Allocation of flights to land at the terminals of an airport

Pradeep Kumar and Shambhu Sharma
Department of Mathematics, Faculty of Science, Dayalbagh Educational Institute (Deemed University), Dayalbagh, Agra-282005 (Uttar Pradesh), India.

Email: pradeepkumar_dei@yahoo.com

Abstract: In this paper, an algorithm for solving flight landing problem has been developed to reduce the chaos at the airport. This algorithm starts with an initial basic feasible solution and then the algorithm works on the principle of steepest descent direction search method from current solution. The algorithm is easy to comprehend. The performance of this algorithm has been checked using a numerical example with known results. The flight landing problem is an application of Quadratic Assignment Problem. Since Quadratic Assignment Problem is NP-hard, it is quite tedious to get an optimal solution even for small size instances. However, as far as optimality of the solution is concerned, this algorithm leads to very favourable results.

Keywords: Allocation, Flight, Terminal, Optimal Solution, NP-hard.

1. Introduction
The landing and taking off the flights is an obvious phenomena at any national or international airport. Some passengers get down from one flight at one terminal and move away to get another flight at another terminal. Sometimes this movement of passengers at the airport results in a chaotic situation. Our prime aim is to reduce the chaos in any system. Considering this problem the flights must be allocated to the terminals so that the total distance covered by the passengers is minimized and the chaos caused by the movement of passengers could be reduced. Note that, the flights must be allocated to the terminals in such a way that no single flight is assigned to two different terminals and no single terminal is assigned to two different flights at the same time. The problem of allocating the flights to the terminals is actually an allocation problem, generally known as the assignment problem.

2. Mathematical formulation
Consider $n$ flights $F_1, F_2, \ldots, F_i, \ldots, F_n$ to be assigned at $n$ terminals $D_1, D_2, \ldots, D_k, \ldots, D_i, \ldots, D_n$ each at one as shown in the Figure 1.

Figure 1: Exhibiting the flights allocation to the terminals
Let $i^{th}$ flight $F_i$ be assigned at $k^{th}$ terminal $D_k$ and $j^{th}$ flight $F_j$ be assigned at $l^{th}$ terminal $D_l$. Let $f_{ij}$ be the number of passengers get down from $i^{th}$ flight and then travelling to get in the $j^{th}$ flight and $d_{il}$ be the distance from $k^{th}$ terminal to $l^{th}$ terminal. Then total distance covered by $f_{ij}$ number of passengers from $D_k$ to $D_l$ is $f_{ij}d_{il}$ and also total distance covered by $f_{ij}$ number of passengers from $D_l$ to $D_k$ is $f_{ij}d_{il}$. Then total two way distance covered between $D_k$ and $D_l$ terminals is $f_{ij}d_{il} + f_{ij}d_{il}$.

Therefore for $n$ flights and $n$ terminals the overall distance covered is $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ij}d_{il}x_{ik}x_{jl}$. The value of $x_{ik} = 1$ if $i^{th}$ flight is assigned at $k^{th}$ terminal otherwise zero i.e. $\sum_{k=1}^{n} x_{ik} = 1 \forall i = 1...n$ and the value of $x_{jl} = 1$ if $l^{th}$ terminal is assigned to $j^{th}$ flight otherwise zero i.e. $\sum_{j=1}^{n} x_{jl} = 1 \forall l = 1...n$. In general the flights, terminals and passengers are called facilities, locations and flow respectively. The problem is to assign the facilities to the locations such that there must be one-one correspondence between facilities and locations, so as to minimize total distance covered by total flow. i.e.

$$\text{Min } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ij}d_{il}x_{ik}x_{jl}$$

Such that $\sum_{k=1}^{n} x_{ik} = 1 \forall i = 1,...,r,...,n$.

$$\sum_{j=1}^{n} x_{jl} = 1 \forall l = 1,...,u,...,n.$$  

$x_{ik} \in \{0,1\}, x_{jl} \in \{0,1\}$

The mathematical formulation of above problem contains $2n$ constraints and $n^2$ decision variables. Out of $2n$ constraints one is always redundant. For the sake of brevity, without loss of generality the last constraint can be deleted and after that making left hand side equal to 0. The problem reduces to

$$\text{Min } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ij}d_{il}x_{ik}x_{jl}$$

Such that $\sum_{k=1}^{n} x_{ik} = 1 \forall i = 1,...,r,...,n$.

$$\sum_{j=1}^{n} x_{jl} = 1 \forall j = 1,...,u,...,n-1.$$  

$x_{ik} \in \{0,1\}, x_{jl} \in \{0,1\}$

Writing above formulation in compact form

$$\text{Min } Z = f(X)$$

Such that $g(X) = 0$

Where $g(X) = [g_1, g_2, ..., g_{2n-1}]$

$X = [x_{i1}, x_{i2}, ..., x_{ir}, ..., x_{i1}, x_{i2}, ..., x_{ir}, ..., x_{n1}, x_{n2}, ..., x_{nm}]$

$x_{ij} \in \{0,1\}$

It can easily be noticed that the summation form of objective function consists the terms of product of the two variables that makes objective function quadratic. Therefore the formulated problem is known as the Quadratic Assignment Problem (QAP).

