



Available online at <http://scik.org>

J. Math. Comput. Sci. 11 (2021), No. 4, 4810-4840

<https://doi.org/10.28919/jmcs/5695>

ISSN: 1927-5307

## AN EFFICIENT ALTERNATIVE APPROACH TO SOLVE AN ASSIGNMENT PROBLEM

A.N.M. REZAUL KARIM\*

Dept. of Computer Science & Engineering, International Islamic University Chittagong, Bangladesh

Copyright © 2021 the author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract:** There is a difficulty in assigning different resources to different tasks. Not all of them have an equal capacity to carry out certain tasks. Different entities have different capacity to complete a similar job and these different skills are expressed in terms of cost/profit/time associated with work performance. A specific type of problem with linear programming is the assignment problem. A new method has been developed in this paper to solve an assignment problem, which shows that this method provides an optimum result as well. The proposed approach has been illustrated with some numerical examples to demonstrate its effectiveness. The programming language of Python 3.8 was used to implement this novel approach.

**Keywords:** linear programming; transportation method; Hungarian method; assignment problem.

**2010 AMS Subject Classification:** 90C08.

### 1. INTRODUCTION

Assignment problem is a specific kind of linear programming problem that allocates resources in the best optimal way for different jobs [1-3]. The allocation of resources is problematic because resources, for example, men, materials, machines, money etc. have different levels of skills to

---

\*Corresponding author

E-mail address: [zakianaser@yahoo.com](mailto:zakianaser@yahoo.com)

Received March 14, 2021, Accepted April 15, 2021, Published June 15, 2021

carry out different activities, so the cost, profit or loss for performing different jobs is different. Assignment problems are one of the primary tasks for integrating scheduled processes [4]. Since the Assignment models play a significant role in logistics and supply chains, it was studied for several centuries, and still, it shows the importance of finding a cost-effective process in this modern and busy world.

There are three types of processing in formulation of Assignment Problem: parallel processing, serial processing, hybrid processing. Parallel processing means simultaneous implementation of jobs using different resources in divide construction units (work zones). Serial processing means a sequence of processes within a single working zone. Hybrid processing [5] is a combination of the last two earlier methods.

## **2. LITERATURE REVIEW**

A number of convenient methods have been identified which are suitable for solving assignment problems in different situations such as Simplex Method, Hungarian Method, Enumeration Method, and Transportation Method. However, all of these methods require a fairly difficult task (i.e., to find a route in the assignment problem table), thus the development of a cost-effective and realistic solution is highly desirable to solve the assignment problem in linear programming.

In 1955, The Hungarian method was developed by H.W. Kuhn [6] combining the thoughts of two mathematicians: D. König [7] and J. Egerváry [8]. The method consists of five steps [6, 9-11]. A new algorithm has been developed by A. Ahmed to get the best possible solution [12].

To Maximize a linear equations subject to inequalities and Computational Algorithm of the Simplex Method are proposed by Dantzig, G.B [13,14], as well as by Dr. R.G. Kedia [15] and Lemke, C. E [16]. The foundation of transportation problem was first offered by Hitchcock, [17], and then Koopmans, These two contributions have contributed to the development of transport methods. The Hungarian method is the best one among them. Although the Hungarian method is better to solve the Assignment problem but it has some limitations such as it requires a long mathematical calculations step which is quite complicated and time consuming. As a result, numerous researchers have been worked to modify the Hungarian method. This article

introduces a new algorithm to obtain a rapid way to solve the assignment problems that are quick to learn and easy to implement.

### 3. METHODOLOGY

When researchers developed an algorithm in the past, programming language and skills did not advance as rapidly. More functional techniques have emerged as a result of recent developments in accelerated computing and the advancement of user-friendly methods.

The key difference between our approach and the existing literature is that it was built using the most up-to-date computational scheme, which makes calculations very simple. Because of advances in computational language and related coding, this approach yields the desired results almost instantly after a simple input (with some basic data). This research was carried out using the advanced computational language Python.

### 4. THEORETICAL CONCEPT OF A PARTICULAR ASSIGNMENT PROBLEM

Assume there are  $n$ - jobs that need to be done and  $n$ -people are available to do those jobs.

