*Iosr Journal Of Economics And Finance (Iosr-Jef)*
*E-Issn: 2321-5933, P-Issn: 2321-5925.Volume 15, Issue 5 Ser. 6 (Sept. – Oct. 2024), Pp 63-88*
*www.Iosrjournals.Org*

# Reducing The Order Of The Travelling Salesman Problems By Minimin Optimization Theory And To Study By DAS Technique

Prasanta Kumar Das
*School Of Applied Sciences (Mathematics) KIIT University, BhubaneswarOdisha, India, 751024*

***Abstract:***
*The purpose of this article is to propose a new approach for finding the guaranteed solution set of minimized assignment problems and maximized assignment problems. Firstly the existence of the minimin and maximax optimization problems are studied with the help of newly defined weakly φ-convex function in φ-convex set. Next the assignment problems and traveling problems are converted to a complete bipartite graph in a unified approach. The concept dominated assignment simulation (DAS) technique based on the theory of minimin and maximax optimization problems.*
*Later a pair of traveling salesman problems are studied using the DAS technique and complete bipartite graph as an application of minimin optimization problem. Finally, a pair of comparison studies are discussed to show equality of solutions and the number of steps in between Hungarian method and DAS technique.*
***Keywords:****MiniMin and Maximax optimization problems, Minimized and Maximized Assignment problems, Dominated Assignment Simulation Technique, Traveling salesman problem.*

---

Date of Submission: 05-10-2024 Date of Acceptance: 15-10-2024

---

## 1. Introduction

The theory of assignment problems ($AP$) is one of the best tool to find the solution of various problems like assigning jobs to workers, workers to machines, Salesman to different sales area problems, Chinese Post man Problems, Network Flow Problems, Classes to rooms problems, Drivers to trucks, Vehicle to delivery routes, Contract to bidders, Pairing of crew with flight schedules, Problems to research terms and many more. Carl Gustav Jacobi introduced the concept of AP in 1890 and developed the algorithm of it. In 1931, the two Hungarian Mathematicians D. König and E. Egervàry have developed the algorithm to solve the AP Assignment Problem. AP was firstly seen in the article of Votaw and Oden in 1952. Later Kuhn ([16], 1955) gave the recognition and significance of the algorithm developed by two Hungarian Mathematicians D. König and E. Egervàry and coined the term of the algorithm "Hungarian method". Now a days "Hungarian Method" is one of the most useful method to solve linear assignment problem. The development of the AP is done due to various researchers such as Burkard and Cela [4], Bukard [5], Flood [11], Ford and Fulkerson [12], Kuhn [16], Kuhn [17], Levit and Mandrescu [18], Deming [9], Janson [14], Amponsah et al. [1], Dimitri [10], Rao and Srinivas [23] and the references therein. Although the basic version of the $AP$ can be solved very efficiently (say by the Hungarian method in O($n^3$) steps [21]), there are certain variants of this problem which are much harder, some being $NP$-complete or with undecided computational complexity. One of them is the parity $AP$: Obviously, $n$ entries of an $n \times n$ matrix, no two belonging to the same row or column, correspond to a permutation of the set $N = \{1,2,\cdots,n\}$. In the classical $AP$, no additional conditions are set on the optimal permutation. In the parity $AP$, this permutation has to be of a prescribed parity. In 2003, Butkovic [6] has shown that a diagonally dominant matrix can be transformed to a normal form by adding constants to the rows and/or columns and no permutations of the rows or columns are needed. These constants can be found in a straightforward way, without using the Hungarian method or other method for solving the $AP$. In 2018, Porchelvi and Anitha [22] have studied the assignment problem using average total opportunity cost method. According to Dantzig et al. [8], Hassler Whitney introduced the travelling salesman problem (TSP) in in his talk at Princeton University. It is to be noted that if the additional requirement is that the permutation is cyclic, then the arising task is the well-known (*NP*-complete) travelling salesman problem.

## 1.1 ASSIGNMENT PROBLEM

The assignment problem in which $n$ workers are assigned to $n$ jobs can be represented as an *LP* model: Define $a_{ij}$ is the cost of assigning worker $i$ to job $j$ $(i, j = 1,2,\cdots,n)$ which are given in the Table 1.1:

| Job→<br>Worker↓ | $B_1$ | $B_2$ | ⋯ | $B_n$ |
|---|---|---|---|---|
| $A_1$ | $a_{11}$ | $a_{12}$ | ⋯ | $a_{1n}$ |
| $A_2$ | $a_{21}$ | $a_{22}$ | ⋯ | $a_{2n}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $A_n$ | $a_{n1}$ | $a_{n2}$ | ⋯ | $a_{nn}$ |

Table 1.1: Assignment Problem.

and define

$$x_{ij} = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ job is assigned to } j^{\text{th}} \text{ person}\,; \\ 0 & \text{otherwise}\,. \end{cases}$$

Then, the *LP* model for assignment problem is defined by

Minimize

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij}$$

Subject to

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

$$\text{where} \quad x_{ij} = 0 \text{ or } 1 \;\; for\; i, j = 1,2,\cdots,n.$$

In second section, the results on minimin and maximax optimization problems are studied in $\phi$-convex set with the help of newly defined weakly $\phi$-convex function on it. Das technique is based on the theory of minimin optimization problems. In third section, the linear minimized assignment problem (or linear maximized assignment problem) is reduced to a bipartiate graph and studied with the help of DAS-technique associated with a decision matrix and the corresponding dominated column (or dominating column) for the undecided computational complexity. Some numerical examples are also given. The process to reduce the order of the minimized assignment problem is discussed as an application of the theory of minimin optimization problems. Algorithm and complete bipartiate graph for solving the traveling salesman problem in DAS-technique is discussed with some numerical examples. Finally in sixth section, a pair of comparison study are done to show equality of solutions and number of steps results in between Hungarian method and DAS technique.

# 2 MiniMin and MaxiMax Problems

In this section the existence of the solution of minimin and maximax optimization problems are studied in $\phi$-convex set with the help of newly defined weakly $\phi$-convex function on it. The results are useful to study the minimized and maximized assignment problems.

## Minimized Assignment Problem

Let $X$ and $Y$ be two vector spaces. Let $A \subset X$ and $B \subset Y$ be two $n$-dimensional convex subsets. Consider a network game with $n$ agents interacting over a network $w \in A \times B$ such that for $x_i \in A$ and $y_j \in B$, $w(x_i, y_j)$ is identified by

$$f(x) = \begin{cases} w_{ij} \geq 0, \text{ influence of } i \text{ on } j; \\ w_{ij} = 0, \text{ no self loops.} \end{cases}$$

Each agent $i$ has strategy $x_i \in A$ in the feasible set $X_i \subset A$ for the agent $j$. The cost of each agent $i$ to get the response from agent $j$ is

$$w_{ij} = f(x_i, y_j(x)): A \times B \to \mathbb{R}$$

where $y_j(x)$ is the aggregator of each agent $j$ on agent $i$ is defined by the rule

$$y_j(x) = \sum_{i=1}^{n} w_{ij} x_i.$$

The aim of the problem is to find the best response $(BR)$,

$$BR(x_i(y_j)) = \underset{x_i \in X_i}{\operatorname{argmin}} f(x_i, y_j(x)).$$

## 2.1 The Minimin and Maximax Problems

Let $A$ and $B$ be two subsets. Let $f: A \times B \to \mathbb{R}$ be a real valued function. The min-min problem and max-max problems are defined as follows:

$(P_1)$ The minimin problem is to find the optimal solution

$$\min_{x \in A} \min_{y \in B} f(x, y) = \min_{y \in B} \min_{x \in A} f(x, y).$$

$(P_2)$ The maximax problem is to find the optimal solution

$$\max_{x \in A} \max_{y \in B} f(x, y) = \max_{y \in B} \max_{x \in A} f(x, y).$$

Consider the set of solutions of the problems $P_1$ and $P_2$ as follows:

$$S(P_1) = \{(x^*, y^*) \in A \times B: (x^*, y^*) \text{ solves } P_1\}$$

$$S(P_2) = \{(x^*, y^*) \in A \times B: (x^*, y^*) \text{ solves } P_2\}.$$

If the function $f$ is differentiable on $A \times B$, then the Kuhn-Turker (KT) condition for both the problems $P_1$ and $P_2$ is to find the solution $u^* = (x^*, y^*) \in A \times B$ such that

$$\nabla f(u^*) = \begin{pmatrix} \nabla f_x(x^*, y^*) \\ \nabla f_y(x^*, y^*) \end{pmatrix} = 0.$$

## 2.2 Weakly $\phi$-Convex Functions and The results

The definition of weakly $\phi$-convex function in $\phi$-convex set is defined as follows.

