.59 & 7.89 & 5.45 & 7.02 \\
1.28 & 2.94 & 6.01 & 7.10 & 8.70 & 5.63 & 3.96 & 1.28 & 0.23 & 2.59 & 0.00 & 5.30 & 2.89 & 4.55 \\
5.08 & 5.18 & 3.37 & 2.38 & 4.06 & 0.66 & 1.77 & 4.38 & 5.52 & 7.89 & 5.30 & 0.00 & 2.61 & 1.77 \\
3.12 & 3.98 & 4.64 & 4.80 & 5.82 & 2.81 & 1.07 & 2.36 & 3.12 & 5.45 & 2.89 & 2.61 & 0.00 & 2.67 \\
3.94 & 3.62 & 2.01 & 2.59 & 5.80 & 2.43 & 2.50 & 3.37 & 4.73 & 7.02 & 4.55 & 1.77 & 2.67 & 0.00
\end{bmatrix} \tag{3}$$

Results

Figures 1, 2 and 3 respectively show the initial configuration (vertex arrangement), the final configuration and the objective function with iterations for Case 1 with 14 vertices. Here, 70 iterations were carried out, updating the tabu abnd aspiration lists continually maintained.

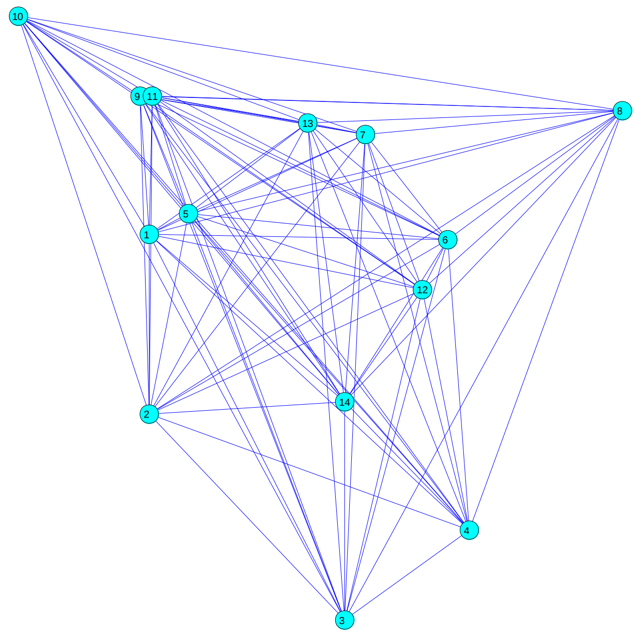

Figure 1: Initial configuration for Case 1

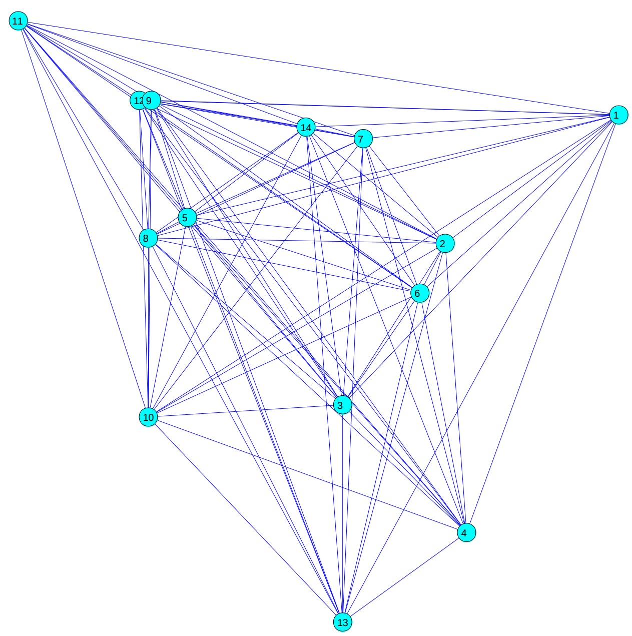

Figure 2: Optimal configuration for Case 1

Similar approach was followed for the more complicated Case 2 with 48 vertices. Here likewise, the objective function reduces.

