Nonlinear Functional Analysis and Applications
Vol. 25, No. 3 (2020), pp. 545-562
ISSN: 1229-1595(print), 2466-0973(online)

https://doi.org/10.22771/nfaa.2020.25.03.10
http://nfaa.kyungnam.ac.kr/journal-nfaa

EXTENSION OF EGERVÀRY THEOREM ON OPTIMAL SOLUTION OF ASSIGNMENT PROBLEM: LOGICAL APPROACH

Susanta Kumar Mohanta[1] and Prasanta Kumar Das[2]

[1]Department of Mathematics
Government College Sundargarh, Odisha-770002, India
e-mail: smohanta129@gmail.com

[2]School of Applied Sciences (Mathematics)
KIIT Deemed to be University Odisha-751024, India
e-mail: dasprasantkumar@yahoo.co.in

Abstract. A technique to solve the balanced linear assignment problem is introduced using graph theory and is based on logical approach. In the method, the aim is to find a matching in which the sum of weights of the edges is as large as possible, in a weighted bipartite graph. Generally it consists of finding a minimum-weight perfect matching and is a specialization of the maximum weight matching problem for bipartite graphs. The concept of decision matrix ([1]) is applied for finding last two assignments.

1. Introduction

The assignment problem is a fundamental combinatorial optimization problem. In the course of time several methods and algorithms has been developed to solve assignment problems for more specific variations of its formulation, out of all these Hungarian method is commonly used for its simplicity. These approaches do not always find the true optimal solution. Instead, they will often consistently find good solutions to the problems. These good solutions are typically considered to be good enough simply because they are the best

[0]Received March 4, 2020. Revised April 17, 2020. Accepted April 21, 2020.
[0]2010 Mathematics Subject Classification: 90B06, 90B10, 90C08, 90C60, 90C90.
[0]Keywords: Algorithmic complexity, assignment problem, favorable cost, opportunity cost matrix, optimal solution, perfect matching, decision matrix.
[0]Corresponding author: P. K. Das(dasprasantkumar@yahoo.co.in).

that can be found in a reasonable amount of time. Therefore, optimization often takes the role of finding the best solution possible in a reasonable amount of time. The proposed logical approach is studied using modified Egerváry theorem with numerical examples and comparative study on its algorithmic complexity. This methods gives a true optimal solution to the assignment problem with reasonable short time.

In section 2, some results of assignment problem based on graph theory are revisited. In section 3, a new method of assignment problem is defined and studied some results on assignment problem based using graph theory, logical approach and DAS technique proposed by Das et al. ([1]). In section 4, some concrete examples are given to establish the theorems. In section 5, algorithm complexity of the method is discussed. In section 6, order of the method is analyzed. Finally in section 7, the conclusion of the paper are discussed.

2. Preliminaries

The problem is as follows: The assignment problem has a number of worker and same number of tasks. Any worker be assigned to perform any task, incurring some cost that may vary depending on the worker-task assignment. It is required to perform all tasks by assigning exactly one worker to each task and exactly one task to each worker in such a way that the total cost of the assignment is minimized.

Kuhn ([3]) developed and published the Hungarian method in 1955; is a combinatorial optimization algorithm that solves the assignment problem in polynomial time. Munkres ([4]) studied the algorithm and observed that it is strongly polynomial. In this article our main focus is to develop a favorable matching of edges that will minimizing the total cost of the assignment problem, we recall some definitions and results for our needs.

2.1. **Some known results.**

Let G be a graph having E(G) as a set of edges and V(G) is a set of vertices.

Definition 2.1. ([2]) Let G be a graph and $M \subseteq E(G)$. Then M is a matching in G if no two edges of M have a common end-vertex. We say that M is a maximum matching if it has maximum cardinality over all matchings in G. A vertex $v \in V(G)$ is M-saturated if v is incident with an edge of M. We say that M is a perfect matching in G if every vertex of G is M-saturated.

Thus, if M is a perfect matching, then $|M| = \frac{1}{2}|V(G)|$ and M is necessarily a maximum matching. Let match(G) denote the size of a maximum matching in G.

Definition 2.2. ([2]) Let G be a graph and U $\subseteq$ V(G). We say that U is a cover of G if every edge of G is incident with a vertex in U. We say that U is a minimum cover if it has minimum cardinality over all covers of G. Let cov(G) denote the size of a minimum cover of G.

Definition 2.3. ([2]) The complete bipartite graph $K_{m;n}$ is the bipartite graph with bipartition $\{X;Y\}$ where $|X| = m$, $|Y| = n$ and each vertex of X is adjacent to every vertex of Y.

Theorem 2.4. (D. König, 1931) ([2]) *Let* G *be a bipartite graph. Then match* (G) = (G).

Let N be a network obtained from $K_{m;m}$ by giving each edge e an integer weight $w(e)$. A perfect matching of maximum weight in N can be represent as w(M).

Definition 2.5. ([2]) A feasible vertex labeling for N is a function $l : V(N) \to \mathbb{Z}$ such that $l(x) + l(y) \geq w(xy)$ for all $x \in X$ and $y \in Y$. We define the size of l, by

$$\text{size}(l) = \sum_{v \in V(N)} l(v).$$

Lemma 2.6. ([2]) *Let* l *be a feasible vertex labeling for* N *and* M *be a perfect matching in* N. *Then* $w(M) \leq \text{size}(l)$.

Definition 2.7. ([2]) Let l be a feasible vertex labeling of N. Then the equality subgraph $G(l)$ for l in N is the spanning subgraph of N containing all edges xy for which $l(x) + l(y) = w(xy)$.

Lemma 2.8. ([2]) *Let* l *be a feasible vertex labeling for* N *and* M *be a perfect matching in the equality subgraph* $G(l)$. *Then* $w(M) = \text{size}(l)$ *and hence* M *is a maximum weight perfect matching in* N *and* l *is a minimum size feasible vertex labeling of* N.