**Definition 2.1** *Let $K$ be a nonempty subset of a vector space $X$.*

(i) A set $K \subset X$ is said to be $\phi$-convex on $K$ if there exist a function $\phi: \mathbb{R} = (0, \infty) \to [0,1]$ such that $\phi(\alpha) \to 0$ as $\alpha \to 0$ and $\phi(\alpha)x + (1 - \phi(\alpha))y \in K$ for all $x, y \in K$. If particular if $\phi(\alpha) = \alpha$, then $K$ is affine set.

(ii) The mapping $f: K \subset X \to Y$ is weakly $\phi$-convex on $K$ if for all $x, y \in K$,

$$f(\phi(\alpha)x + (1 - \phi(\alpha))y) \le \frac{\phi(\alpha)}{1 + \phi(\alpha)} f(x) + \frac{1}{1 + \phi(\alpha)} f(y)$$

**Theorem 2.2** *Let $K$ be a nonempty $\phi$-convex subset of a vector space $X$ and $f: K \to Y$ be weakly $\phi$-convex on $K$, then for any $y \in K$,*

$$f(x) - f(y) \ge 0 \text{ for all } x \in K \Leftrightarrow \langle \nabla f(y), x - y \rangle \ge 0 \text{ for all } x \in K.$$

***Proof***. It can be easily prove that

$$f(x) - f(y) \ge 0 \text{ for all } x \in K \Leftrightarrow \langle \nabla f(y), x - y \rangle \ge 0 \text{ for all } x \in K.$$

Conversely, let for any $y \in K$, $\langle \nabla f(y), x - y \rangle \ge 0$ for all $x \in K$. From the definition of weakly $\phi$-convexity of $f$ on $K$, we obtain

$$(1 + \phi(\alpha)) f(\phi(\alpha)x + (1 - \phi(\alpha))y) \le \phi(\alpha) f(x) + f(y)$$

for all $x, y \in K$. For any $y \in K$,

$$\begin{aligned}(1 + \phi(\alpha))[f(\phi(\alpha)x + (1 - \phi(\alpha))y) - f(y)] &\le \phi(\alpha) f(x) + f(y) - (1 + \phi(\alpha)) f(y) \\ &= \phi(\alpha)[f(x) - f(y)]\end{aligned}$$

for all $x \in K$, i.e.,

$$(1 + \phi(\alpha)) \frac{f(y + \phi(\alpha)(x - y)) - f(y)}{\phi(\alpha)} \le f(x) - f(y)$$

for all $x \in K$. Take limit as $\phi(\alpha) \to 0$ to obtain

$$f(x) - f(y) \ge \langle \nabla f(y), x - y \rangle \ge 0$$

for all $x \in K$. This completes the proof.

Let $g(\cdot) = f(\cdot, y): A \to \mathbb{R}$ and $h(\cdot) = f(x, \cdot): B \to \mathbb{R}$. The Minimin problem $P_1$ can be splitted in two problems:

($P_1^*$) find $y^* \in B$ such that for all $x \in A$,

$$f(x, y) - f(x, y^*) \ge 0 \; \forall \; y \in B \Leftrightarrow \langle \nabla f_y(x, y^*), y - y^* \rangle \ge 0 \; \forall \; y \in B;$$

($P_1^{**}$) find $y^* \in B$ is the minimum point of $h(y) = f(x, y)$, then find the point $x^* \in A$ such that for any $y^* \in B$,

$$f(x, y^*) - f(x^*, y^*) \ge 0 \; \forall \; x \in A \Leftrightarrow \langle \nabla f_x(x^*, y^*), x - x^* \rangle \ge 0 \; \forall \; x \in A.$$

The Maximax problem $P_2$ can be split in to two problems:

($P_2^*$) find $y^* \in B$ such that for all $x \in A$,

$$f(x,y) - f(x,y^*) \leq 0 \; \forall \; y \in B \Leftrightarrow \langle \nabla f_y(x,y^*), y - y^* \rangle \leq 0 \; \forall \; y \in B;$$

$(P_2^{**})$ find $y^* \in B$ is the maximum point of $h(y) = f(x,y)$, then find the point $x^* \in A$

such that for any $y^* \in B$,

$$f(x,y^*) - f(x^*,y^*) \leq 0 \; \forall \; x \in A \Leftrightarrow \langle \nabla f_x(x^*,y^*), x - x^* \rangle \leq 0 \; \forall \; x \in A.$$

**Theorem 2.3** *If $h(y) = f(x,y)$ is weakly $\phi_1$-convex on $B$ and $g(x) = f(x,y)$ is weakly $\phi_2$-convex on $A$, then the problem finding $u^* = (x^*,y^*) \in A \times B$ such that for all $u = (x,y) \in A \times B$,*

$$f(u) - f(u^*) \geq 0 \;\Leftrightarrow\; \langle \nabla f(u^*), u - u^* \rangle \geq 0$$

where

$$\nabla f(u^*) = \begin{pmatrix} \nabla f_x(x^*,y^*) \\ \nabla f_y(x^*,y^*) \end{pmatrix}.$$

***Proof.*** Proof of this theorem can be proved like Theorem 2.2. So the proof is skipped.

**Theorem 2.4** *If $(-h)(y) = (-f)(x,y)$ is weakly $\phi_1$-convex on $B$ and $(-g)(x) = (-f)(x,y)$ is $\phi_2$-convex on $A$, then the problem finding $u^* = (x^*,y^*) \in \Omega$ such that for all $u = (x,y) \in \Omega$,*

$$f(u) - f(u^*) \leq 0 \;\Leftrightarrow\; \langle \nabla f(u^*), u - u^* \rangle \leq 0$$

where

$$\nabla f(u^*) = \begin{pmatrix} \nabla f_x(x^*,y^*) \\ \nabla f_y(x^*,y^*) \end{pmatrix}.$$

***Proof.*** Proof of this theorem can be proved like Theorem 2.3. So the proof is skipped.

## 3 Network representation of AP

Harold Kuhn [15] developed and published the Hungarian method in 1955; is a combinatorial optimization algorithm that solves the assignment problem in polynomial time. James Munkres [20] 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 need. Let G be a graph having E(G) as a set of edges and V(G) is a set of vertices.

**Definition 3.1** [14] *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 3.2** [14] *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 $Y$.*

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

**Theorem 3.3** [14] Suppose N is a network obtained from $K_{n;n}$ by giving each edge $e$ an integer weight. Then the Hungarian method finds a maximum weight perfect matching in N in time $O(n^4)$, under the assumption that all elementary arithmetic operations take constant time.

### 3.1 Simulation Technique

Let $N$ be a network obtained from a complete bipartite graph $K_{n;n}$ with bipartition $\{X; Y\}$ such that $|X| = n$, $|Y| = n$, $V(N) = X \cup Y$ and M be a perfect matching for N. Here the Table 3.2 represents the cost matrix the assignment problem.

| Y→ X↓ | $y_1$ | $y_2$ | ⋯ | $y_n$ |
|---|---|---|---|---|
| $x_1$ | $w_{11}$ | $w_{12}$ | ⋯ | $w_{1n}$ |
| $x_2$ | $w_{21}$ | $w_{22}$ | ⋯ | $w_{2n}$ |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| $x_n$ | $w_{n1}$ | $w_{n2}$ | ⋯ | $w_{nn}$ |

Table 3.2: Cost Matrix.

and define

$$x_{ij} = \begin{cases} 1 & if\ x_i \text{ is assigned to } y_j\,; \\ 0 & \text{otherwise}\,. \end{cases}$$

The graphical form of the network N is given in Figure 3.1. In the network N, each vertex $x \in X$ is adjacent to all vertex $y \in Y$ and represented by $xy$. Let weight function

$$w: X \times Y \to W = \{w_{xy}: w_{x,y} = \text{the weight of the edge } xy\}.$$

The minimum weight function is

$$f_*: X \times Y \to W = \{f_*(x,y): f_{x,y} = \text{the minimum weight of the edge } xy\},$$

i.e.,

$$f_*(x_i, y_j) = \min_{1\le j\le n} x_i y_j = \min_{1\le j\le n} w_{ij}.$$

The maximum weight function is

$$f^*: X \times Y \to W = \{f^*(x,y): f^*(x,y) = \text{the maximum weight of the edge } xy\},$$

i.e.,

$$f^*(x_i, y_j) = \max_{1\le j\le n} x_i y_j = \max_{1\le j\le n} w_{ij}.$$

Let the favorable optimal weight (fow) $=$ min or max. The length function

$$l: X \times Y \to L = \{l_{xy}: l_{xy} = d(w_{xy}, w_{(x+1)y}) = w_{(x+1)y} - w_{xy}\}$$

the difference between the weights.