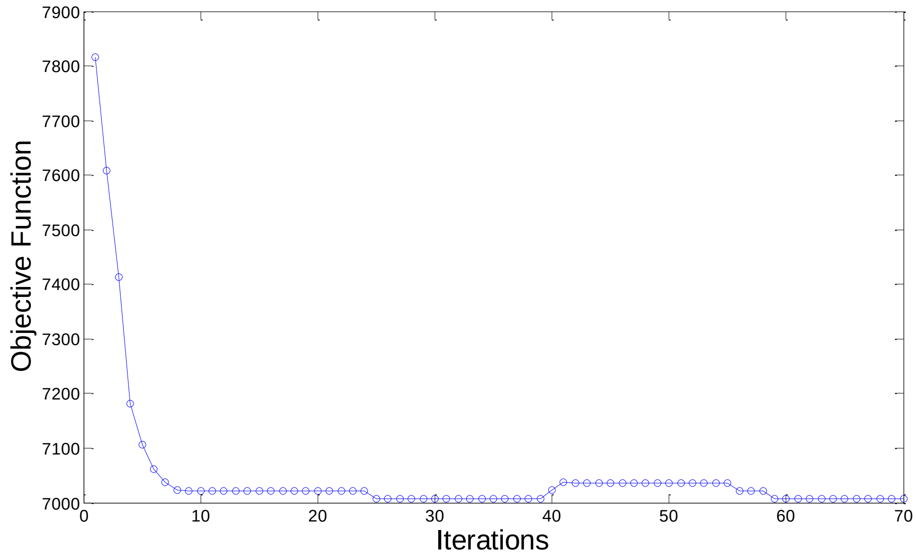

Figure 3: Objective Funtion for Case 1

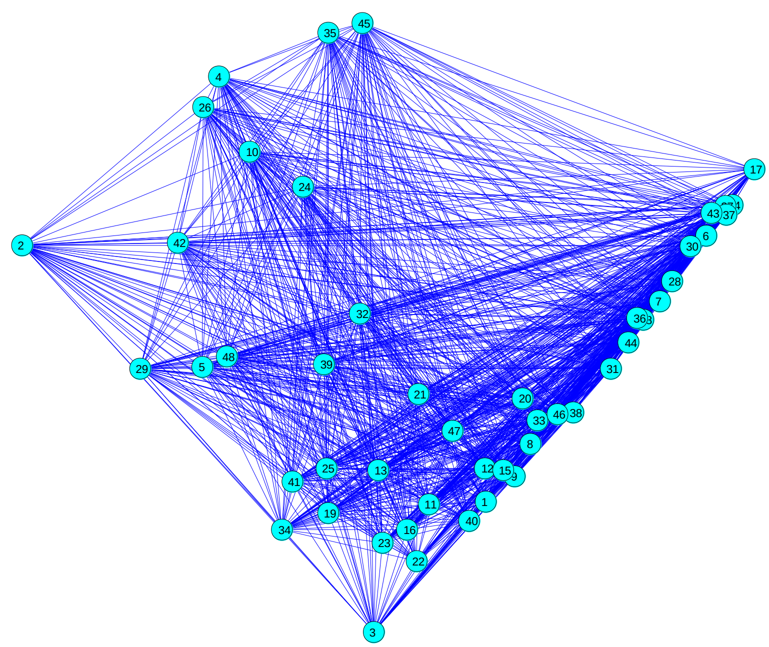

Figure 4: Initial configuration for Case 2

Codes

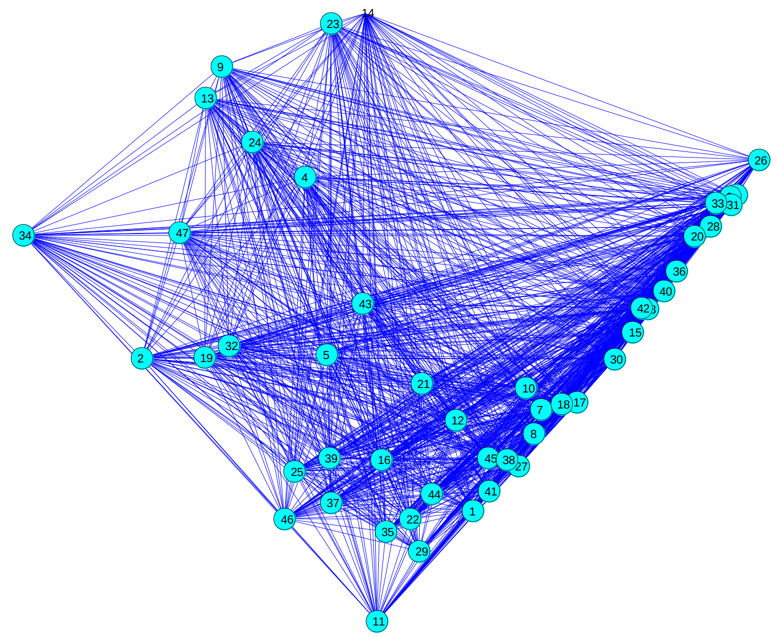

Figure 5: Optimal configuration for Case 2

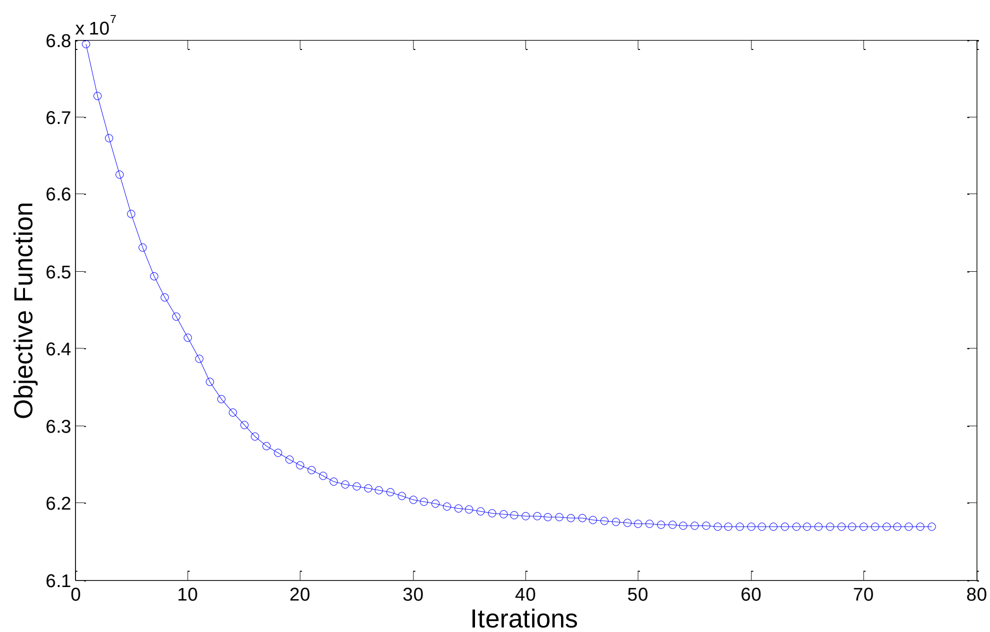

Figure 6: Objective Funtion for Case 2

```
%% AUTHOR: FEYI OLALOTITI-LAWAL
close all; clc; clear all;

% load case14
load case48
n = size(P,1);          % Graph size
permOrder = (1:n)';
T = n;              % Tabu length
As = 5*n^2;        % Aspiration length
maxit = 200;    % Maximum iterations

% Calculate distance matrix and position matrix Z
B = calcDist(P);
Z = zeros(n);

% Initial objective function
OF = sum(sum(A.*B));
DpVec = [];
OF_vec = [];
Dp = zeros(size(A));
for i = 1:n-1
    for j = i+1:n
        P2 = P; P2(i,:) = P(j,:); P2(j,:) = P(i,:);
        B2 = calcDist(P2);
        Dp(i,j) = sum(sum(A.*B2))-sum(sum(A.*B));
    end
end
Dp = Dp+Dp';
[val ind] = sort(Dp(:)); jOpt = rem(ind(1),n); if jOpt==0; jOpt = n; end
iOpt = 1 + (ind(1)-jOpt)/n;
Z(iOpt,jOpt) = Z(iOpt,jOpt) + 1; Z(jOpt,iOpt) = Z(jOpt,iOpt) + 1;
DpOpt = val(1); DpVec = [DpVec; DpOpt];
OFvec = OF + DpOpt; DpBest = DpOpt;