Theorem 2.9. (Egerváry [2], 1931) *Let* N *be a weighted complete bipartite graph. Then the maximum weight of a perfect matching in* N *is equal to the minimum size of a feasible vertex labeling of* N.

Lemma 2.10. ([2]) *Let* N *be a network obtained from* $K_{m;m}$ *by giving each edge* e *an integer weight. Then the number of times the method grows an alternating forest is at most* $2m^2$ *if maximal weight perfect matching in* N *is constructed by Hungarian method.*

Theorem 2.11. ([2]) *Suppose* N *is a network obtained from* $K_{m;m}$ *by giving each edge e an integer weight. Then Hungarian method finds a maximum weight perfect matching in* N *in time* $O(m^4)$*, under the assumption that all elementary arithmetic operations take constant time.*

In 2014, Das et al. ([1]) has developed dominated assignment simulation Technique (DAST) to solve the assignment problem (balanced or unbalanced) via a decision square matrix of order 2 obtained from the cost matrix and is of order n^2.

Definition 2.12. ([1]) The decision cost matrix is a square matrix of order 2 given by

$$D = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$$

obtained from a simulated assignment problem of order n using DAS-technique where d_{11}, d_{22} are diagonals and d_{12}, d_{21} are off-diagonals. Trace and off-trace of the D are

$$\mathrm{tr}\,(D) = d_{11} + d_{22} \text{ and } \mathrm{offtr}\,(D) = d_{12} + d_{21}$$

respectively. In the sense of Das et al. ([1]), the two important assignments a_{n-1} and a_n selected from the decision matrix of the simulated assignment problem are

$$\{a_{n-1}, a_n\} = \begin{cases} \{d_{11}, d_{22}\}, & \text{if } \mathrm{tr}\,(D) < \mathrm{offtr}\,(D) \text{ for minimized SAP;} \\ \{d_{11}, d_{22}\}, & \text{if } \mathrm{tr}\,(D) > \mathrm{offtr}\,(D) \text{ for maximized SAP.} \end{cases}$$

In 2018 Mohanta ([5]) studied an optimal solution to the transportation problem using the favorable cost for each source and destination.

3. Logical approach and computational method

Let N be a network obtained from a complete bipartite graph $K_{m;m}$ with bipartition $\{X; Y\}$ such that $|X| = m$, $|Y| = m$, $V(N) = X \cup Y$ and M be a perfect matching for N. In the network N each vertex $x \in X$ is adjacent to all vertex $y \in Y$. From Figure-1: vertex $x_1 \in X$ has m edges such as $x_1y_1, x_1y_2, \cdots, x_1y_m$ with each edge has an integer weight $w_{11}, w_{12}, \cdots, w_{1m}$ respectively. Let weight of a vertex $w_1 = w(x_1) = \sum_{j=1}^{m} w_{1j}$ be the total weight of all the edges incident on the vertex x_1. The weight represent a single homogeneous components like distance or cost or time. Main focus of this article is to generate a set of edges (favorable matching) that will minimize the total weight of the network. Let N be a network obtained from a complete bipartite graph $K_{m;m}$ with bipartition $\{X; Y\}$ such that $|X| = m$, $|Y| = m$, $V(N) = X \cup Y$ and M be a perfect matching for N.

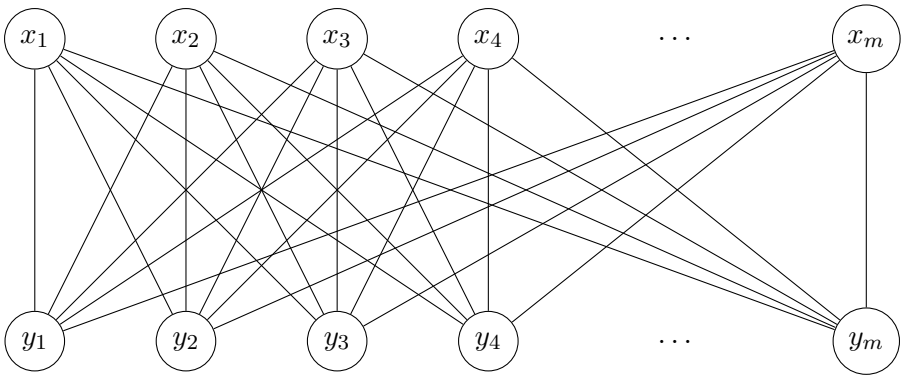

FIGURE 1. Network representation of $\mathrm{K}_{m;m}$

Definition 3.1. Define $f : \mathrm{V(N)} \to \mathbb{Z}$ such that $f(v)$ equal to minimum weight $w > 0$ among all the edges incident on v and $f(v) \leq w(e)$ for each $v \in \mathrm{V}$, where e is the edge in M incident on vertex v. The depth of X by f_X such that

$$f_X = \sum_{v \in X} f(v).$$

Definition 3.2. If $f_X \geq f_Y$, then the set of vertices X is said to be favorable in N where f_X is the depth of X.