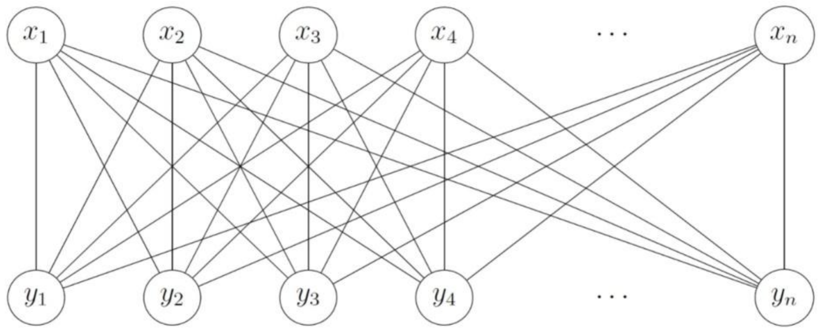


FIGURE 3.1. Network representation of $K_{n;n}$

In Figure 3.1; for $i = 1,2\cdots,n$ and $j = 1,2\cdots,n$, $x_iy_j$ denotes the edge joining the two nodes $x_i \in X$ and $y_j \in Y$ where as $w_{ij}$ denotes the weight of the edge $x_iy_j$. For any vertex $x_i$, $i = 1,2\cdots,n$, fow is

$$w_i = w(x_i) = \underset{j=1}{\overset{n}{\text{fow}}}\, w_{ij} = w_{im}. \quad (say)$$

Here $d(w_1, w_2)$, $d_*(w_1, w_2)$ and $d^*(w_1, w_2)$ we denote the difference, infimum of difference and supremum of difference between the weights of the vertices respectively.

## 4 DAS-Technique

In 2014, Das and Das [7] have developed the dominated assignment simulation (DAS) technique to solve assignment problems and have solved the minimized assignment problems and maximized assignment problems using minimin and maximax approach respectively. The main procedure of **DAS**-technique is based on three important steps which are given below.

(a) Depending on the classification of the objective function (i.e. minimization or maximization) of the *AP*, construct $n$ different assignment problems, called simulated assignment problems (SAP)
(b) Find simulated sum of each SAP.
(c) Optimal solution
 (i) For minimized assignment problem, the optimal solution is the minimum of all simulated sums.
 (ii) For maximized assignment problem, the optimal solution is the maximum of all simulated sums.
(d) Allocate the assignments.

According to Das and Das [7], the *AP* given with a balanced cost matrix of order $n$ is remodeled by $n$ cyclic assignment problems (called simulated assignment problems) (*SAP*)s by cyclic permutation of the rows as

$$\sigma_1 = SAP_1 = 123\cdots n \text{ and } \sigma_i = i(i+1)\cdots n12\cdots(i-1) = SAP_i, i = 2,\cdots,n.$$

The $i^{\text{th}}$ simulated assignment problem is

$\sigma_1 = SAP_1 = AP$ and$\sigma_i = SAP_i = [r_i, r_{i+1}, \cdots, r_n, r_1, r_2, \cdots, r_{i-1}]^T$, $i = 2,3,\cdots,n$, where $i^{\text{th}}$ row of the *AP* is the first row of the $SAP_i$ and the other $(n-1)$ rows are placed in a cyclic form. If $S_i$ is the sum of assignment values of the $SAP_i$, then optimal solution of the AP is $S = \min_i S_i$.

This technique gives unique solution for the minimized assignment problem (maximized assignment problem) with more than one perfect matching of the assignments. They have also reduced the order of the assignment problems to solve it using DAS-technique.

### Decision Matrix

In DAS technique [7], a square decision matrix of order 2 plays an important role for finding last two assignments in the *SAP*.

**Definition 4.1** [7] *Let $A = (a_{ij}) \in M_{n\times n}$ be the set of square matrix of order $n$. The element $a_{ij}$ is the off-diagonal element of $A$ if for all $i, j$, we have $i + j = n + 1$* and sum of off-diagonals of $A$ is denoted by $\text{offtr}(A)$.

### 4.1 Rule of finding the assignments from the Decision Matrix:

The decision matrix of SAP is of order 2 given by $D = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}$ where diagonal elements are $\{d_{11}, d_{22}\}$ and off-diagonal elements are $\{d_{21}, d_{12}\}$. Then trace and off-trace of $D$ are

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

respectively. Here we can obtain the last two assignments from the decision matrix for the cost matrix of minimized (or maximized) assignment problem.

### 4.2 Steps to find decision cost matrix

Let the first simulated assignment problem $(SAP_1)$, that is, the original $AP$ be given. We choose the minimum (maximum) element of the row to select as an assignment of that row if the $AP$ is minimized (maximized). To find the decision matrix from the simulated assignment problem, we follow the steps accordingly:

(a) Select the first assignment from the first row and cover its corresponding row and column.
(b) Select the second assignment from the second row and cover its corresponding row and column, continue this process upto $(n-2)^{\text{th}}$ row.
(c) If $(n-2)$ numbers of assignments are selected, then we have the decision matrix whose entities are the uncovered elements of the last two rows (that is, $(n-1)^{\text{th}}$ row and $n^{th}$ row) from which we find the last two assignments of $SAP_1$. Use this process to find the decision matrix of the problems $SAP_2, SAP_2, \cdots, SAP_n$.

### 4.3 Algorithm of Simulation technique for Minimized Assignment Problem

The algorithm of the DAS-technique for the minimized assignment problem are given as follows:

(a) Select first assignment from first row, i.e.,

$$w_{1k} = f_*(x_1, y_j) = \min_{1\le j\le n} w_{1j},$$

then cover the assignment $w_{1k}$.

(b) Select second assignment from second row, i.e.,

$$w_{2m} = f_*(x_2, y_j) = \min_{1\le j\ne k\le n} w_{2j},$$

then cover the assignment $w_{2m}$.

(c) For tie case in $i^{\text{th}}$ row, dominated column is obtained according to $(i+1)^{\text{th}}$ row, again for tie case in $(i+1)^{\text{th}}$ row, dominated column is obtained according to $(i+$

$2)^{\text{th}}$ row. For tie case in a row, the algorithm to find the dominated column and to select the assignment is given as follows:

(i) If the tie case arises in $i^{\text{th}}$ row, and $k^{\text{th}}$, $v^{\text{th}}$ columns then the assignment selection is

$$f_*(x_i, y_p) = \begin{cases} w_{ik}, & \text{if } l_{ik} = d^*(w_{ik}, w_{(i+1)k}) = w_{(i+1)k} - w_{ik} = \max; \\ \text{otherwise} & \\ w_{iv}, & \text{if } l_{iv} = d^*(w_{iv}, w_{(i+1)v}) = w_{(i+1)v} - w_{iv} = \max, \end{cases}$$

$$= w_{ip} \qquad \text{if } l_{ip} = \max, p \in \{k, v\}.$$

Cover the assignment $w_{ip}$.

(ii) Cover the rows and columns of the selected assignments from top to bottom.

(d) Using step (a) to step (c), select $(n-2)$ assignments from the first $(n-2)$ rows (top to bottom) of the simulated assignment problem.

(e) Last two assignments are obtained from the decision matrix whose elements are uncovered elements of last two rows.

(f) First simulated sum

$$S_1 = w_{1k} + w_{2m} + \cdots$$

is the sum of all selected assignments of the $SAP_1$.

(g) Use the process from (a) to (f) to find all the simulated sums $S_1, S_2, S_3, \cdots, S_n$ of the simulated assignment problems $SAP_1$, $SAP_2$, $SAP_3$, $\cdots$, $SAP_n$ respectively.

(h) The optimal solution of $AP$ is $S$ equals to minimum of all the simulated sums $S_1, S_2, S_3, \cdots, S_n$, that is,

$$S = \min_{1 \le i \le n} S_i = S_k \quad \text{(say)}.$$

(i) Allocate the feasible assignments corresponds to $S_k$.

(j) For tie case of simulated sums, i.e.,

$$S = \min_{1 \le i \le n} S_i = \{S_{k_1}, S_{k_2}, \cdots, S_{k_m}\}$$

where $k_1 < k_2 < \cdots < k_m$. Choose the simulated sum $S$ according to priority.

In the following example, the cost matrix is given for $n = 5$ and the optimal solution estimated for minimized assignment problems using **DAS**-technique.

## Algorithm of DAS-technique for Maximized Assignment Problem

The algorithm of maximized assignment problem is based on dual operation given in the algorithm for minimized assignment problem.

(a) The relation $\le$ will be replaced by $\ge$, $\min$ is replaced by $\max$,
(b) $f_*$ will be replaced by $f^*$,
(c) $d_*$ will be replaced by $d^*$ and
(d) dominated column will be replaced by dominating in the algorithm.