itr = 0;
tabu = Z~=0 & Z+T >= itr & Z'+T >= itr;
aspiratn = Dp <= DpBest | max(max([Z+A Z'+A]))<=itr;
PUp = P; PUp(iOpt,:) = P(jOpt,:); PUp(jOpt,:) = P(iOpt,:);
temp = permOrder; temp(iOpt) = permOrder(jOpt); temp(jOpt) = permOrder(iOpt);
permOrder = temp; clear temp
BUp = calcDist(PUp);
P1 = PUp;
```

```
%Visualize Initial condition
figure(1); hold on;
for ii = 1:n-1; for jj = ii+1:n; coord = [P(ii,:);P(jj,:)];plot(coord(:,1),coord(:,2));end; end
for ii = 1:n
    plot(P(ii,1),P(ii,2),'o','MarkerSize',18,'MarkerFaceColor','c','MarkerEdgeColor','k')
    text(P(ii,1)-40,P(ii,2), num2str(permOrder(ii)));
end
axis tight;axis off

% Begin Search
for itr = 2:maxit
    Dp = zeros(size(A));
    for i = 1:n-1
        for j = i+1:n
            P2 = P1; P2(i,:) = P1(j,:); P2(j,:) = P1(i,:);
            B2 = calcDist(P2);
            Dp(i,j) = sum(sum(A.*B2))-sum(sum(A.*BUp));
        end
    end
    Dp = Dp+Dp';
    Dp_fsble = 10*ones(size(Dp));
    Dp_fsble(~tabu) = Dp(~tabu);
    Dp_fsble(aspiratn) = Dp(aspiratn);
    [val ind] = sort(Dp_fsble(:));

    rejectMove = 1; count = 0;

    while rejectMove
        count = count+1;
        jtemp = rem(ind(count),n); if jtemp==0; jtemp = n; end