Lemma 3.3. *Let* N *be a network obtained from a complete bipartite graph* $\mathrm{K}_{m;m}$ *with bipartition* $\{X;Y\}$ *such that* $|X| = m$, $|Y| = m$, $\mathrm{V(N)} = X \cup Y$ *and* M *be a perfect matching for* N. *If* S_X *is the set of edges generates by* f *over* X *is*

$$\mathrm{S}_X = \{e_i : e_i \text{ is an edge associated with } f(x_i), 1 \leq i \leq m\},$$

then

$$f_X = \sum_{v \in X} f(v) = \sum_{e \in \mathrm{S_X}} w(e) = w(\mathrm{S_X}).$$

Again if S_Y *is the set of edges generates by* f *over* Y *is*

$$\mathrm{S}_Y = \{e_j : e_j \text{ is an edge associated with } f(y_j), 1 \leq j \leq m\},$$

then

$$f_Y = \sum_{v \in Y} f(v) = \sum_{e \in \mathrm{S}_Y} w(e) = w(\mathrm{S}_Y).$$

Proof. As N is a network obtained from a complete bipartite graph $\mathrm{K}_{m;m}$ with bipartition $\{X;Y\}$ such that $|X| = m$, $|Y| = m$, $\mathrm{V(N)} = X \cup Y$ and M is a perfect matching for N. Each vertex of X is adjacent to all vertices of Y; let

$x_1 \in X$ be adjacent to all vertices of Y such as $x_1y_1, x_1y_2, \cdots, x_1y_m$ with each edge has an integer weight $w_{11}, w_{12}, \cdots, w_{1m}$ respectively. By definition of minimum weight function $f : \mathrm{V(N)} \to \mathbb{Z}$, $f(v)$ equal to minimum weight $w > 0$ among all the edges incident on v, implying $f(x_1) =$ minimum of $w_{1j},\ 1 \leq j \leq m$. The result is shown under the following two cases.

Case I: For the set X, if $f(x_1) =$ minimum of $w_{1j},\ 1 \leq j \leq m$ associated with exactly one edge out of $x_1y_1, x_1y_2, \cdots, x_1y_m$; let it be $x_1y_1 = e_1$(say). Similarly each other vertices of $x_i \in X,\ 2 \leq i \leq m$ will give exactly one edge $e_i,\ 2 \leq i \leq m$(say). Now all the edges $e_i,\ 1 \leq i \leq m$ form a set S_X having exactly m edges. Hence

$$f_X = \sum_{v \in X} f(v) = \sum_{e \in \mathrm{S_X}} w(e) = w(\mathrm{S_X}).$$

Case II: Similarly for the set Y, if $f(x_1) =$ minimum of $w_{1j},\ 1 \leq j \leq m$ associated with two edges out of $x_1y_1, x_1y_2, \cdots, x_1y_m$; let these be x_1y_1 and x_1y_2. Now we select $x_1y_1 = e_1$ (say) if total weight of y_1 is greater than or equal to that of y_2 that is $w(y_1) \geq w(y_2)$; else select $x_1y_2 = e_1$(say). Similarly each tie can be break for other vertices (if any) of $x_i \in X,\ 2 \leq i \leq m$ will give exactly one edge $e_i,\ 2 \leq i \leq m$(say). Now all the edges $e_i,\ 1 \leq i \leq m$ form a set S_X having exactly m edges. Hence

$$f_X = \sum_{v \in X} f(v) = \sum_{e \in \mathrm{S_X}} w(e) = w(\mathrm{S_X}).$$

Similarly for the set Y, it can be shown. □

Example 3.4. Let N be a network obtained from complete bipartite graph $\mathrm{K}_{5;5}$ with bipartition $X = \{x_1, x_2, x_3, x_4, x_5\}$ and $Y = \{y_1, y_2, y_3, y_4, y_5\}$ and giving each edge xy an integer weight $w(xy) > 0$ and minimum weight f_x and f_y for each vertex has been represented in the following matrix:

	y_1	y_2	y_3	y_4	y_5	f_x
x_1	6	12	3	11	15	3
x_2	4	2	7	1	10	1
x_3	8	11	10	7	11	7
x_4	16	19	12	23	21	12
x_5	9	5	7	6	10	5
f_y	4	2	3	1	10	

Let $\mathrm{M} = \{x_1y_1, x_2y_4, x_3y_5, x_4y_3, x_5y_2\}$ be a perfect matching of maximum weight in N. Then $w(\mathrm{M}) = 6 + 1 + 11 + 12 + 5 = 35$ is obtained by Hungarian

method. Now $f(x_1)$ equal to the minimum weight $w_{1j} > 0$, $1 \leq j \leq 5$ of an edge incident to x_1 and so on for other vertices.

$$\begin{aligned} f(x_1) = 3 &: \text{ contributes an edge } x_1y_3 \text{ to } \mathrm{S}_X, \\ f(x_2) = 1 &: \text{ contributes an edge } x_2y_4 \text{ to } \mathrm{S}_X, \\ f(x_3) = 7 &: \text{ contributes an edge } x_3y_4 \text{ to } \mathrm{S}_X, \\ f(x_4) = 12 &: \text{ contributes an edge } x_4y_3 \text{ to } \mathrm{S}_X, \\ f(x_5) = 5 &: \text{ contributes an edge } x_5y_2 \text{ to } \mathrm{S}_X. \end{aligned}$$

Now

$$\mathrm{S}_X = \{x_1y_3, x_2y_4, x_3y_4, x_4y_3, x_5y_2\}$$

is the set of edges generated by f over X. Therefore

$$f_X = \sum_{x \in X} f(x) = \sum_{\epsilon \in \mathrm{S}_X} w(\epsilon) = w(\mathrm{S}_X) = 28.$$

Similarly,

$$\begin{aligned} f(y_1) = 4 &: \text{ contributes an edge } y_1x_2 \text{ to } \mathrm{S}_Y, \\ f(y_2) = 2 &: \text{ contributes an edge } y_2x_2 \text{ to } \mathrm{S}_Y, \\ f(y_3) = 3 &: \text{ contributes an edge } y_3x_1 \text{ to } \mathrm{S}_Y, \\ f(y_4) = 1 &: \text{ contributes an edge } y_4x_2 \text{ to } \mathrm{S}_Y, \\ f(y_5) = 10 &: \text{ contribute edges } y_5x_2 \text{ or } y_5x_5 \text{ to } \mathrm{S}_Y, \end{aligned}$$

because $w(x_5) > w(x_2)$, select y_5x_5 to S_Y, where $w(x_5) = 37$; $w(x_2) = 24$, now

$$\mathrm{S}_Y = \{y_1x_2, y_2x_2, y_3x_1, y_4x_2, y_5x_5\}$$

is the set of edges generated by f over Y with $|S_Y| = 5$, therefore

$$f_Y = \sum_{y \in Y} f(y) = \sum_{\epsilon \in \mathrm{S}_Y} w(\epsilon) = w(\mathrm{S}_Y) = 20.$$

We have $f_X > f_Y$, so X is favorable set of vertices to generate favorable matching in the network. Now verify our definition; x_3 generates an edge $\epsilon = x_3y_4$ to S_X and the same vertex has an associate edge $e = x_3y_5$ in M that is $w(e) > w(\epsilon)$. Similarly, y_4 generates an edge $\epsilon = y_4x_2$ to S_y and the same vertex has an associate edge $e = y_4x_2$ in M, that is, $w(e) = w(\epsilon) = 1$ and similarly others can be obtained.