**Example 4.2** Let network game with 5 agents interacting over a network with cost matrix of the minimized assignment problem be given in the Table 4.3.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 13 | 8 | 16 | 18 | 19 |
| $x_2$ | 9 | 15 | 24 | 9 | 12 |
| $x_3$ | 12 | 9 | 4 | 4 | 4 |
| $x_4$ | 6 | 12 | 10 | 8 | 13 |
| $x_5$ | 15 | 17 | 18 | 12 | 20 |

Table 4.3: Cost Matrix.

In this minimized assignment problem, the agents $x_1$, $x_2$, ⋯, $x_5$ are interacting to the networks $y_1$, $y_2$, ⋯, $y_5$.

(i) For $1 \leq j \leq n$, $f_*(x_1, y_j) = \min_j w_{1j} = 8 = w_{12}$. Covering first row and 2nd column, the table is obtained as

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 13 | **8** | 16 | 18 | 19 |
| $x_2$ | 9 | 15 | 24 | 9 | 12 |
| $x_3$ | 12 | 9 | 4 | 4 | 4 |
| $x_4$ | 6 | 12 | 10 | 8 | 13 |
| $x_5$ | 15 | 17 | 18 | 12 | 20 |

Table 4.4: (SAP-1) 1st assignment selection.

(ii) For $1 \leq j \leq n, j \neq 2$,

$$f_*(x_2, y_j) = \min_j w_{1j} = 9 = \{w_{21}, w_{24}\}.$$

For 2nd row, tie case arises in 1st, 4th columns. As

$$l_{21} = d^*(w_{21}, w_{31}) = 3$$

$$l_{24} = d^*(w_{24}, w_{34}) =- 5,$$

and max $\{l_{21}, l_{24}\}$ = max $\{3, -5\} = 3 = l_{21}$ or max $\{w_{31}, w_{34}\}$ = max $\{12,4\} = 12 = w_{31}$, we have first column ($y_1$) is dominated column.

Thus the assignment for the second row is $f_*(x_2, y_j) = w_{21} = 9$. Covering the second row and first column, the table obtained as

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 13 | **8** | 16 | 18 | 19 |
| $x_2$ | **9** | 15 | 24 | 9 | 12 |
| $x_3$ | 12 | 9 | 4 | 4 | 4 |
| $x_4$ | 6 | 12 | 10 | 8 | 13 |
| $x_5$ | 15 | 17 | 18 | 12 | 20 |

Table 4.5: (SAP-1) 2nd assignment selection.

(iii) For $1 \leq j \leq n, j \neq 1,2, f_*(x_3, y_j) = \min_j w_{3j} = 4 = \{w_{33}, w_{34}, w_{35}\}$. For 3rd row, tie case arises in 3rd, 4th and 5th columns. Since

$$l_{33} = d^*(w_{33}, w_{43}) = 6$$

$$l_{34} = d^*(w_{34}, w_{44}) = 4$$

$$l_{35} = d^*(w_{35}, w_{45}) = 9$$

and since in the third row, $\max\{l_{33}, l_{34}, l_{35}\} = \max\{6,4,9\} = 9 = l_{35}$ or $\max\{w_{43}, w_{44}, w_{45}\} = w_{45}$, the dominated column is the fifth column ($y_5$), implying the assignment for the third row is $w_{35} = 4$, i.e., we have

$$f_*(x_3, y_j) = w_{35} = 4.$$

Covering the third row and fifth column, the table obtained as

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 13 | **8** | 16 | 18 | 19 |
| $x_2$ | **9** | 15 | 24 | 9 | 12 |
| $x_3$ | 12 | 9 | 4 | 4 | **4** |
| $x_4$ | 6 | 12 | 10 | 8 | 13 |
| $x_5$ | 15 | 17 | 18 | 12 | 20 |

Table 4.6: SAP-1 3rd assignment selection.

(iv) Since the cost matrix of the $AP$ is of order 5 and we have selected 3 numbers of assignments from the first three rows of the matrix, so the last two assignments will be obtained from the decision matrix $D_1$ whose elements are the uncovered elements fourth row and fifth row. The decision cost matrix $D_1$ for $SAP_1$ is

$$D_1 = \begin{bmatrix} w_{43} & w_{44} \\ w_{53} & w_{54} \end{bmatrix} = \begin{bmatrix} 10 & 8 \\ 18 & 12 \end{bmatrix}$$

which gives the assignments $w_{43} = 10$ and $w_{54} = 12$, since $\text{tr}(D_1) < \text{offtr}(D_1)$. Hence the first simulated sum is

$$S_1 = w_{12} + w_{21} + w_{35} + (w_{43} + w_{54}) = 8 + 9 + 4 + (10 + 12) = 43$$

where the bracket is given because of decision matrix assignments.

In $SAP_2$, for first three rows the assignments selected in Table 4.7 are $w_{21} = 9$, $w_{35} = 4$ and $w_{42} = 8$. Last two assignments obtained from the decision matrix

$$D_2 = \begin{bmatrix} w_{52} & w_{53} \\ w_{12} & w_{13} \end{bmatrix} = \begin{bmatrix} 17 & 18 \\ 8 & 16 \end{bmatrix}$$

are $w_{53} = 18$ and $w_{12} = 8$ since offtr$(D_2) < tr(D_2)$.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_2$ | **9** | 15 | 24 | 9 | 12 |
| $x_3$ | 12 | 9 | 4 | 4 | **4** |
| $x_4$ | 6 | 12 | 10 | **8** | 13 |
| $x_5$ | 15 | 17 | 18 | 12 | 20 |
| $x_1$ | 13 | 8 | 16 | 18 | 19 |

Table 4.7: SAP-2.

Hence the second simulated sum is

$$S_2 = w_{21} + w_{35} + w_{44} + (w_{53} + w_{12}) = 9 + 4 + 8 + (18 + 8) = 47.$$

In $SAP_3$, for first three rows the assignments selected in Table 4.8 are $w_{35} = 4$, $w_{41} = 6$ and $w_{54} = 12$. Last two assignments obtained from the decision matrix

$$D_3 = \begin{bmatrix} w_{12} & w_{13} \\ w_{22} & w_{23} \end{bmatrix} = \begin{bmatrix} 8 & 16 \\ 15 & 24 \end{bmatrix}$$

are $w_{13} = 16$ and $w_{22} = 15$, since offtr$(D_3) < tr(D_3)$. Hence the third simulated sum is $S_3 = w_{35} + w_{41} + w_{54} + (w_{13} + w_{22}) = 4 + 6 + 12 + (16 + 15) = 53$.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_3$ | 12 | 9 | 4 | 4 | **4** |
| $x_4$ | **6** | 12 | 10 | 8 | 13 |
| $x_5$ | 15 | 17 | 18 | **12** | 20 |
| $x_1$ | 13 | 8 | 16 | 18 | 19 |
| $x_2$ | 9 | 15 | 24 | 9 | 12 |

Table 4.8: SAP-3.

In $SAP_4$, for first three rows the assignments selected in Table 4.9 are $w_{41} = 6$, $w_{54} = 12$ and $w_{12} = 8$. Last two assignments obtained from the decision matrix

$$D_4 = \begin{bmatrix} w_{23} & w_{25} \\ w_{33} & w_{35} \end{bmatrix} = \begin{bmatrix} 24 & 12 \\ 4 & 4 \end{bmatrix}$$

are $w_{25} = 12$ and $w_{33} = 4$ since offtr$(D_4) < tr(D_4)$. Hence the fourth simulated sum is $S_4 = w_{41} + w_{54} + w_{12} + (w_{25} + w_{33}) = 6 + 12 + 8 + (12 + 4) = 42$.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_4$ | **6** | 12 | 10 | 8 | 13 |
| $x_5$ | 15 | 17 | 18 | **12** | 20 |
| $x_1$ | 13 | **8** | 16 | 18 | 19 |
| $x_2$ | 9 | 15 | 24 | 9 | 12 |
| $x_3$ | 12 | 9 | 4 | 4 | 4 |

Table 4.9: SAP-4.

In $SAP_5$, for first three rows the assignments selected in Table 4.10 are $w_{54} = 12$, $w_{12} = 8$, and $w_{21} = 9$. Last two assignments obtained from the decision matrix

$$D_5 = \begin{bmatrix} w_{33} & w_{35} \\ w_{43} & w_{45} \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 10 & 13 \end{bmatrix}$$

are $w_{35} = 4$ and $w_{43} = 4$ since $\text{offtr}(D_5) < tr(D_5)$. Hence the fifth simulated sum is $S_5 = w_{54} + w_{12} + w_{21} + (w_{35} + w_{43}) = 12 + 8 + 9 + (4 + 10) = 43$.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_5$ | 15 | 17 | 18 | **12** | 20 |
| $x_1$ | 13 | **8** | 16 | 18 | 19 |
| $x_2$ | **9** | 15 | 24 | 9 | 12 |
| $x_3$ | 12 | 9 | 4 | 4 | 4 |
| $x_4$ | 6 | 12 | 10 | 8 | 13 |

Table 4.10: SAP-5.