Suppose that every person can perform every job at a term. Although with different degrees of skill, if the  $j$ -th person is assigned to the  $i$ -th job and  $c_{ij}$  be the cost. Finding an assignment is the problem (Which task should be assigned to which individual on a single basis) So that the overall cost for all jobs is a minimum, this is an assignment problem.

**Table 1:** The cost matrix ( $c_{ij}$ )

		Operator							Available
Job		$A_1$	$A_2$	$A_3$	-	-	-	$A_n$	1
	$R_1$	$c_{11}$	$c_{12}$	$c_{13}$	-	-	-	$c_{1n}$	1
	$R_2$	$c_{21}$	$c_{22}$	$c_{23}$	-	-	-	$c_{2n}$	1
	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-
	$R_n$	$c_{n1}$	$c_{n2}$	$c_{n3}$	-	-	-	$c_{nn}$	1
Required		1	1	1				1	

The cost matrix is the same as the transport problem, except that there is unity in the availability of each resource and the requirement for each destination. Here, jobs represent “sources” and machines represent “destinations.”

Let  $x_{ij}$  symbolize the assignment of j-th person to i-th job, then

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

The mathematical formulation for assignment problem can be written as

Minimize the total cost

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

Subject to constraints

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

$$i = 1, 2, 3, \dots, n$$

Which means that only one job is completed by the j-th person  $j = 1, 2, 3, \dots, n$ .

$$ii. \sum_{j=1}^n x_{ij} = 1$$

$$j = 1, 2, 3, \dots, n$$

Which means that only one person should be assigned to the i-th job,  $i = 1, 2, 3, \dots, n$ .

## 5. EXISTING METHOD

To date, the following approaches have been shown to be effective in resolving an assignment problem.

- i. Simplex method
- ii. Complete Enumeration method
- iii. Transportation method
- iv. Hungarian method (HM)

### 5.1. Simplex Method

The Simplex method is considered to solve problems related to linear programming with a great number of variables. The method progressively approaches through an iterative process and ultimately reaches the maximum or minimum values of the objective function.

Steps to simplex method:

- i. Determine the initial possible solution
- ii. Choose an input variable using the optimal condition, Turn off if there is no access variable
- iii. Select an output variable using the probability condition

The problem of assignment is mathematically formulated. there are  $n \times n$  decision variables and  $n + n = 2n$  equalities. As for example, there will be 36 decision variables and 12 equalities for a problem involving 6 workers/jobs. However, it's impossible to solve manually once more.

### 5.2. Complete Enumeration method

This method generates a list of all possible resource and job assignments. Then a low-cost assignment is the best option. If two or more assignments have the same minimum cost (or maximum profit), time, or distance, there are many optimal solutions to the problem.

When a problem entails  $n$  workers/jobs, the total number of possible assignments is  $n!$ . For instance, in the case of  $n=4$  workers/jobs problem, we have to assess a total of  $4!$  or 24 assignments. If  $n$  is big, the method is not suitable for hand calculations. As a result, only small  $n$  can benefit from this procedure.

### 5.3. Transportation Method

Transportation Problem (TP) is a special class of network optimization models, in which products are sent from different sources to different destinations, with the goal of reducing total transportation costs [18]. Factory owners who can meet the needs of buyers in a timely manner are at the forefront of business competition, thus a feasible solution of the transportation problem

becomes an important issue. The approaches used to solve the transportation problem are as follows: North West corner method, Row minima method, Column minima method, least cost method, Vogel's approximation method.

If the delivery and requirements of the product is equal ( $\sum a_i = \sum b_j$ ) then it is referred to as balanced transportation problem, or else an unbalanced transportation problem. An unbalanced transportation problem can be easily resolved by taking a dummy column or row. Mathematically, the transportation problem is a linear programming wherein the objective function has to minimize the shipping cost with demand and supply constraints.

$$\text{Minimize } z = \sum_i^m \sum_j^n c_{ij} x_{ij}$$

Subject to the constraints,

$$\sum_j^n x_{ij} = a_i, i = 1, 2, \dots, m \text{ (supply constraints)}$$

$$\sum_i^m x_{ij} = b_j, j = 1, 2, \dots, n \text{ (demand constraints)}$$

$$x_{ij} \geq 0 \text{ for all } i \text{ and } j$$

The fundamental steps in resolving a transportation problem

- i. Formulate the issue and arrange it in a matrix.
- ii. Obtain a simple, workable solution
- iii. Check for optimality in the first solution.
- iv. Refreshing the solution

#### 5.4. Hungarian Method (HM)

The Hungarian method [6] of assignment problem provides an effective way to get the best possible solution without directly comparing of each result. It is based on the standard of the cost matrix reduction to the prospect cost matrix. Opportunity cost shows relative penalties for the allocation of resources to a job, as presented to the best or least cost allocation. Optimal execution will be possible if we can decrease the cost matrix by having at least one zero for each row and column.