Lemma 3.5. *Let* N *be a network obtained from a complete bipartite graph* $\mathrm{K}_{m;m}$ *with bipartition* $\{X;Y\}$ *such that* $|X| = m$, $|Y| = m$, $\mathrm{V(N)} = X \cup Y$ *and* $f : \mathrm{V(N)} \to \mathbb{Z}$ *such that* $f(v)$ *equal to minimum weight* $w(\epsilon) > 0$ *of an edge* ϵ *incident on* v. *The function* f *generates a favorable matching* S_X *a set*

of edges obtained from a set of favorable vertices X *and* M *a perfect matching for* N. *Then* $w(\mathrm{M}) \geq w(\mathrm{S}_X)$, *where* $w(\mathrm{S}_X) = \sum_{\epsilon \in S_X} w(\epsilon)$.

Proof. Let N be a network obtained from a complete bipartite graph $\mathrm{K}_{m;m}$ with bipartition $\{X;Y\}$ such that $|X| = m$, $|Y| = m$, $\mathrm{V(N)} = X \cup Y$ and $f : \mathrm{V(N)} \to \mathbb{Z}$ such that $f(v)$ equal to minimum weight $w(\epsilon) > 0$ of an edge ϵ incident on v, and M be a perfect matching for N. In the network N each vertex of X is adjacent to every vertex of Y. Let X be the set of favorable vertices. Clearly from the figure 1, m edges such as $x_1y_1, x_1y_2, \cdots, x_1y_m$ with each edge has an integer weight $(w_{1j} > 0, 1 \leq j \leq m)$ are incident on vertex $x_1 \in X$, so $f(x_1) = \min(w_{1j},\ 1 \leq j \leq m)$. The weight represent a single homogeneous components like distance or cost or time. According to the Lemma 3.3 the function f will select an edge having minimum weight for each vertex on the favorable set of vertex to generate favorable matching. Therefore, each vertex $x \in X$ will contribute an edge ϵ to generate S_X by the function f with weight $w(\epsilon) = f(x)$. Since X has m vertices, so also the favorable matching S_X have m edges. Let M be a perfect matching for N. Now from the definition of favorable vertices, we have the following

$$w(\mathrm{M}) = \sum_{e \in \mathrm{M}} w(e) \geq \sum_{x \in X} f(x) = f_X = \sum_{\epsilon \in \mathrm{S}_X} w(\epsilon) = w(\mathrm{S}_X)$$

holds fairly because $f(x)$ equal to minimum weight $(w > 0)$ of an edge incident on x. □

Example 3.6. Using the result of Example 3.4 we can verify the Lemma 3.3; total weight of the perfect matching $w(\mathrm{M}) = 35$ and the total weight of the favorable matching $w(\mathrm{S}_X) = 28$ satisfy the statement of the Lemma 3.5, that is

$$w(\mathrm{M}) = \sum_{e \in \mathrm{M}} w(e) > \sum_{\epsilon \in \mathrm{S}_X} w(\epsilon) = w(\mathrm{S}_X).$$

Definition 3.7. A favorable matching S of N is said to be optimal; if the equality subgraph $\mathrm{G_S}$ for f in N is the spanning subgraph of N containing all edges ϵ for which $w(\epsilon) = w(e)$ for each $\epsilon \in \mathrm{S}$, where $e \in \mathrm{M}$ is a perfect matching in N.

Lemma 3.8. *Let* S *be a optimal favorable matching and* M *be a perfect matching for* N. *Then* $w(\mathrm{M}) = w(\mathrm{S})$ *and* S *is a maximum size favorable matching of* N *with* $|\mathrm{S}| = m$.

Proof. Let N be a network obtained from a complete bipartite graph $\mathrm{K}_{m;m}$ with bipartition $\{X;Y\}$ such that $|X| = m$, $|Y| = m$, $\mathrm{V(N)} = X \cup Y$ and

M be a perfect matching for N. Define $f : \mathrm{V(N)} \to \mathbb{Z}$ such that $f(v)$ equal to minimum weight $w(\epsilon) > 0$ of an edge ϵ incident on v. In the network N each vertex of X is adjacent to every vertex of Y. Let X be the set of favorable vertices. Clearly from figure 1, vertex $x_1 \in X$ has m edges such as $x_1y_1, x_1y_2, \cdots x_1y_m$ with each edge has an integer weight ($w_{1j} > 0, 1 \leq j \leq m$). The weight represent a single homogeneous components like distance or cost or time. The function f will select an edge having minimum weight to generate favorable matching. Therefore by Definition 3.7 each vertex $x \in X$ will contribute an edge ϵ to generate S by the function f with weight $w(\epsilon) = f(x)$. Since X has m vertices, so also the favorable matching S have m edges. Let M be a perfect matching for N. Now the equality subgraph of S is $\mathrm{G_S}$ is a spanning subgraph containing all edges ϵ such that $w(\epsilon) = w(e)$, where $e \in \mathrm{M}$. Thus we have

$$w(\mathrm{M}) = \sum_{e \in \mathrm{M}} w(e) = \sum_{\epsilon \in \mathrm{M}} w(\epsilon) = w(\mathrm{S}).$$

This established the statement. □

Example 3.9. Using the result of Example 3.4, we can obtained a equality subgraph $\mathrm{G_{S_X}}(f)$ for $\mathrm{S}_X = \{x_1y_3, x_2y_4, x_3y_4, x_4y_3, x_5y_2\}$.