Thus, the optimal simulated sum is

$$S = \min\ \{S_1, S_2, S_3, S_4, S_5\} = \min\ \{43,47,53,42,43\} = 42 = S_4.$$

Hence, $SAP_4$ gives allocation of the assignments as follows:

$$x_4 \to y_1; x_5 \to y_4; x_1 \to y_2; x_2 \to y_5 \quad \text{and} \quad x_3 \to y_3.$$

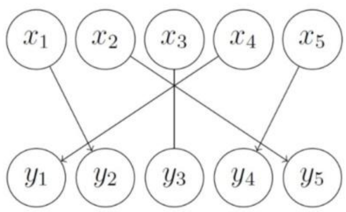


FIGURE 4.2. Network optimal solution of $K_{5;5}$

**Solution of Unbalanced Assignment Problems**

The unbalanced assignment model with $m$ workers and $n$ jobs where $m \neq n$ is also known as rectangular assignment problem. The following example showing the existence of the optimal solution in rectangular assignment problem solved using **DAS**-technique.

**Example 4.3** The owner of a small machine shop has four mechanics available to each assign each jobs for the day. Five jobs are offered with expected profit for each mechanic on each job which are given in the following Table 4.11.

| Jobs → <br> Mechanics ↓ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 62 | 78 | 50 | 101 | 82 |
| $x_2$ | 71 | 84 | 61 | 73 | 59 |
| $x_3$ | 87 | 92 | 111 | 71 | 81 |
| $x_4$ | 48 | 64 | 87 | 77 | 80 |

Table 4.11: Profit Matrix.

To find the assignment of the mechanics to the job such that the result is in maximum profit and to know the job declination, we introduce a dummy mechanic 5 with all elements 0 as given in the table 4.12.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 62 | 78 | 50 | 101 | 82 |
| $x_2$ | 71 | 84 | 61 | 73 | 59 |
| $x_3$ | 87 | 92 | 111 | 71 | 81 |
| $x_4$ | 48 | 64 | 87 | 77 | 80 |
| $x_5$ | 0 | 0 | 0 | 0 | 0 |

Table 4.12: Profit Matrix with dummy profits.

Since the problem is maximized assignment problem, the Maximax criterion will be used.

(i) The first simulated assignment problem $(SAP_1)$ given in Table 4.13.

For $1 \leq j \leq n$, $f^*(x_1, y_j) = \max_j w_{1j} = 101 = w_{14}$.

For $1 \leq j \leq n, j \neq 4$, $f^*(x_2, y_j) = \max_j w_{2j} = 84 = w_{22}$.

For $1 \leq j \leq n, j \neq 2,4$, $f^*(x_3, y_j) = \max_j w_j = 111 = w_{33}$.

Last two assignments $w_{45} = 80$ and $w_{55} = 0$ obtained from first decision matrix

$D_1 = \begin{bmatrix} w_{41} & w_{45} \\ w_{51} & w_{55} \end{bmatrix} = \begin{bmatrix} 48 & 80 \\ 0 & 0 \end{bmatrix}$ since offtr$(D_1) > tr(D_1)$. From $(SAP_1)$, we have

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 62 | 78 | 50 | **101** | 82 |
| $x_2$ | 71 | **84** | 61 | 73 | 59 |
| $x_3$ | 87 | 92 | **111** | 71 | 81 |
| $x_4$ | 48 | 64 | 87 | 77 | **80** |
| $x_5$ | **0** | 0 | 0 | 0 | 0 |

Table 4.13: SAP-1.

the first simulated sum $S_1$ as $S_1 = w_{14} + w_{22} + w_{33} + (w_{45} + w_{51}) = 101 + 84 + 111 + (80 + 0) = 376$.

(ii) The second simulated assignment problem $(SAP_2)$ given in Table 4.14

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_2$ | 71 | **84** | 61 | 73 | 59 |
| $x_3$ | 87 | 92 | **111** | 71 | 81 |
| $x_4$ | 48 | 64 | 87 | 77 | **80** |
| $x_5$ | **0** | 0 | 0 | 0 | 0 |
| $x_1$ | 62 | 78 | 50 | **101** | 82 |

Table 4.14: SAP-2.

which has the second simulated sum $S_2$ as

$$S_2 = w_{22} + w_{33} + w_{45} + (w_{51} + w_{14}) = 84 + 111 + 80 + (0 + 101) = 376$$

where the elements given in the brackets are the assignments $w_{51} = 0$ and $w_{14} = 101$ obtained from second decision matrix $D_2 = \begin{bmatrix} w_{51} & w_{54} \\ w_{11} & w_{14} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 62 & 101 \end{bmatrix}$ since $\text{tr}(D_2) > \text{offtr}(D_2)$.

(iii) In the third simulated assignment problem $(SAP_3)$ given in Table 4.15,

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_3$ | 87 | 92 | **111** | 71 | 81 |
| $x_4$ | 48 | 64 | 87 | 77 | **80** |
| $x_5$ | **0** | 0 | 0 | 0 | 0 |
| $x_1$ | 62 | 78 | 50 | **101** | 82 |
| $x_2$ | 71 | **84** | 61 | 73 | 59 |

Table 4.15: SAP-3.

the assignments for first row and second row are $w_{33} = 111$ and $w_{45} = 80$ respectively.

In the third row all the elements are $0$, so to find the dominating column. Here $y_1$ is the dominating column since the fourth row contains the unique minimum element $w_{41} = 62$. So the assignment for third row is $w_{51} = 0$. The third decision matrix is $D_3 = \begin{bmatrix} w_{12} & w_{14} \\ w_{22} & w_{24} \end{bmatrix} = \begin{bmatrix} 78 & 101 \\ 84 & 73 \end{bmatrix}$. Thus the last two assignments obtained from the decision matrix $D_3$ are $w_{14} = 101$ and $w_{22} = 84$ since $\text{offtr}(D_3) > tr(D_3)$. Hence the third simulated sum $S_3$ is

$$S_3 = w_{33} + w_{45} + w_{51} + (w_{14} + w_{22}) = 111 + 80 + 0 + (101 + 84) = 376.$$

(iv)The fourth simulated assignment problem $(SAP_4)$ is given in Table 4.16.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_4$ | 48 | 64 | **87** | 77 | 80 |
| $x_5$ | **0** | 0 | 0 | 0 | 0 |
| $x_1$ | 62 | 78 | 50 | **101** | 82 |
| $x_2$ | 71 | **84** | 61 | 73 | 59 |
| $x_3$ | 87 | 92 | 111 | 71 | **81** |

Table 4.16: SAP-4.

In $(SAP_4)$, the assignments for first row is $w_{43} = 87$. In the second row, all the elements are $0$, but in the fourth row $w_{41} = 62$ is the unique minimum element, so $y_1$ is the dominating column. Thus for second row $w_{51} = 0$ is the assignment. The assignment for third row is $w_{14} = 101$. Last two assignments obtained from the decision matrix

$$D_4 = \begin{bmatrix} w_{22} & w_{25} \\ w_{32} & w_{35} \end{bmatrix} = \begin{bmatrix} 84 & 59 \\ 92 & 81 \end{bmatrix}$$

are $w_{22} = 84$ and $w_{35} = 81$ since $\text{tr}(D_4) > \text{offtr}(D_4)$. Hence the fourth simulated sum $S_4$ is $S_4 = w_{43} + w_{51} + w$

(v) The fifth simulated assignment problem $(SAP_5)$ is given in Table 4.17.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_5$ | 0 | 0 | **0** | 0 | 0 |
| $x_1$ | 62 | 78 | 50 | **101** | 82 |
| $x_2$ | 71 | **84** | 61 | 73 | 59 |
| $x_3$ | **87** | 92 | 111 | 71 | 81 |
| $x_4$ | 48 | 64 | 87 | 77 | **80** |

Table 4.17: SAP-5.