This is stunning fact that the QAP appears in numerous real life problems. The QAP was first discussed by Koopmans and Beckmann [20] in 1957, where they had modelled the QAP as assigning a
set of economic activities to a set of locations such that no single activity is assigned to at two different locations and no single location is given to two different economic activities. The fundamental application of QAP was used by Dickey and Hopkins [8] in location theory in which they applied model of QAP in campus planning model. Furthermore QAPs are found in other applications such as development of control boards to minimize eye fatigue [24], backboard wiring [28], scheduling parallel production lines [14], typewriter keyboards and control panels [25], hospital planning [10], ranking of archaeological data [19] economic problems [17] statistical analysis [18], forest parks [5], analysis of reaction chemistry [12], numerical analysis [6] and placement of electronic components [23,9].

QAP has numerous real life applications as discussed earlier. Due to its applicability in daily life QAP draws attention of mathematicians towards it. There are several exact method such as branch and bound method [4,13,15,16,21,28], branch and cut algorithm [11], dynamic programming, lexisearch algorithm [1], cutting plane technique [3] and heuristic methods such as robust taboo search [29], branch and bound heuristic method [2] simulated annealing scheme [7], greedy randomized adaptive search procedure [22] etc. have been developed for solving QAP.

3. Mathematical background
QAP is a combinatorial optimization problem which is very hard to solve. It had been shown by Shani and Gonzales [26] that QAP is NP-hard problem. Therefore exact methods have some limitations in terms of time required to get optimal solution is too large and as problem size increases methods fail to get optimal solution. Similarly obtainment of optimal solution by using heuristics is by chance but in many situations optimal solution is direly required. Therefore an algorithm based on steepest descent direction search method has been developed for solving QAP optimally.

The solution procedure begins by considering an initial basic feasible solution i.e. \( x_i = 1, \forall i = j \) and \( x_i = 0, \forall i \neq j \). The \( n^2 \) decision variables are divided into two sets; set of \( (2n-1) \) variables is called set of basic variables \( (X_b) \) and set of remaining \( (n-1)^2 \) variables is called set of non-basic variables \( (X_n) \). Moreover, number of \( x_{ij} \)'s belonging to \( X_n \) which are equal to 1, are \( n \) in numbers resulting into \( (n-1) \) \( x_{ij} \)'s are equal to zero in the set \( X_n \).

The thought behind to this proposed method is to search feasible steepest descent direction from current solution. In order to do that the objective function \( f(X) \) and constraints function \( g(X) = [g_1, g_2, ... g_k] \) of QAP are approximated by Taylor’s approximation. Since \( f(X) \) is quadratic and \( g(X) \) is linear therefore in the approximation of \( f(X) \) first and second order gradient terms are appeared and in approximation of \( g(X) \) only first order gradient terms are appeared. Second order gradient of \( f(X) \) is nothing but the hessian matrix of order \( n^2 \) and first order gradient of constraints function \( g(X) \) is the Jacobian of size \( (2n-1) \times (n-1)^2 \). The Jacobian of \( g(X) \) which is computed with respect to basic variables \( X_b \) is denoted by \( J_b \) and that of with respect to non-basic variables \( X_n \) is denoted by \( J_n \). Note that variables in \( X_b \) are taken in such a way that the Jacobian \( J_b \) is always non-singular. Therefore feasible change \( (\partial X_n) \) in basic variables \( X_b \) can be calculated with respect to feasible change \( (\partial X_n) \) in non-basic variables \( X_n \) and hence change in value of objective function \( (\partial f(X)) \) can easily be computed by substituting value of \( \partial X_n \) in terms of \( \partial X_n \) in the expression of \( \partial f(X) \). If \( (\partial f(X))_k \) be the maximum negative change in the
value of objective function with respect to change \( \partial x_{a_k} \) in \( k^{th} \) non-basic variable \( x_{a_k} \) then \( x_{a_k} \) will enter in the basis and new solution will be calculated.

4. Algorithm

STEP 0: Write \( F \) and \( D \) matrices.