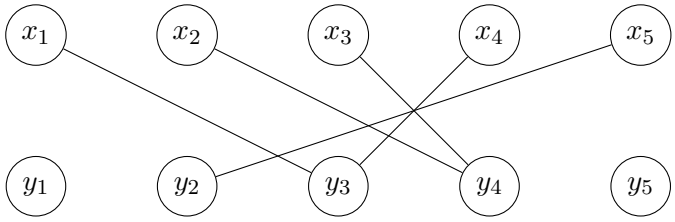

FIGURE 2. Equality subgraph $\mathrm{G_{S_X}}(f)$

Theorem 3.10. (*Extension of Egerváry Theorem*) *Let* N *be a weighted complete bipartite graph, the maximum weight of a perfect matching is* $w(\mathrm{M})$. *Then using proposed method to obtained an optimal favorable matching in* N *is* S_X *and* $w(\mathrm{S}_X) = w(\mathrm{M})$.

Proof. The proof is directly followed from the results of Lemma 3.3, Lemma 3.5 and Lemma 3.8]. □

3.1. **Computational Steps of Logical Approach.** Suppose N is a network obtained from $\mathrm{K}_{m;m}$ by giving each edge e an integer weight $w(e)$. The algorithm iteratively constructs a sequence of favorable matching S_{X_1}; $\mathrm{S}_{X_2}, \cdots, \mathrm{S}_{X_m}$ for N such that $w(\mathrm{S}_{X_{i+1}}) > w(\mathrm{S}_{X_i})$, and a sequence of matchings S_{X_i} such that S_{X_i} is a favorable matching in the equality subgraph $\mathrm{G}_{S_{X_i}}(f)$,

for all $1 \leq i \leq m$. It stops when it finds a optimal favorable vertex matching S_{X_i}.

Basis step:

- Compute δ_x for each $x \in X$, if set of vertices X is favorable; else compute δ_y for each $y \in Y$, where

$$\begin{aligned}\delta_i = \delta(x_i) &= \{\max_j w_{ij} - (\min_j w_{ij} + \text{next } \min_j w_{ij})\},\ 1 \leq i \leq m,\\ \delta_j = \delta(y_i) &= \{\max_i w_{ij} - (\min_i w_{ij} + \text{next } \min_i w_{ij})\},\ 1 \leq j \leq m.\end{aligned}$$

- Construct a new favorable matching

$$\mathrm{S}_{X^*} = \{e_i :\ e_i \text{ is associated with } \min_j w_{ij};\ 1 \leq i \leq m\}$$

  starting from the vertex having $\min \delta_x$ followed by next $x \in X$.
- Last two edges of S_{X^*} are obtained from the decision matrix of order 2 (Das et al. [1] Definition 2.12).
- For a tie on $\min \delta_x$ in the i^{th} vertex; preference should be given to that one having $\min w_{ij}$ for all j; but in case of tie on $\min w_{ij}$ for all j preference should be given to that $\min w_{ij}$ for all j having greater total weight of vertex, i.e., $w(y_j)$.

Recursive step: Suppose that we have constructed a favorable matching S_{X_i} of N with total weight $w(\mathrm{S}_{X_i})$; and maximum matching M in G for some $i \geq 1$.

- If $w(\mathrm{M}) \neq w(\mathrm{S}_{X_i})$, then construct a new favorable matching $\mathrm{S}_{X_{i+1}}$ for N as follows:
  in i^{th} vertex; set $\delta = \min_j w_{ij}$. Modify the δ_x for each $x \in X$ such that

$$\delta_x = \begin{cases} \delta_x + \delta, & \delta_x < 0;\\ \delta_x - \delta, & \delta_x > 0;\\ \delta_x, & \delta_x = 0. \end{cases} \tag{3.1}$$

- Construct favorable matching

$$\mathrm{S}_{X_{i+1}} = \{e_i :\ e_i \text{ associated with } \min_j w_{ij},\ 1 \leq i \leq m\}$$

  starting from the larger δ_x to small for each $x \in X$.
- if $w(\mathrm{M}) = w(\mathrm{S}_{X_{i+1}})$, or $i = m$; then stop and out put $\mathrm{S}_{X_{i+1}}$ and $w(\mathrm{S}_{X_{i+1}})$; else iterate.

It is to be noticed that

(a) the method must terminate since each iteration decreases the number of vertices, and depth of favorable matching is bounded above by the weight of perfect matching of N.

(b) when the algorithm terminates it outputs a optimal favorable matching S_{X_i} and a perfect matching M_i in the equality subgraph $\mathrm{G}_{\mathrm{S_X}}(f)$ such that $w(\mathrm{M}_i) = w(\mathrm{S}_{X_i})$.

(*c*) The complete bipartite network with unequal vertex partition $K_{m;n}$, i.e., $|X| = m \neq |Y| = n$, can be modified by adding dummy vertices with each edge having weight $w(xy) = 0$.
(*d*) Maximization problem can be solved by converting to minimization problem.

4. Examples

Some numerical examples on assignment problem are studied to illustrate the process of calculation for the proposed logical approach.