In the first row, all the elements are $0$, but in the second row $w_{13} = 50$ is the unique minimum element, so $y_3$ is the dominating column. Thus for first row, $w_{53} = 0$ is the assignment. The assignments for second and third rows are $w_{14} = 101$ and $w_{22} = 84$ respectively. The last two assignments obtained from the decision matrix $D_5 = \begin{bmatrix} w_{31} & w_{35} \\ w_{41} & w_{45} \end{bmatrix} = \begin{bmatrix} 87 & 81 \\ 48 & 80 \end{bmatrix}$ are $a_{31} = 87$ and $a_{45} = 8$ since $\text{tr}(D_5) > \text{offtr}(D_5)$. Hence the fifth simulated sum $S_5$ is

$$S_5 = w_{53} + w_{14} + w_{22} + (w_{31} + w_{45}) = 0 + 101 + 84 + (87 + 80) = 352.$$

We have max $\{S_1, S_2, S_3, S_4, S_5\} = 376 = \{S_1, S_2, S_3\}$ where

(a) $S_1 = w_{14} + w_{22} + w_{33} + (w_{45} + w_{51}) = 101 + 84 + 111 + (80 + 0) = 376.$
(b) $S_2 = w_{22} + w_{33} + w_{45} + (w_{51} + w_{14}) = 84 + 111 + 80 + (0 + 101) = 376.$
(c) $S_3 = w_{33} + w_{45} + w_{51} + (w_{14} + w_{22}) = 111 + 80 + 0 + (101 + 84) = 376.$

In this case, the fifth mechanic is a dummy and job $y_1$ is assigned to the fifth mechanic, so this job is declined. The optimal solution $S = S_1 = 376$ is taken by index priority concept where the allocation of the assignments are

$x_1 \to y_4; x_2 \to y_2; x_3 \to y_3;$

$$x_4 \to y_5 \quad \text{and} \quad x_5 \to y_1$$

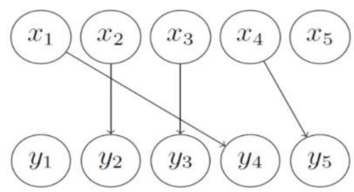


FIGURE 4.3. Network optimal solution of $K_{5;5}$

## 4.4 Minimizing the order of assignment problem

As an application Theorem 2.3 and Theorem 2.4 on has, the confirmed optimal points for the Minimized assignment problem and Maximized assignment problem can obtained by the following two rules $(a)$ and $(b)$ respectively to reduced the size of the problem.

(a) *row minimum* = *column minimum* for minimized *AP*,
(b) *row maximum* = *column maximum* for maximized *AP*.

For tie cases, select the first assignment to that element which is unique to both row and column, then cover its cell. Covering the assignments one by one, we can get a reduced assignment problem where the rest assignments can be obtained using **DAS**-technique. The reduced assignment problem contains the elements where the above two conditions fails according to type of the problem (minimize or maximize). For instance we take the problem given in Example 4.2 and Example 4.3.

**Example 4.4** The cost matrix of the minimized assignment problem, i.e., Example 4.2 is given in Table 4.3.

| | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | Row minimum |
|---|---|---|---|---|---|---|
| $x_1$ | 13 | **8** | 16 | 18 | 19 | 8 |
| $x_2$ | 9 | 15 | 24 | 9 | 12 | 9 |
| $x_3$ | 12 | 9 | **4** | 4 | 4 | 4 |
| $x_4$ | **6** | 12 | 10 | 8 | 13 | 6 |
| $x_5$ | 15 | 17 | 18 | 12 | 20 | 12 |
| Column minimum | 6 | 8 | 4 | 4 | 4 | |

Table 4.18: Cost matrix: Example 4.2

In the Table 4.18, the cells satisfying $\text{row minimum} = \text{column minimum}$ are $w_{33} = 4$ (since it is unique among row minimums), $w_{41} = 6$ and $w_{12} = 8$. Last two assignments obtained from the decision matrix $D = \begin{bmatrix} w_{24} & w_{25} \\ w_{54} & w_{55} \end{bmatrix} = \begin{bmatrix} 9 & 12 \\ 12 & 20 \end{bmatrix}$ are $w_{25} = 12$ and $w_{54} = 12$ since $\text{offtr}(D) < tr(D)$. Hence the simulated sum is

$$S = w_{33} + w_{41} + w_{12} + (w_{25} + w_{54}) = 4 + 6 + 8 + (12 + 12) = 42$$

which is the optimal solution.

The profit matrix of the maximized assignment problem is solved in Example 4.3. In Example 4.5, the same problem is solved using reduced size concept and DAS technique.

**Example 4.5** In the Table 4.19, the cells $w_{14} = 101$ and $w_{33} = 111$ are satisfying the condition $\text{row maximum} = \text{column maximum}$.

| Jobs → Mechanics ↓ | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | row maximum |
|---|---|---|---|---|---|---|
| $M_1$ | 62 | 78 | 50 | **101** | 82 | 101 |
| $M_2$ | 71 | 84 | 61 | 73 | 59 | 84 |
| $M_3$ | 87 | 92 | **111** | 71 | 81 | 111 |
| $M_4$ | 48 | 64 | 87 | 77 | 80 | 87 |
| $M_5$ | 0 | 0 | 0 | 0 | 0 | 0 |
| column maximum | 87 | 92 | 111 | 101 | 82 | |

Table 4.19: Profit matrix: Example 4.2

The reduced assignment problem is given in the Table 4.20 as follows.

| | $J_1$ | $J_2$ | $J_5$ |
|---|---|---|---|
| $M_2$ | 71 | 84 | 59 |
| $M_4$ | 48 | 64 | 80 |
| $M_5$ | 0 | 0 | 0 |

Table 4.20: Reduced order AP.

where the sum of rows maximums is $\bar{S} = 84 + 80 + 0 = 164$. Applying DAS-technique in Table 4.20, we get the first simulated sum is $S_1 = w_{22} + (w_{45} + w_{51}) = 84 + (80 + 0) = 164$ that equals to $\bar{S}$, so the assignments for Table 4.20 are $w_{22} = 84$, $w_{45} = 80$ and $w_{51} = 0$. Hence the assignments for Table 4.19 are

$$w_{14} = 101, w_{33} = 111, w_{22} = 84, w_{45} = 80 \quad \text{and} \quad w_{51} = 0.$$

The simulated sum is

$$S = w_{14} + w_{33} + [w_{22} + (w_{45} + w_{51})] = 101 + 111 + [84 + (80 + 0)] = 376$$

which equals to the optimal solution. The square bracket in $S$ indicates that the assignments are obtained from the reduced assignment problem.

## 5 Traveling Salesman Problem

In 1800s, the traveling salesman problem was mathematically formulated by the Irish mathematician W.R. Hamilton and by the British mathematician Thomas Kirkman. Hamiltons Icosian Game was a recreational puzzle based on finding a Hamiltonian cycle [2]. The general form of the TSP appears to have been first studied by mathematicians during the 1930s in Vienna and at Harvard. A traveling salesman has to visit $n$ places. He has to return to starting place after visiting all other $n-1$ places. If the starting place is $x_k$ and before returning to the place $x_k$, last visited place is $x_m$, then the path for the salesman is $x_k \cdots x_m x_k$ and the simulated sum is $S_k$. If the travelling cost from the place $x_i$ to $x_j$ is $c_{ij}$, then travelling salesman problem is to minimize the travelling cost

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} c_{ij}$$

subject to

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ and } \sum_{j=1}^{n} x_{ij} = 1,\ x_{ij} = 0 \text{ or } 1 \text{ for all } i, j = 1,2,\cdots,n.$$

### 5.1 DAST for Travelling Salesman Problems

Since the salesman is not allowed to come back to starting place before visiting all the places, so the selection column of the starting place should be neglected in the middle of the visit. Since for the salesman, destination place of one journey is starting place for the next journey, so for DAST, the selection of assignments is taken accordingly. The process of DAST for travelling salesman problem for

$$\sigma_i = SAP_i = (r_i, r_{i+1}, \cdots, r_n, r_1, r_2, \cdots, r_{i-1}, r_i)$$

are given below:

(a) for $1 \le j \le n, j \ne i$, find $f_*(x_i, y_j) = w_{ik}$ and cover the $k^{th}$ column,

(b) for $1 \le j \le n, j \ne i, k$, find $f_*(x_k, y_j) = w_{kl}$, and cover the $l^{th}$ column,

(c) for $1 \le j \le n, j \ne i, k, l$, find $f_*(x_l, y_j) = w_{lm}$ and over the $m^{th}$ column.

(d) Continue the process. After selecting $n-2$ assignments, find the last two assignments from the decision matrix which contains a starting place $x_i$ and another one is last remained visiting place $x_s$.

(e) The simulated sum $S_i$ of the path $x_i x_k x_l x_m \cdots x_s x_i$ as

$$S_i = w_{ik} + w_{kl} + w_{lm} + \cdots + w_{si}.$$

(f) For tie case, dominated column concept will be used.

## 5.2 Numerical Examples

The following TSP, multiple choice of path is obtained for the minimum travelling cost.

**Example 5.1** Consider the following travelling salesman problem so as to minimize the cost per cycle.

| X→<br>↓X | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $x_1$ | ∞ | 3 | 6 | 2 | 3 |
| $x_2$ | 3 | ∞ | 5 | 2 | 3 |
| $x_3$ | 6 | 5 | ∞ | 6 | 4 |
| $x_4$ | 2 | 2 | 6 | ∞ | 6 |
| $x_5$ | 3 | 3 | 4 | 6 | ∞ |

Table 5.21: Cost per cycle for traveling salesman.