Firstly, make sure that the matrix is square by adding dummy rows/columns if required. Traditionally, the biggest number in the matrix is the element of a dummy row/column.

**Step 1: *Row reduction*:** select the smallest value in each row and deduct it from each element of that row.

**Step 2: *Column reduction*:** in the same way, Select the smallest value in each column and subtract it from each column item.

**Step 3:** Make a minimum number of lines covering all zero of the cost matrix entries.

**Step 4:** i) if  $N = n$ , then an optimal solution can be made ( $N =$  order of a matrix,  $n =$  number of lines)

ii) If  $N < n$ , so solution is not optimal. Then go to next step.

**Step 5:** Find out the smallest value from all exposed values, subtract this smallest value from all exposed values, and add this smallest to each cross-connected value. If a value is enclosed twice, add the minimum value to it twice. Repeat- Step 3.

## **6. PROPOSED METHOD**

### **6.1. The projected method's workflow is as follows:**

**Step 1:** Create a table and then adapt it to equilibrium if unbalanced.

**Step 2:** select all columns and find their minimum cost or time entry for the respective rows consecutively

**Step 3:** Choose the column so that there is unique cost or time entry; then set that row for the corresponding column. Highlighting those columns and rows for removal; Thus select the minimum spend or time entry for the rest.

**Step 4:** If multiple columns have a minimum cost or time at the same row, take the difference in the minimum cost or time for each of them. Choose the column with the biggest difference and accordingly assign row to respective column and then remove involved row and column.

**Step 5:** Steps 2-4 are to be repeated until there are a distinct row for each column.

## 6.2. Algorithm of the Proposed Method

An algorithm is created and presented in Table 2 after the five steps outlined above. Table 3 shows the required computational code (program code) based on this algorithm.

**Table 2:** Algorithm for calculation of the total minimum cost in assignment problem

```

set matrix:= Matrix
define assignedJob as an unsigned container

function run
for i = 0 to matrixSize
colUsed.pushBack(false) // colUsed[i] = false
for row = 0 to matrix.size()
    col = 0
while(colUsed[col++])
    col--
    define jobs as NUM in container
jobs.pushBack(getCost(matrix, col, row, colUsed)) // jobs[index] = getCost

    set i_min := smallest elements among all jobs elements
    set index := distance from firstElement of jobs container to i_min
    set colUsed[index] = true
    Job[index] = index // Jobs.pushBack(index)
    print Jobs container
function getCost (Matrix& matrix, size_t x, size_t y, vector<bool>&colUsed) return
NUM

    set pathCost = matrix.matrix[x++][y] as NUM
for col = 0 to matrix.size()
ifcolUsed[col] = false
    define min = INF as NUM
    set row := x
for row to matrix.matrix.size()
if min >matrix.matrix[row][col]
        matrix = matrix.matrix[row][col]
        set pathCost := pathCost + min
returnpathCost
// End of the function