Example 4.1. Let N be a network obtained from $K_{3;3}$ with bipartition $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2, y_3\}$ and giving each edge xy an integer weight $w(xy) > 0$ in the following matrix:

	y_1	y_2	y_3	f_x
x_1	250	400	350	250
x_2	400	600	350	350
x_3	200	400	250	200
f_y	200	400	250	

Here $f_X = 250 + 350 + 200 = 800$ and $f_Y = 200 + 400 + 250 = 850$. Since $f_Y > f_X$, so the vertices of set Y is favorable for matching. Let $M = \{x_1y_2, x_2y_3, x_3y_1\}$ be a perfect matching of maximum weight in N is $w(M) = 400 + 350 + 200 = 950$ by Hungarian method. We apply the logical approach as follows:

Step 1: Compute δ_y for each $y \in Y$ and represent all in the following matrix

	y_1	y_2	y_3
x_1	250	400	350
x_2	400	600	350
x_3	200	400	250
δ_y	-50	-200	-250

Step 2: We start to construct the favorable matching from the vertex y having $\min \delta_y$ for all $y \in Y$, that is y_3 having $\delta_y = -250$.

- First matching: here we choose minimum weight of an edge associated with the vertex y_3, that is, $\min\{w(x_1y_3), w(x_2y_3), w(x_3y_3)\} = w(x_3y_3) = 250$ and remove the corresponding vertices x_3 and y_3 from N.
- Second and third matching: By Das et al. ([1]), here the second and third matching can be done by using the corresponding (2×2)

decision matrix

$$D = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 250 & 400 \\ 400 & 600 \end{bmatrix} \end{array}$$

Clearly, $\mathrm{tr}\,(D) = 850 > \mathrm{offtr}\,(D) = 800$; so off-diagonal entries are the required choice for favorable matching i.e., x_1y_2 and x_2y_1.

We have the favorable matching $\mathrm{S}_{Y_1} = \{x_3y_3, x_1y_2, x_2y_1\}$ having $w(\mathrm{S}_{Y_1}) = 250 + 400 + 400 = 1050$.

Step 3: Since maximum matching is $w(\mathrm{M}) < w(\mathrm{S}_{Y_1})$, the current favorable matching S_{Y_1} is not optimal. Now we have to modify (see (3.1)) each δ_y by adding $\delta = 250 = \min \delta_{y_3}$; for all $x \in X$ and values has been represented in the following matrix:

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ \delta_y \end{array} \begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 250 & 400 & 350 \\ 400 & 600 & 350 \\ 200 & 400 & 250 \end{bmatrix} \\ \begin{array}{ccc} 200 & 50 & 0 \end{array} \end{array}$$

Next we modify the matching from the vertex y having $\max \delta_y$ for all $y \in Y$ that is y_1 having $\delta_y = 200$.

- First matching: Here we choose minimum weight of an edge associated with the vertex y_1, that is, $\min\{w(x_1y_1), w(x_2y_1), w(x_3y_1)\} = w(x_3y_1) = 200$ and remove the corresponding vertices x_3 and y_1 from N.
- Second and third matching: By Das et al. ([1]), here the second and third matching can be done by using the corresponding (2×2) decision matrix:

$$D = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{cc} y_2 & y_3 \end{array} \\ \begin{bmatrix} 400 & 350 \\ 600 & 350 \end{bmatrix} \end{array}$$

Since $\mathrm{tr}\,(D) = 750 < \mathrm{offtr}\,(D) = 950$; the diagonal entries are the required choice for favorable matching, i.e., x_1y_2 and x_2y_3.

Now we have the favorable matching $\mathrm{S}_{Y_1} = \{x_3y_1, x_1y_2, x_2y_3\}$ having $w(\mathrm{S}_{Y_2}) = 200 + 400 + 350 = 950$ and $|\mathrm{S}_{Y_2}| = 3$.

Step 4: Clearly, maximum matching is $w(\mathrm{M}) = w(\mathrm{S}_{Y_2}) = 950$. So the current favorable matching S_{Y_2} is optimal.

Example 4.2. Let N be a network obtained from $K_{4;4}$ with bipartition $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3, y_4\}$ and giving each edge xy an integer weight $w(xy) > 0$ in the following matrix:

	y_1	y_2	y_3	y_4	f_x
x_1	1	4	6	3	1
x_2	8	7	10	9	7
x_3	4	5	11	7	4
x_4	6	7	8	5	5
f_y	1	4	6	3	

Here $f_X = 1+7+4+5 = 17$ and $f_Y = 1+4+6+3 = 14$. Since $f_X > f_Y$, so the vertices of set X is favorable for matching. Let $M = \{x_1y_1, x_2y_3, x_3y_2, x_4y_4\}$ be a perfect matching of maximum weight in N is $w(M) = 1+10+5+5 = 21$ by Hungarian method. Now we apply the logical approach as follows:

Step 1: Compute δ_x for each $x \in X$ and represent all in the following matrix

	y_1	y_2	y_3	y_4	δ_x
x_1	1	4	6	3	2
x_2	8	7	10	9	-5
x_3	4	5	11	7	2
x_4	6	7	8	5	-3

Step 2: We construct our favorable matching from the vertex x having $\min \delta_x$ for all $x \in X$, that is x_2 having $\delta_x = -5$.

- First matching: Here we choose minimum weight of an edge associated with the vertex x_2, that is,
  $\min\{w(x_2y_1), w(x_2y_2), w(x_2y_3), w(x_2y_4)\} = w(x_2y_2) = 7$
  and remove the corresponding vertices x_2 and y_2 from N.
- Second matching: Here we choose minimum weight of an edge associated with the next vertex x_3, that is,
  $\min\{w(x_3y_1), w(x_3y_3), w(x_3y_4)\} = w(x_3y_1) = 4$
  and remove the corresponding vertices x_3 and y_1 from N.
- Third and fourth matching: By Das et al. ([1]) here the third and fourth matching can be done by using the corresponding decision square matrix

$$D = \begin{array}{c} \\ x_1 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cc} y_3 & y_4 \end{array} \\ \begin{bmatrix} 6 & 3 \\ 8 & 5 \end{bmatrix} \end{array}$$

Now clearly, $\mathrm{tr}\,(D) = 11 \leq \mathrm{offtr}\,(D) = 11$; so diagonal entries are the required choice for favorable matching i.e., x_1y_3 and x_4y_4.