For destination place, consider column $X = Y$, i.e., $x_i = y_i$ for $i = 1,2,3,4,5$.

(a) For SAP-1, the path is $x_1 \overset{2}{\rightarrow} y_4 = x_4 \overset{2}{\rightarrow} y_2 = x_2 \overset{3}{\rightarrow} y_5 = x_5 \overset{3}{\rightarrow} y_3 = x_3 \overset{6}{\rightarrow} y_1 = x_1$ and the simulated sum is $S_1 = 16$ whose the network is

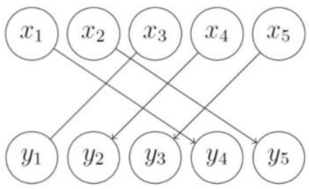


FIGURE 5.4. Network of simulated solution of SAP-1

(b) For SAP-2, the path is $x_2 \overset{2}{\rightarrow} y_4 = x_4 \overset{2}{\rightarrow} y_1 = x_1 \overset{3}{\rightarrow} y_5 = x_5 \overset{4}{\rightarrow} y_3 = x_3 \overset{5}{\rightarrow} y_2 = x_2$ and the simulated sum is $S_2 = 16$ whose the network is

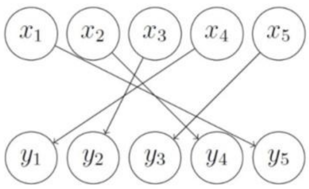


FIGURE 5.5. Network of simulated solution of SAP-2

(c) For SAP-3, the path is $x_3 \overset{4}{\rightarrow} y_5 = x_5 \overset{3}{\rightarrow} y_1 = x_1 \overset{2}{\rightarrow} y_4 = x_4 \overset{2}{\rightarrow} y_2 = x_2 \overset{5}{\rightarrow} y_3 = x_3$ and the simulated sum $S_3 = 16$ whose the network is

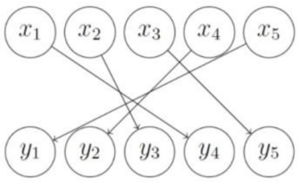


FIGURE 5.6. Network of simulated solution of SAP-3

(d) For SAP-4, the path is $x_4 \xrightarrow{2} y_1 = x_1 \xrightarrow{3} y_2 = x_2 \xrightarrow{3} y_5 = x_5 \xrightarrow{4} y_3 = x_3 \xrightarrow{6} y_4 = x_4$ and the simulated sum is $S_4 = 18$ whose the network is

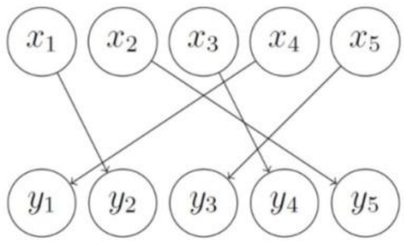


FIGURE 5.7. Network of simulated solution of SAP-4

(e) For SAP-5, the path is $x_5 \xrightarrow{3} y_1 = x_1 \xrightarrow{2} y_4 = x_4 \xrightarrow{2} y_2 = x_2 \xrightarrow{5} y_3 = x_3 \xrightarrow{4} y_5 = x_5$ and the simulated sum is $S_5 = 16$ whose the network is

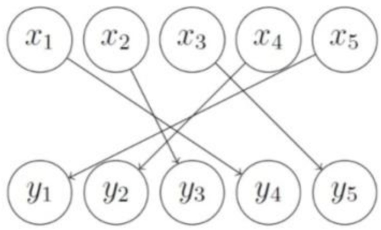


FIGURE 5.8. Network of simulated solution of SAP-5

Optimal solution of the problem is $\min\{S_1, S_2, S_3, S_4, S_5\} = 16 = \{S_1, S_2, S_3, S_5\}$. Hence the path for the salesman $P(S_i)$ are

(1) $P(S_1): x_1 \xrightarrow{2} x_4 \xrightarrow{2} x_2 \xrightarrow{3} x_5 \xrightarrow{3} x_3 \xrightarrow{6} x_1$

(2) $P(S_2): x_2 \xrightarrow{2} x_4 \xrightarrow{2} x_1 \xrightarrow{3} x_5 \xrightarrow{4} x_3 \xrightarrow{5} x_2$

(3) $P(S_3): x_3 \xrightarrow{4} x_5 \xrightarrow{3} x_1 \xrightarrow{2} x_4 \xrightarrow{2} x_2 \xrightarrow{5} x_3$

(4) $P(S_5): x_5 \xrightarrow{3} x_1 \xrightarrow{2} x_4 \xrightarrow{2} x_2 \xrightarrow{5} x_3 \xrightarrow{4} x_5$.

The reduced order TSP is given in the following table where the confirmed paths are $x_1 \rightarrow x_4$ or $x_4 \rightarrow x_1$ and $x_2 \rightarrow x_4$ or $x_4 \rightarrow x_2$.

| →Y ↓X | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | Row min |
|---|---|---|---|---|---|---|
| $x_1$ | ∞ | 3 | 6 | **2** | 3 | 2 |
| $x_2$ | 3 | ∞ | 5 | 2 | 3 | 2 |
| $x_3$ | 6 | 5 | ∞ | 6 | 4 | 4 |
| $x_4$ | 2 | **2** | 6 | ∞ | 6 | 2 |
| $x_5$ | 3 | 3 | 4 | 6 | ∞ | 3 |
| Column min | 2 | 2 | 4 | 2 | 3 | |

Table 5.22: Cost per cycle for traveling salesman..

The following traveling salesman problem is unsolvable using the rules of Hungarian method. To solve the problem, additional conditions is required. Using DAST, the problem is solved without taking additional conditions.

**Example 5.2** Consider the following traveling salesman problem so as to minimize the cost per cycle given in Table 5.23.

| X→ ↓X | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
|---|---|---|---|---|---|
| $x_1$ | ∞ | 4 | 10 | 14 | 2 |
| $x_2$ | 12 | ∞ | 6 | 10 | 4 |
| $x_3$ | 16 | 14 | ∞ | 8 | 14 |
| $x_4$ | 24 | 8 | 12 | ∞ | 10 |
| $x_5$ | 2 | 6 | 4 | 16 | ∞ |

Table 5.23: Cost per cycle for travelling salesman.

For destination place, consider column $X = Y$, i.e., $x_i = y_i$ for $i = 1,2,3,4,5$.

(a) For SAP-1, the path is

$$x_1 \overset{2}{\rightarrow} y_5 = x_5 \overset{4}{\rightarrow} y_3 = x_3 \overset{8}{\rightarrow} y_4 = x_4 \overset{8}{\rightarrow} y_2 = x_2 \overset{12}{\rightarrow} y_1 = x_1$$

and the simulated sum is $S_1 = 34$ whose the network is given in Figure 5.9.

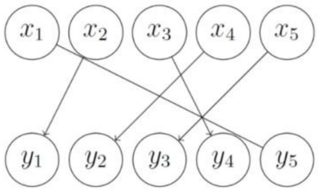


FIGURE 5.9. Network of simulated solution of SAP-1.

(b) For SAP-2, the path is

$$x_2 \overset{4}{\rightarrow} y_5 = x_5 \overset{2}{\rightarrow} y_1 = x_1 \overset{10}{\rightarrow} y_3 = x_3 \overset{8}{\rightarrow} y_4 = x_4 \overset{8}{\rightarrow} y_2 = x_2$$

and the simulated sum is $S_2 = 32$ whose the network is given in Figure 5.10.

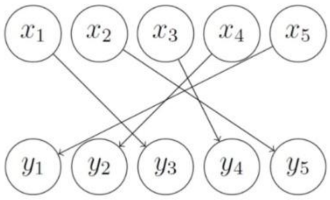


FIGURE 5.10. Network of simulated solution of SAP-2.

(c) For SAP-3, the path is

$$x_3 \overset{8}{\rightarrow} y_4 = x_4 \overset{8}{\rightarrow} y_2 = x_2 \overset{4}{\rightarrow} y_5 = x_5 \overset{2}{\rightarrow} y_1 = x_1 \overset{10}{\rightarrow} y_3 = x_3$$

and the simulated sum $S_3 = 32$ whose the network is given in Figure 5.11.

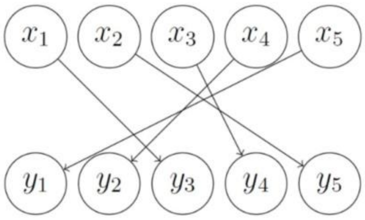


FIGURE 5.11. Network of simulated solution of SAP-3.