STEP 1: Construct an initial basic feasible solution as follow.

\[
X = \{x^1\}
\]

\[
\hat{X}_0 = [x^1_i]_{i=1, \ldots, n}
\]

Where \( x^1_i \) = \begin{cases} 1 & \forall i = j \\ 0 & \text{otherwise} \end{cases}

STEP 2: Calculate the cost of the assignment with respect to initial basic feasible solution.

\[
Z = \sum_{i=1}^{n} \sum_{j=1, \ldots, n} f_{ij} d_{ij} x_{ijk}
\]

STEP 3: Divide \( X \) into two sets, set of basic variables say \( (X_B) \) and set of non-basic variables say \( (X_N) \). Remember, variables in \( X_N \) preserves the order of sequence of variables of \( X \).

Where

\[
X_B = I \cup O_c = \{x_{b1}, x_{b2}, \ldots, x_{bn}, \ldots, x_{n(2n-1)}\}
\]

\[
I = \{x^1_i : x^1_j = 1\}
\]

\[
O_c = \{x^r_x : x^s_x = 0, \forall r = 2, 3, \ldots, n\}
\]

\( r \): First suffix value of \( r^{th} \) element in set \( I \)

\( s \): Second suffix value of \( (r-1)^{th} \) element in set \( I \)

\[
X_N = X - X_B = \{x_{n1}, x_{n2}, \ldots, x_{nk}, \ldots, x_{n(n-1)}\}
\]

\[
\hat{X}_C = [\hat{X}_B \hat{X}_N]^T
\]

STEP 4: Construct a set containing all the variables of the set \( O_c \). This set is called the set of replaceable variables and is denoted by \( R \).

\( R = O_c \) and go to STEP 6.

STEP 5: Calculate

\[
X_B = I \cup O_{new} = \{x_{b1}, x_{b2}, \ldots, x_{bn}, \ldots, x_{n(2n-1)}\}
\]

\[
X_N = X - X_B = \{x_{n1}, x_{n2}, \ldots, x_{nk}, \ldots, x_{n(n-1)}\}
\]

\[
\hat{X}_C = [\hat{X}_B \hat{X}_N]^T
\]

STEP 6: Calculate \( \nabla_{X_N} g(X) = J_B \) and \( \nabla_{X_N} g(X) = J_N \) matrices.

STEP 7: Calculate \( J_B^{-1} \).
STEP 8: Calculate $A = J_B^{-1} J_N'$. 

Where $A = \left[ a_{pq} \right]_{(2n-1) \times (n-1)^2}$

STEP 9: If $X_B \neq I \cup O_{new}$ then construct a matrix $P$ as follow.

$$P = \left[ P_{pq} \right]_{(2n-1) \times (n-1)^2}$$

The matrix $P$ is called pivot matrix.

Otherwise go to STEP 10.

STEP 10: Calculate $W_q, \forall q = 1, 2, \ldots, (n-1)^2$. Note that

$$\frac{\partial f(X)}{\partial x_{rt}} = 2 \sum_{u=1}^{n} f_{rs} a_{ru} x_{su}$$

$$W_q = \left[ \frac{\partial f(X)}{\partial x_{Nq}} - \sum_{p=1}^{2n-1} \frac{\partial f(X)}{\partial x_{Bp}} a_{pq} \right]_{X = X_i}$$

STEP 11: Calculate $W_{qq}, \forall q = 1, 2, \ldots, (n-1)^2$. Note that

$$\frac{\partial^2 f(X)}{\partial x_{rt} \partial x_{su}} = 2 f_{rs} a_{ru} \quad \forall r, s = 1, 2, \ldots, n.$$ 

$$W_{qq} = \left[ \sum_{p=1}^{2n-1} \sum_{p' \neq p} a_{pq} \frac{\partial^2 f(X)}{\partial x_{Bp} \partial x_{Bp'}} a_{pq} - 2 \sum_{p=1}^{2n-1} a_{pq} \frac{\partial^2 f(X)}{\partial x_{Bp} \partial x_{Nq}} \right]$$

STEP 12: Calculate $(\tilde{\partial f})_q = W_q + \frac{1}{2} W_{qq}$.

$$\mathcal{W} = \left\{ (\tilde{\partial f})_1, (\tilde{\partial f})_2, \ldots, (\tilde{\partial f})_k, \ldots, (\tilde{\partial f})_{(n-1)^2} \right\}$$

- If $\alpha = (\tilde{\partial f})_k = \min_i (\tilde{\partial f})_i < 0$ go to STEP 14.

- If $\alpha = (\tilde{\partial f})_k = \min_i (\tilde{\partial f})_i \geq 0$ go to STEP 19.