        itemp = 1 + (ind(count)-jtemp)/n;
        rejectMove = tabu(itemp,jtemp)==1 && aspiratn(itemp,jtemp)==0;
    end

    iOpt = itemp; jOpt = jtemp;
    Z(iOpt,jOpt) = itr;
    Z(jOpt,iOpt) = itr;
    tabu = Z~=0 & Z+T >= itr & Z'+T >= itr;
    aspiratn = Dp <= DpBest | max(max([Z+A Z'+A]))<=itr;

    DpOpt = val(count);
    DpVec = [DpVec; DpOpt];
    check = ceil(maxit/10); if itr>check+1 && length(find(DpVec(itr-check:itr)~=0)) <=3; break; end
    OFvec = [OFvec; OFvec(itr-1) + DpOpt];
    if DpOpt <= DpBest; DpBest = DpOpt; end
```

```
PUp = P1; PUp(iOpt,:) = P1(jOpt,:); PUp(jOpt,:) = P1(iOpt,:);

    temp = permOrder; temp(iOpt) = permOrder(jOpt); temp(jOpt) = permOrder(iOpt);
    permOrder = temp; clear temp

    BUp = calcDist(PUp);
    P1 = PUp;

end

%Visualize Final condition
figure(2); hold on;
for ii = 1:n-1
    for jj = ii+1:n
        coord = [P(ii,:);P(jj,:)];
        plot(coord(:,1),coord(:,2));
    end
end
for ii = 1:n
    plot(P(ii,1),P(ii,2),'o','MarkerSize',18,'MarkerFaceColor','c','MarkerEdgeColor','k')
    text(P(ii,1)-40,P(ii,2), num2str(permOrder(ii)));
end
axis tight;axis off

%Objective Function
figure(3); plot(OFvec,'o-'); xlabel('Iterations'); ylabel('Objective Function');
```