Now we have the favorable matching $\mathrm{S}_{X_1} = \{x_2y_2, x_3y_1, x_4y_4, x_1y_3\}$ having $w(\mathrm{S}_{X_1}) = 7 + 4 + 5 + 6 = 22$.

Step 3: Since maximum matching $w(\mathrm{M}) \neq w(\mathrm{S}_{X_1})$, the current favorable matching S_{X_1} is not optimal. Now we have to modify (see (3.1)) each δ_x by adding and subtracting $\delta = 7 = \delta_{x_2}$; *for* $y \in Y$ and values has been represented in the following matrix

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 1 & 4 & 6 & 3 \\ 8 & 7 & 10 & 9 \\ 4 & 5 & 11 & 7 \\ 6 & 7 & 8 & 5 \end{bmatrix} \end{array} \begin{array}{c} \delta_x \\ -5 \\ 2 \\ -5 \\ 4 \end{array}$$

Now we modify favorable matching from the vertex x having $\max \delta_x$ for all $x \in X$, that is, x_4 having $\delta_x = 4$.

- First matching: here we choose minimum weight of an edge associated with the vertex x_4, that is,
  $\min\{w(x_4y_1), w(x_4y_2), w(x_4y_3), w(x_4y_4)\} = w(x_4y_4) = 5$
  and remove the corresponding vertices x_4 and y_5 from N.
- Second matching: here we choose minimum weight of an edge associated with the next vertex x_1, that is,
  $\min\{w(x_1y_1), w(x_1y_2), w(x_1y_3)\} = w(x_1y_1) = 1$
  and remove the corresponding vertices x_1 and y_1 from N.
- Third and fourth matching: By Das et al. ([1]) here the third and fourth matching can be done by using the corresponding (2×2) decision matrix

$$\begin{array}{c} \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_2 & y_3 \end{array} \\ \begin{bmatrix} 7 & 10 \\ 5 & 11 \end{bmatrix} \end{array}$$

  Now clearly, $\mathrm{tr}\,(D) = 18 > \mathrm{offtr}\,(D) = 15$; so off-diagonal entries are the required choice for favorable matching, that is, x_2y_3 and x_3y_2.

Now we have the favorable matching
$\mathrm{S}_{X_2} = \{x_4y_4, x_1y_1, x_2y_3, x_3y_2\}$
having $w(\mathrm{S}_{X_2}) = 5 + 1 + 10 + 5 = 21$ and $|\mathrm{S}_{X_2}| = 4$.

Step 4: Clearly, maximum matching $w(\mathrm{M}) = w(\mathrm{S}_{X_2})$. So the current favorable matching S_{X_2} is optimal.

Example 4.3. Let N be a network obtained from $K_{5;5}$ with bipartition $X = \{x_1, x_2, x_3, x_4, x_5\}$ and $Y = \{y_1, y_2, y_3, y_4, y_5\}$ and giving each edge xy an integer weight $w(xy) \geq 0$ in the following matrix:

	y_1	y_2	y_3	y_4	y_5	f_x
x_1	50	40	60	20	0	20
x_2	40	30	40	30	0	30
x_3	60	20	30	20	0	20
x_4	30	30	20	30	0	20
x_5	10	20	10	30	0	10
f_y	10	20	10	20	0	

Here $f_X = 20+30+20+20+10 = 100$ and $f_Y = 10+20+10+20 = 60$. Since $f_X > f_Y$, so the vertices of set X is favorable for matching. y_5 is the dummy vertex added to the network N with all the weights $w(xy) = 0$ for all $x \in X$ for our need because initially the network is complete bipartite graph with unequal partition. $M = \{x_1y_4, x_2y_5, x_3y_2, x_4y_3, x_5y_1\}$ be a perfect matching of maximum weight in N is $w(M) = 20 + 0 + 20 + 20 + 10 = 70$ by Hungarian method. Now we apply the logical approach as follows:

Step 1: Compute δ_x for each $x \in X$ represent all in the following matrix

	y_1	y_2	y_3	y_4	y_5	δ_x
x_1	50	40	60	20	0	0
x_2	40	30	40	30	0	-20
x_3	60	20	30	20	0	20
x_4	30	30	20	30	0	-20
x_5	10	20	10	30	0	10

Step 2: Let us start to construct our favorable matching from the vertex x_2 and x_4 having $\min \delta_x = -20$ for all $x \in X$. This is a tie on -20.

- First matching: here we choose minimum weight of an edge associated with the vertex x_2, that is,

  $$\min\{w(x_2y_1), w(x_2y_2), w(x_2y_3), w(x_2y_4), w(x_2y_5)\}$$
  $$= w(x_2y_5) = 0$$

  because it is associated with larger row sum, i.e.,$w(x_2) > w(x_4)$ and remove the corresponding vertices x_2 and y_5 from N.
- Second matching: here we choose minimum weight of an edge associated with the next vertex x_3, that is,
  $\min\{w(x_3y_1), w(x_3y_2), w(x_3y_3), w(x_3y_4)\} = w(x_3y_2) = 20$
  because it is associated with maximum column sum,i.e., $w(y_2) > w(y_4)$ and remove the corresponding vertices x_3 and y_2 from N.

- Third matching: here we choose minimum weight of an edge associated with the next vertex x_4, that is,
  $\min\{w(x_4y_1), w(x_4y_3), w(x_4y_4)\} = w(x_4y_3) = 20$
  and remove the corresponding vertices x_4 and y_3 from N.
- Fourth and fifth matching: By `Das` [1] here the fourth and fifth matching can be done by using the corresponding (2×2) decision matrix

$$D = \begin{array}{c} \\ x_1 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_4 \end{array} \\ \begin{bmatrix} 50 & 20 \\ 10 & 30 \end{bmatrix} \end{array}$$

  Now clearly, $\mathrm{tr}\,(D) = 80 > \mathrm{offtr}\,(D) = 30$; so off-diagonal entries are the required choice for favorable matching i.e., x_1y_4 and x_5y_1.