STEP 13: Calculate $\mathcal{W}' = \mathcal{W} - \left\{ (\tilde{\partial f})_k \right\}$.

- If $\alpha_{new} = (\tilde{\partial f})_k = \min_i (\tilde{\partial f})_i < 0$ put $\mathcal{W}' = \mathcal{W}', k = k'$ and go to STEP 14.

- If $\alpha_{new} = (\tilde{\partial f})_k = \min_i (\tilde{\partial f})_i \geq 0$ go to STEP 19.

STEP 14: Calculate

$$\left[ \frac{\partial x_{Nq}}{\partial x_{(n-1)^2+i}} \right]_{(n-1)^2} = e_k$$

Where $e_i = \left[ 0, \ldots, 0, \hat{1}, 0, \ldots, 0 \right] \Rightarrow k$

Here $\hat{1}$ means 1 is the $k^{th}$ entry.

Let $\tilde{\mathcal{X}}_N = \left[ \frac{\partial x_{Nq}}{\partial x_{(n-1)^2+i}} \right]_{(n-1)^2}$

\( \forall q = 1, 2, \ldots, (n-1)^2 \)}
STEP 15: Calculate
\[ \hat{c}x_{bp} = -a_{pk} \forall p = 1,2,\ldots,h,\ldots,(2n-1). \]
Let \( \hat{\partial} \hat{X}_b = \left[ \hat{\partial}x_{bp} \right]_{(2n-1)\times 1} \)

STEP 16: Calculate
\[ \hat{\partial} \hat{X}_c = \left[ \hat{\partial} \hat{X}_b \hat{\partial} \hat{X}_N \right]^T \]

STEP 17: Calculate
\[ \hat{X}_{next} = \hat{X}_c + \hat{\partial} \hat{X}_c \]
\[ X = X_{next} \]
If \( X_{next} \) is feasible solution then go to STEP 18. Otherwise go to STEP 13.

STEP 18: Calculate
\[ Z^* = Z + \alpha \] and go to STEP 3.

STEP 19: If \( R \neq \emptyset \)
If position of \( x_{rs} \in R \) is \( p \) in \( X_B \) then search all \( P_{pq} = 1, \forall q = 1,2,\ldots,(n-1)^2 \) in \( P \). If \( P_{pq} = 1 \) in \( P \) then \( p^{th} \) element \( (x_{rs} \in X_B) \) can be replaced by \( q^{th} \) element \( (x_m \in X_N) \).
Therefore construct \( R_p = \{ X : x_m \in X_N \) those which are corresponding to \( P_{pq} = 1 \} \)
Otherwise go to STEP 23.

STEP 20: Replace first element \( x_n \) of \( R \) by first element \( x_m \) of \( R_p \) and update \( R_p \) as follow.
\[ R'_p = R_p - \{x_m\} \]
\[ R_p = R'_p \]

STEP 21: If \( R_p = \emptyset \) then
\[ R = R - \{x_{rs}\} \]
\[ R = R' \]
Otherwise go to STEP 22.

STEP 22: Calculate
\[ O_{new} = O - \{x_{rs}\} \cup \{x_m\} \] go to STEP 5.

STEP 23: There will be no change in the value of objective function. Therefore current solution is optimal or nearest to optimal and value of objective function is \( Z^* \).

5. Numerical Example
There are 3 flights and 3 terminals and the number of passengers are travelling from one terminal to another terminal and the distances between these terminals are given in matrix form as follow. The problem is to assign these flights so that the total distance covered by passengers could be minimized. The step by step solution of this problem based on our algorithm is presented as follow.

ITERATION 1:
STEP 0: Enter the flow and distance matrices.
\[
F = \begin{bmatrix} F_1 & F_2 & F_3 \end{bmatrix} \quad D = \begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix}
\]
\[
F = \begin{bmatrix} 1 & 0 & 5 \\ 1 & 0 & 3 \\ 5 & 3 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 0 & 4 \\ 3 & 4 & 0 \end{bmatrix}
\]
STEP 1: Construct an initial basic feasible solution.
\[ x_1 = 1, x_2 = 0, x_3 = 0, x_1 = 0, x_2 = 0, x_3 = 0, x_3 = 1 \]

STEP 2: Calculate the cost of the assignment with respect to initial basic feasible solution
\[ Z = 1 \times 2 + 3 \times 3 + 4 + 2 + 3 \times 3 \times 4 = 58 \]

STEP 3: Divide \( X \) into two sets, set of basic variables say \( B \) and set of non-basic variables say \( N \). Where
\[ B = \{ x_{11}, x_{22}, x_{33}, x_{21}, x_{32} \} \]
\[ N = \{ x_{12}, x_{23}, x_{31} \} \]

STEP 4: Construct the set of replaceable variables \( R \).
\[ R = \{ x_{12}, x_{33} \} \]
go to STEP 6.