(d) For SAP-4, the path is $x_4 \overset{8}{\rightarrow} y_2 = x_2 \overset{4}{\rightarrow} y_5 = x_5 \overset{2}{\rightarrow} y_1 = x_1 \overset{10}{\rightarrow} y_3 = x_3 \overset{8}{\rightarrow} y_4 = x_4$ and the simulated sum is $S_4 = 32$ whose the network is given in Figure 5.12.

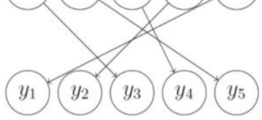


FIGURE 5.12. Network of simulated solution of SAP-4.

(e) For SAP-5, the path is $x_5 \overset{2}{\rightarrow} y_1 = x_1 \overset{4}{\rightarrow} y_2 = x_2 \overset{6}{\rightarrow} y_3 = x_3 \overset{8}{\rightarrow} y_4 = x_4 \overset{10}{\rightarrow} y_5 = x_5$ and the simulated sum is $S_5 = 30$ whose the network is given in Figure 5.13.

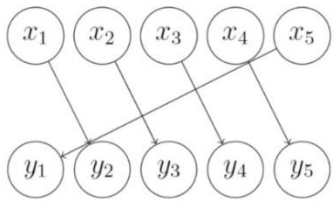


FIGURE 5.13. Network of simulated solution of SAP-5.

Optimal solution of the problem is $\min \{S_1, S_2, S_3, S_4, S_5\} = 30 = \{S_5\}$. Hence the path for the salesman is $x_5 \overset{2}{\rightarrow} x_1 \overset{4}{\rightarrow} x_2 \overset{6}{\rightarrow} x_3 \overset{8}{\rightarrow} x_4 \overset{10}{\rightarrow} x_5$. The reduced order TSP is given in the following table 5.24 where the confirmed paths are $x_1 \rightarrow x_5$ or $x_5 \rightarrow x_1$ and $x_3 \rightarrow x_4$.

| →Y<br>↓X | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | Row min |
|---|---|---|---|---|---|---|
| $x_1$ | ∞ | 4 | 10 | 14 | **2** | 2 |
| $x_2$ | 12 | ∞ | 6 | 10 | 4 | 4 |
| $x_3$ | 16 | 14 | ∞ | **8** | 14 | 8 |
| $x_4$ | 24 | 8 | 12 | ∞ | 10 | 8 |
| $x_5$ | **2** | 6 | 4 | 16 | ∞ | 2 |
| Column min | 2 | 4 | 4 | 8 | 2 | |

Table 5.24: Reduced order of TSP Example 5.2.

## 6 Result Analysis

In this section the results obtained by DAST are compared with results obtained by other existing methods with their optimal solutions. The size of the minimized assignment problem given in Example 4.2 is reduced which is shown in Example 4.4 and is solved in less number of steps. The size of the maximized assignment problem given in Example 4.3 (Rectangular AP) is reduced which is shown in Example 4.5 and is solved in less number of steps. The following Table 6.25 and Table 6.26 summarize all the results of Example 4.2, Example 4.3, Example 5.1 and Example 5.2.

| | Methods | |
|---|---|---|
| Examples | Hungarian Method | DAST |
| Ex. 4.2 | 42 | 42 |
| Ex. 4.3 | 376 | 376 |
| Ex. 5.1 | 16 | 16 |
| Ex. 5.2 | 30 | 30 |

Table 6.25: Optimal Solution.

| | Methods | |
|---|---|---|
| Examples | Hungarian Method | DAST(Reduced Order) |
| Ex. 4.2 | 64 | 47(29) |
| Ex. 4.3 | 69 | 29 |
| Ex. 5.1 | 162 | 30 |
| Ex. 5.2 | 167 | 30 |

Table 6.26: Number of Steps to find optimal solution.

The optimal favorable matching of a network N by proposed DAST and Hungarian method are coinciding 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 $n^2$, i.e., $O(n^2)$ under the assumption that all algebraic calculations can take equal time.

## 7 Conclusion

In this paper, the following conclusions are obtained.

(i) The minimin optimization theory is developed to reduce the order of the minimized assignment problems and to solve by the DAS technique.
(ii) The maximax optimization theory is developed to reduce the order of the maximized assignment problem and to solve by the DAS technique.
(iii) The process of DAS technique gives a simple approach to solve the problems.
(iv) In DAS technique, one can get the optimal solutions of the assignment problems fastly in a very simple way.
(v) This technique gives the shortest path for the traveling salesman problems (TSP) according to cyclic nonrepeated permutation form of the destination selection without any complexity.
(vi) The advantage of this method that more than one group of assignment solutions can be obtained to find the optimal solution. In case of TSP, more than one group of paths can be obtained to get optimal solution.

## References

[1] S. K. Amponsah, D. Otoo and E. Quayson, Proposed heruistic method for solving assignment probliems, American Journal of Operation Research, **6**, 436 - 441 (2016).
[2] N. Biggs, E Keith Lloyd and Robin J Wilson. Graph Theory, Oxford University Press, 1736 - 1936 (1976).
[3] R. E. Burkard, B. Klinz and R. Rudolf, Perspectives of Monge properties in Optimazation, Discrete Applied Mathematics **70**, 95 - 161 (1996).
[4] R. E. Burkard and E. C e la, Linear assignment problems and extensions. In: P.M. Pardalos and D.-Z. Du (Eds) Handbook of Combinatorial Optimization, (Dortrecht: Kluwer Academic Publishers), 75 - 149 (1999).
[5] R. E. Burkard, Selected topics on assignment problems, Discrete Applied Mathematics, **123**, 257 - 302 (1999).
[6] P. Butkovic, Max-algebra:the linear algebra of combinatorics?, Liear Algebra and its application. **367**, 313 - 335 (2003).
[7] P. K. Das and S. K. Das, Dominated Assignment Simulation Technique.Journal of the

Orissa Mathematical Society, **33(2)** , 71 - 108 (2014).
[8] G. B. Dantzig, D. R. Fulkerson and S. M. Johnson, Solution of a large scale travelling salesman problem, Journal of Operation Research Society of America, **2**, 393 - 410 (1954).
[9] Robert W. Deming, Independence numbers of graphs-an extension of the Koenig-Egervary theorem, Discrete Mathematics, **27**(1), 23 - 33 (1979). DOI:10.1016/0012-365X(79)90066-9
[10] P. B. Dimitri, A new algorithm for the assignment problem, Mathematical Programming, **21**, 152 - 171 (1981).
[11] M. M. Flood, The Travelling Salesman Problem, Operations Research, **4**, 61 - 75 (1956).
[12] Jr. L. R. Ford, D. R. Fulkerson, *A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem*, Canadian Journal of Mathematics, **9**, 210 - 218 (1957).
[13] Gibbons, Algorithmic Graph Theory. *Cambridge University Press*, Cambridge UK (1987).
[14] B. Jackson, Graph Theory and Application, *Qeen Mary University of London*, UK. http//www.Maths.qmul.ac.uk/ bill/MAS210/ch6.pdf.
[15] H.W. Khun, The Hungarian Method for the Assignment Problem, Naval Research Logistics Quarterly, **2**, 83 - 97 (1955).
[16] H. W. Kuhn, The Hungarian Method for the Assignment Problem, Naval Research Loqistics Quarterly, **2**, 83 - 97 (1955).
[17] H. W. Kuhn, Variants of the Hungarian Method for Assignment Problem, Naval Research Loqistics Quarterly, **3**, 253 - 258 (1956).
[18] Vadim E. Levit and Eugen Mandrescu, *A charaterization of König-Agerváry graphs using a common property of all maximum maching*, Electronics Notes in Discrete Mathematics, **38**, 565 - 570 (2011). https://doi.org/10.1016/j.endm.2011.09.092
[19] 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*, 1 - 22 (2018). DOI:10.13140/RG.2.2.19271.34720
[20] J. Munkres, *Algorithms for the Assignment and Transportation Problems*, Journal of the Society for Industrial and Applied Mathematics, **5(1)**, 32 - 38 (1957).
[21] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization - Algorithms and Complexity,83 -. (Mineda, NY: Dover Publications) (1998).
[22] R. S. Porchelvi and M. Anitha, Optimal solution for assignment problem by average total opertunity cost method, Journal of Mathematics and Informatics, **13** , 21 - 27 (2018). DOI: http://dox.doi.org/10.22457/jmi.v13a3
[23] S. S. Rao and M. Srinivas, *An effective algorithm to solve assignment problem opportunity cost approach*, International Journal of Mathematics and Scientific Computing, **6** , 48 - 50 (2016).
[24] H. A. Taha, Operation Research; An introduction, (Prentice-Hall of India Private ltd, New Delhi), 2005. l and Applied Mathematics, **5(1)**, 32 - 38 (1957).
[25] R. Wilson and J. Watkins, Graphs an Introductory Approach. *Wiley Publications*, New York (1990).