Now we have the favorable matching $\mathrm{S}_X = \{x_2y_5, x_3y_2, x_4y_3, x_5y_1, x_1y_4\}$ having $w(\mathrm{S}_X) = 20 + 0 + 20 + 20 + 10 = 70$ and $|\mathrm{S}_X| = 5$.

Step 3: Clearly, maximum matching $w(\mathrm{M}) = w(\mathrm{S}_X)$. So the current favorable matching S_X is optimal.

5. Algorithm complexity

In this section our interest is in the efficiency of the algorithm (time complexity). The complexity of an algorithm is simply the number of computational steps that it takes to transform the input data to the result of a computation. Now we mainly focus on our proposed algorithm and for this purpose, we have the following results.

Lemma 5.1. *Let* N *be a network obtained from a complete bipartite graph by giving each edge an integer weight. Suppose we use the proposed logical approach to construct a maximum weight perfect matching in* N. *Then the number of times the method grows an alternating favorable matching is at most* $\frac{1}{2}|\mathrm{V(N)}|$, *where* $|\mathrm{V(N)}|$ *is cardinality of vertex set in* N.

Proof. Let N is a network obtained from $\mathrm{K}_{m;m}$ by giving each edge e an integer weight $w(e)$ with cardinality of vertex set $|\mathrm{V(N)}| = 2m$. The algorithm proceeds by selecting a $\min \delta_x$ from each $x \in X$. Now for an $x \in X$ associate with $\min \delta_x$ have exactly one favorable matching. Each $x \in X$ will grow an alternative favorable matching S_{X_i}, for any $1 \leq i \leq m$. Out of these m alternative favorable matching at least one must coincides with the maximum matching M of equality spanning subgraph of N. Thus the proposed logical approach can have at most $m = \frac{1}{2}|\mathrm{V(N)}|$, alternating favorable matching where $|\mathrm{V(N)}|$ is cardinality of vertex set in N. □

Theorem 5.2. *Suppose* N *is a network obtained from a complete bipartite graph by giving each edge an integer weight. Then the proposed method finds a maximum weight favorable matching in* N *in time* $O(|\mathrm{V(N)}|^2)$, *where* $|\mathrm{V(N)}|$ *is cardinality of vertex set in* N; *under the assumption that all elementary arithmetic operations take constant time.*

Proof. Let N be a network obtained from $\mathrm{K}_{m;m}$ by giving each edge e an integer weight $w(e)$ with cardinality of vertex set $|\mathrm{V(N)}| = 2m$. The algorithm proceeds by growing alternating favorable matching S_{X_i} for $1 \leq i \leq m$. Growing an alternating favorable matching in an equality subgraph $\mathrm{G}_{\mathrm{S_X}}(\mathrm{f})$ by breadth first search takes $10m$ time. Now growing an m-alternating favorable matching in an equality subgraph $\mathrm{G_{SX}(f)}$ takes $10m \times m = 10m^2$ time. Thus the total time spent on alternating favorable matching is $O(10m^2) = O(m^2) = O(|\mathrm{V(N)}|^2)$. □

6. Result analysis

In this section the results obtained by logical approach are compared with results obtained by other existing methods with their optimal solutions. The following tables Table 6.1 and Table 6.2 summarize all the results of Example 4.1, Example 4.2 and Example 4.3.

	Methods		
Examples	Logical Approach	Hungarian Method	DAST
Ex. 4.1	950	950	950
Ex. 4.2	21	21	21
Ex. 4.3	70	70	70

Table 1. Optimal Solution $w(\mathrm{S}_X)$ or $w(\mathrm{M})$

	Methods		
Examples	Logical Approach	Hungarian Method	DAST
Ex. 4.1	27	27	19
Ex. 4.2	36	64	29
Ex. 4.3	81	162	36

Table 2. Number of Steps to find optimal solution $w(\mathrm{S}_X)$ or $w(\mathrm{M})$

The optimal favorable matching of a network N by proposed logical method and Hungarian method are coincide with the same numerical value. The time complexity of proposed logical method is fairly less than as compare to complexity of Hungarian method. Here total number of algebraic calculations

needed to convert the input data to the optimal solution is multiple of n^2, that is, $O(n^2)$ under the assumption that all algebraic calculations can take equal time.

7. Conclusions

A large number of real world problems can be modeled as an assignment problem because of its combinatorial nature. Till date several methods and algorithms has been develop to solve the assignment problem. But, it is very important to choose the perfect method or approach to deal the problem, to an obtained optimal solution or closer to optimal solution depending on the nature of complexity of the problem. In recent trends some approaches are top choice for the solution of an assignment problem because they produce good but not certainly optimal solution. In this context our proposed method produce good as well as optimal solution in reasonable short amount of time.

References

[1] P.K. Das and S.K. Das, *Dominated Assignment Simulation Technique*, J. Orissa Math. Soc., **33**(2) (2014), 71–108.

[2] B. Jackson, *Graph Theory and Application*, Qeen Mary University of London, UK. `http//www.Maths.qmul.ac.uk/ bill/MAS210/ch6.pdf`.

[3] H.W. Khun, *The Hungarian Method for the Assignment Problem*, Naval Research Logistics Quarterly, **2** (1955), 83–97.

[4] J. Munkres, *Algorithms for the Assignment and Transportation Problems*, J. Soc. for Indust. and Appl. Math., **5**(1) (1957 March), 32–38.

[5] S.K. Mohanta, *An Optimal Solution for Transportation Problems: Direct Approach*, Project Submited to 35^{th} Orientation Programme from 31-01-2018 to 27-02-2018. UGC-HRDC, Sambalpur University, Burla, Odisha-768019, India, (2018), 1–22. `DOI:10.13140/RG.2.2.19271.34720`.