STEP 6: Calculate \( \nabla_{x_B} g(X) = J_B \) and \( \nabla_{x_N} g(X) = J_N \).
\[
J_B = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
\end{bmatrix}
\]
\[
J_N = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

STEP 7: Calculate \( J_B^{-1} \).
\[
J_B^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 \\
1 & 1 & 1 & -1 & -1 \\
-1 & 0 & 0 & 1 & 0 \\
-1 & -1 & 0 & 1 & 1 \\
\end{bmatrix}
\]

STEP 8: Calculate \( A = J_B^{-1} J_N \).
\[
A = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
-1 & -1 & 0 & 1 \\
0 & -1 & -1 & 1 \\
\end{bmatrix}
\]

STEP 9: Since \( X_B \neq I \cup O \), then construct the pivot matrix \( P \).
\[
P = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 1 & 1 & -1 \\
0 & 1 & 1 & 0 \\
-1 & -1 & 0 & 1 \\
0 & -1 & -1 & 1 \\
\end{bmatrix}
\]

STEP 10: Calculate \( W_q, \forall q = 1, 2, 3, 4 \).
\[ W_1 = -4, W_2 = -70, W_3 = -56, W_4 = 2 \]

STEP 11: Calculate \( W_q, \forall q = 1, 2, 3, 4 \).
\[ W_1 = 16, W_2 = 140, W_3 = 96, W_4 = 48 \]

STEP 12: Calculate \( (\tilde{c}f)_q = W_q + \frac{1}{2} W_{qr} \) \( \forall q = 1, 2, 3, 4 \).
\[ \forall \mathcal{V} = \{4, 0, -8, 26\} \quad \alpha = (\tilde{c}f)_1 = -8 < 0 \] go to STEP 14.
STEP 14: Calculate \[ \hat{\mathbf{x}}_{i_0} = \mathbf{e}_j = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \quad \forall q = 1, 2, 3, 4. \]
\[ \partial X_n = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \]

STEP 15: Calculate \[ \hat{x}_{q_0} = -\alpha_{p_3} \quad \forall p = 1, 2, \ldots, 5. \]
\[ \partial X_p = \begin{bmatrix} 0 & -1 & -1 & 0 & 1 \end{bmatrix} \]

STEP 16: Calculate \[ \hat{X}_{\text{new}} = X_c + \partial X_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad X = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \]

STEP 17: Calculate \[ \hat{\mathbf{x}}_{i_0} = \mathbf{e}_j = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \quad \forall q = 1, 2, 3, 4. \]

STEP 18: Calculate \[ Z' = Z - \alpha = 58 + (-8) = 50 \]

ITERATION 2

STEP 3: Divide \( X \) into two sets, set of basic variables say \( (X_B) \) and set of non-basic variables say \( (X_N) \).

Where \[ X_B = I \cap O_x = \{x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}\} \quad I = \{x_{i_1}, x_{i_2}, x_{i_3}\} \]
\[ O_x = \{x_{i_2}, x_{i_3}\} \quad X_N = X - X_B = \{x_{i_2}, x_{i_1}, x_{i_2}, x_{i_1}\} \]
\[ \hat{X}_c = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

STEP 4: Construct the set of replaceable variables \( \mathcal{R} \).
\[ \mathcal{R} = \{x_{i_1}, x_{i_3}\} \] go to STEP 6.

After calculating STEP 6, STEP 7 and STEP 8.

STEP 9: Since \( X_B \neq I \cap O_x \) then construct the pivot matrix \( P \).
\[ P = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & -1 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 1 \\ -1 & 0 & -1 & 1 \end{bmatrix} \]

STEP 10: Calculate \( W_q, \forall q = 1, 2, 3, 4. \)
\[ W_1 = -50, \quad W_2 = -4, \quad W_3 = -40, \quad W_4 = 6 \]

STEP 11: Calculate \( W_{q_0}, \forall q = 1, 2, 3, 4. \)
\[ W_{q_1} = 124, \quad W_{q_2} = 24, \quad W_{q_3} = 96, \quad W_{q_4} = 72 \]

STEP 12: Calculate \( (\partial f)_{q_0} = W_4 + \frac{1}{2} W_{q_0} \quad \forall q = 1, 2, 3, 4. \)
\[ \mathcal{V} = \{12, 8, 8, 42\} \quad \alpha = (\partial f)_{q_0} = 8 > 0 \] go to STEP 19.

STEP 19: Since \( \mathcal{R} \neq \emptyset \)
The position of \( x_{i_3} \in \mathcal{R} \) is \( p = 4 \) in \( X_B \) then search all \( P_q = 1, \forall q = 1, 2, 3, 4. \) in \( P \). Since \( P_4 = 1 \) in \( P \) then 4th element \( (x_{i_3} \in X_B) \) can be replaced by \( 4^\text{th} \) element \( (x_{i_1} \in X_N) \).
Therefore \( \mathcal{R}_p = \{x_{i_1}\} \)

STEP 20: Replace first element \( x_{i_1} \) of \( \mathcal{R} \) by first element \( x_{i_1} \) of \( \mathcal{R}_p \) and update \( \mathcal{R}_p \).
\[ \mathcal{R}_p' = \mathcal{R}_p - \{x_{i_1}\} = \emptyset \]
\[ \mathcal{R}_p = \emptyset \]
STEP 21: Since $\mathcal{R}_p = \emptyset$

\[ \mathcal{R}' = \mathcal{R} - \{x_{21}\} = \{x_{33}\} \]

\[ \mathcal{R} = \{x_{51}\} \]

STEP 22: Calculate

\[ O_{\text{new}} = O_e - \{x_{5}\} \cup \{x_{n}\} = \{x_{21}, x_{33}\} - \{x_{21}\} \cup \{x_{33}\} = \{x_{33}\} \]

go to STEP 5.

ITERATION 3

STEP 5: Calculate

\[ X_g = I \cup O_{\text{new}} = \{x_{11}, x_{22}, x_{33}, x_{33}\} \]

\[ \hat{X}_c = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

\[ X = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \]

After calculating STEP 6, STEP 7 and STEP 8.

STEP 10: Calculate $W_{q_i}$, $\forall q = 1, 2, 3, 4$.

\[ W_1 = -44, \quad W_2 = 2, \quad W_4 = -40, \quad W_4 = -6 \]

STEP 11: Calculate $W_{q_i}$, $\forall q = 1, 2, 3, 4$.

\[ W_1 = 80, \quad W_{22} = 120, \quad W_9 = 96, \quad W_{44} = 72 \]

STEP 12: Calculate \( (\partial f)_{q_i} = W_{q_i} + \frac{1}{2} W_{q_i} \)

\[ \forall q = 1, 2, 3, 4. \]

\[ \mathcal{W} = \{-4, 62, 8, 30\} \]

\[ \alpha = (\partial f)_i = -4 < 0 \]

go to STEP 14.

After calculating the STEP 14, STEP 15 and STEP 16.

STEP 17: Calculate

\[ \hat{X}_{\text{new}} = X_c + \partial X_c = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \]

\[ X = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \]

STEP 18: Calculate

\[ Z' = Z + \alpha = 50 + (-4) = 46 \]

go to STEP 3.

ITERATION 4

STEP 3: Divide $X$ into two sets, set of basic variables say \( (X_g) \) and set of non-basic variables say \( (X_n) \).

Where

\[ X_g = I \cup O_e = \{x_{11}, x_{22}, x_{33}, x_{33}\} \]

\[ I = \{x_{12}, x_{23}, x_{31}\} \]

\[ O_e = \{x_{22}, x_{33}\} \]

\[ X_n = X - X_g = \{x_{11}, x_{33}\} \]

\[ \hat{X}_c = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

STEP 4: Construct the set of replaceable variables.

\[ \mathcal{R} = \{x_{22}, x_{33}\} \]

After calculating STEP 5, STEP 6, STEP 7 and STEP 8.

STEP 9: Since $X_g \neq I \cup O_{\text{new}}$ then construct pivot matrix $P$.

\[ P = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 1 \\ -1 & 0 & -1 & 1 \end{bmatrix} \]

After calculating STEP 10 and STEP 11.
STEP 12: Calculate \( \bar{f}_q = W_q + \frac{1}{2}W_{qq} \) \( \forall q = 1, 2, 3, 4 \).

\[ \forall q = 1, 2, 3, 4 \]

\[ \alpha = (\bar{f}_q) = 4 > 0 \text{ go to STEP 19}. \]

STEP 19: Since \( R \neq \emptyset \)

The position of \( x_{q2} \in R \) is \( p = 4 \) in \( X_q \) then search all \( P_{q4} = 1, \forall q = 1, 2, 3, 4 \) in \( P \). Since \( P_{q4} = 1 \) in \( P \) then 4\textsuperscript{th} element \( (x_{q2} \in X_q) \) can be replaced by 4\textsuperscript{th} element \( (x_{q2} \in X_q) \).

Therefore \( R_p = \{ x_{q2} \} \).

STEP 20: Replace first element \( x_{q2} \) of \( R \) by first element \( x_{q3} \) of \( R_p \) and update \( R_p \).

\[ R'_p = R_p - \{ x_{q2} \} = \emptyset \]

\[ R_p = \emptyset \]

STEP 21: Since \( R_p = \emptyset \) then

\[ R = R - \{ x_{q3} \} = \emptyset \]

\[ R = \emptyset \]
STEP 22: Calculate

\[ O_{\text{new}} = O - \{x_n\} \cup \{x_m\} = \{x_{21}, x_{33}\} - \{x_{11}, x_{13}, x_{21}, x_{33}\} = \{x_{22}, x_{32}\} \]

go to STEP 5.

ITERATION 6:

STEP 5: Calculate

\[ X_B = I \cup O_{\text{new}} \]
\[ X_N = X - X_B = \{x_{11}, x_{13}, x_{21}, x_{33}\} \]
\[ \hat{X}_e = [1, 1, 0, 0, 0, 0, 0, 0] \]

After calculating STEP 6, STEP 7, STEP 8, STEP 10 and STEP 11.

STEP 12: Calculate

\[ (\vec{\alpha} f)_q = \frac{1}{2} \frac{\alpha}{W_q} \]
\[ \forall q = 1, 2, 3, 4. \]
\[ \mathcal{W} = \{4, 4, 2, 50\} \]

STEP 23: There will be no change in the value of objective function. Therefore current solution

\[ x_{11} = 0, x_{12} = 1, x_{21} = 0, x_{22} = 0, x_{33} = 1, x_{32} = 0, x_{33} = 0 \]

is optimal solution and value of objective function is \( Z^* = 46 \).

According to the solution the 1st flight, 2nd flight and the 3rd flight must be assigned at 2nd terminal, 3rd terminal and 1st terminal respectively to minimize the total distance covered by the passengers. The distance travelled by the passengers is 46 units with respect to this assignment which is minimum among all the assignment.

6. Results and discussion

We have developed an algorithm for solving QAP and applied it on randomly generated problem of size 3 (where size 3 stands for number of flights and terminals) in this paper. It is found that the problem of size 3 can be solved up to optimal solution whereas problem of size 4 and greater can be solved up to optimal or up to nearest to optimal solution. For example one problem of size 4 solved in the paper of S S Syed Abdullah [27] by applying quadratic assignment problem with fixed assignment (QAPFA) using branch and bound approach. The objective function value corresponding to the solution obtained by QAPFA is \( 348 \). Whereas the objective function value corresponding to the solution \( x_{41} = 1, x_{42} = 1, x_{31} = 1, x_{32} = 1 \) (remaining variables are equals to 0) obtained by our algorithm is \( 302 \). The optimal solution of this problem is \( x_{12} = 1, x_{22} = 1, x_{31} = 1, x_{42} = 1 \) (remaining variables are equals to 0) and corresponding objective function value is \( 294 \). The obtained solution by using our algorithm is nearest to optimal and far better than the QAPFA.

On the other hand we have solved another problem, \( F = [0 2 1 4; 2 0 1 2; 1 1 0 1; 4 2 1 0] \) and \( D = [0 2 2 4; 2 0 3 1; 2 3 0 2; 4 1 2 0] \), of size 4 by using our algorithm up to optimal solution \( x_{12} = 1, x_{23} = 1, x_{31} = 1, x_{44} = 1 \) and corresponding objective function value is \( 44 \).

The limitation of this method is that it takes at least \( (n-1) \) extra iterations to terminate after reaching the solution since in the set of basic variables \( X_s \) contains \( (n-1) \) variables which are equal to 0. We replace these \( (n-1) \) variables by non-basic variables after reaching the solution from where there is no descent direction. After replacing all these variables by non-basic variables the algorithm terminates in finite number of iterations.

7. Conclusion

Various instances of QAP of size 3 and greater have been solved using our algorithm. This algorithm works on the principle of steepest descent direction search method. The instances of size 3 can be solved optimally whereas some problems of size greater than 3 can be solved nearest to optimal and
some can be solved up to optimal solution. It can be clearly concluded that if the considered initial basic feasible solution is nearer to the optimal solution then optimal solution can be obtained by performing lesser number of iterations using proposed algorithm. So one can propose an algorithm for finding an initial basic feasible solution which is nearest to optimal and then this algorithm can be applied to get an optimal solution.

8. Acknowledgements
We are very much thankful to the reviewers for their valuable comments and suggestions which help us to improve the paper. This research is supported by the Rajiv Gandhi National Fellowship for SC Candidate, UGC ID: F1-17.1/2015-16/RGNF-2015-17-SC-UTT-21377/(SA-III/Website).

References
[1] Ahmed Z H 2014 A data-guided lexisearch algorithm for the quadratic assignment problem Indian journal of Science and Technology 7(4) 480-490.
[2] Bazara M S and Kirca O 1983 A branch-and-bound based heuristic for solving the quadratic assignment problem Naval Research Logistics (NRL) 30(2) 287-304.
[3] Bazara M S and Sherali H D 1982 On the use of exact and heuristic cutting plane methods for the quadratic assignment problem The Journal of Operational Research Society 33(11) 991-1003.
[4] Bazara M S and Elshafei A N 1979 An exact branch and bound procedure for quadratic assignment problems Naval Research Logistics Quarterly 26(1) 109-120.
[5] Bose J 1993 A quadratic assignment problem solved by simulated annealing Journal of Environmental Management 37(2) 127-145.
[6] Brusco M J and Sahni S 2000 Using quadratic assignment problem to generate initial permutations for least-squares unidimensional scaling of symmetric proximity matrices Journal of Classification 17(2) 197-223.
[7] Connolly D T 1990 An Improved annealing scheme for the quadratic assignment problem European Journal of Operational Research 46 93–100.
[8] Dickey J W and Hopkins J W 1972 Campus building arrangement using TOPAZ Transportation Resear, 6 59-68.
[9] Duman E and Ilhan O 2007 The quadratic assignment problem in the context of the printed circuit board assembly process Computers and Operations Research 34(1) 163-179.
[10] Elshafei A N 1977 Hospital layout as a quadratic assignment problem Journal of the Operational Research Society 28(1) 167-179.
[11] Erdogan G andTansel B A 2007 Branch-and-cut algorithm for the quadratic assignment problems based on linearizations Computers & Operations Research 34(4) 1085-1106.
[12] Forsberg J H et. al. 1994 Analyzing lanthanide-included shifts in the NMR spectra of lanthanide (III) complexes derived from 1,4,7,10-tetraakis (N,N-diethylacetamido)-1,4,7,10-tetraazacyclododecane Inorganic Chemistry 34 3705-3715.
[13] Gavett J W and Plyter N V 1966 The optimal assignment of facilities to locations by branch and bound Operations Research 14(2) 210-232.
[14] Geoffrion A M and Graves G W 1976 Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach Operations Research 24(4) 595–610.
[15] Gilmore P 1962 Optimal and suboptimal algorithms for the quadratic assignment problem Journal of the Society of Industrial and Applied Mathematics 10(2) 305–313.
[16] Hahn P et. al. 1998 A branch-and-bound algorithm for the quadratic assignment problem based on Hungarian method European Journal of Operational Research 108(3) 629-640.
[17] Heffley D R 1980 Decomposition of the Koopmans- Beckmann problem Regional Science and Urban Economics 10(4) 571-580.
[18] Hubert L 1987 Assignment methods in combinatorial data analysis *Statistics: Textbook and Monographs Series*, 73 Marcel Dekker.
[19] Krarup J and Pruzan P M 1978 Computer-aided layout design *Mathematical Programming Study* 9 75-94.
[20] Koopmans T C and Beckmann M J 1957 Assignment problems and the location of economic activities *Econometrica* 25(1) 53-76.
[21] Lawler E L 1963 The quadratic assignment problem *Management Science* 9(4) 586-599.
[22] Li Y, Pardalos P M and Resende M G 1994 A greedy randomized adaptive search procedure for the quadratic assignment problem *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 16 237–261.
[23] Miranda G et.al. 2005 A performance guarantee heuristic for electronic components placements problems including thermal effects *Computers and Operations Research* 32(11) 2937-2997.
[24] McCormick E J 1970 Human factors engineering McGraw-Hill New York
[25] Pollatschek M A et.al. 1976 Optimization of the typewriter keyboard by simulation Angewandte Informatik 10 438-439.
[26] Sahni S and Gonzales T 1976 P-complete approximation problems *Journal of the Association for Computing Machinery (JACM)* 23 555-565.
[27] Sharifah Shuthairah Syed-Abdullah et al 2018 *IOP Conf. Ser. :Mater. Sci. Eng.* 300 012002
[28] Steinberg L 1961 The backboard wiring problem: A placement algorithm, *SIAM Review* 3(1) 37-50.
[29] Taillard E D 1991 Robust taboo search for the quadratic assignment problem *Parallel Computing* 17(4-5) 443-455.