```



**Table 3:** Python programming language in Table

```
import numpy as np

R = int(input("Enter the number of jobs:"))
C = int(input("Enter the number of machines:"))

matrix = []
print("Enter the entries rowwise:")

for i in range(R):
    a = []
    for j in range(C):
        a.append(int(input()))
    matrix.append(a)

for i in range(R):
    for j in range(C):
        print(matrix[i][j], end = " ")
    print()

import sys
minmatrix=[]

for i in range(C):
    min=sys.maxsize
    for j in range(R):
        if matrix[j][i]<min:
            min=matrix[j][i]
            mr=j
            mc=i
    minmatrix.append([min,mr,mc])

row=[ro[1] for ro in minmatrix]

from collections import Counter
i=0
sum=0
a=0
b=0
nmin=[]
job=[]
njob=[]
nrow=[]
```

```

for x in row:
    c=row.count(x)
    if c>1:
        njob.append([minmatrix[i][1],minmatrix[i][2]])
        if x not in nrow:
            nrow.append(minmatrix[i][1])
    if c==1:
        job.append([minmatrix[i][1],minmatrix[i][2]])
        sum=sum+minmatrix[i][0]
        nr=minmatrix[i][1]
        nc=minmatrix[i][2]
        for j in range(C):
            matrix[nr][j]=-1
        for j in range(R):
            matrix[j][nc]=-1
    i=i+1

uniqrow=[]
for i in range(R):
    if i not in row:
        uniqrow.append(i)
        nrow.append(i)
j=0
sub=[]
for i in range(0,len(njob),2):
    r1=njob[i][0]
    c1=njob[i][1]
    r2=njob[i+1][0]
    c2=njob[i+1][1]
    ur=uniqrow[j]
    a=abs(matrix[r1][c1]-matrix[ur][c1])
    b=abs(matrix[r2][c2]-matrix[ur][c2])
    if a>b:
        job.append([r1,c1])
        job.append([ur,c2])
        sum=sum+matrix[r1][c1]+matrix[ur][c2]
    else:
        job.append([r2,c2])
        job.append([ur,c1])
        sum=sum+matrix[r2][c2]+matrix[ur][c1]
    j=j+1
job=np.array(job)
sortjob= job[job[:,0].argsort()]
for i in range(R):
    print("Job",sortjob[i][0]+1,"assign to",sortjob[i][1]+1)

print("Total processing time is=",sum)

```

## 7. IMPLEMENTATION

### 7.1. Numerical example 1

The proposed approach is used to solve the following assignment problem. The processing times in hours are represented by the matrix entries.

**Table 4:** Cost matrix of example 1

	Operator				
job	1	2	3	4	5
1	8	10	13	11	6
2	5	14	11	9	13
3	13	12	7	6	9
4	10	8	9	13	12
5	6	11	12	8	11

**Solution:**

By applying the proposed algorithm, find their minimum time entry for each column. This is exposed in the table below:

**Table 5:** Operation 1

	Operator				
job	1	2	3	4	5
1	8	10	13	11	<b>6</b>
2	5	14	11	9	13
3	13	12	<b>7</b>	<b>6</b>	9
4	10	<b>8</b>	9	13	12
5	6	11	12	8	11

Here, operator 1, 2, and 5 has unique job entries at row 2, 4, and 1 and it does not occur again and as such assign operators 1, 2, and 5 to jobs 2, 4, and 1 respectively. As a result, mark column 1, 2, 5, and corresponding row 2, 4, and 1.

**Table 6:** Operation 2

	Operator				
job	1	2	3	4	5
1	<del>8</del>	<del>10</del>	<del>13</del>	<del>11</del>	<b>6</b>
2	<b>5</b>	<del>14</del>	<del>11</del>	<del>9</del>	<del>13</del>
3	13	12	7	6	9
4	<del>10</del>	<b>8</b>	<del>9</del>	<del>13</del>	<del>12</del>
5	6	11	12	8	11

Since, operator 3 and 4 has a single minimum time at job 3. The minimum time difference is now taken for operators 3 and 4.

Here, minimum time difference for operator 3 is  $(12-7) = 5$

Minimum time difference of operator 4 is  $(8-6) = 2$

**Table 7:** Operation 3

	Operator				
job	1	2	3	4	5
1	8	10	13	11	<b>6</b>
2	<b>5</b>	14	11	9	13
3	13	12	<b>7</b>	6	9
4	10	<b>8</b>	9	13	12
5	6	11	12	<b>8</b>	11

Since, operator 3 has the maximum time difference, assign operator 3 to job 3.

Now, only one row and column (column 4 and row 5) remains, assign operator 4 to job 5.

Therefore optimal solution is

Job 1 is assigned to operator 5

Job 2 is assigned to operator 1

Job 3 is assigned to operator 3

Job 4 is assigned to operator 2

Job 5 is assigned to operator 4

Optimal Cost is =  $(6+5+7+8+8)$  hours = 34 hours.

This completes the solution.

## 7.2. Numerical example 2

Taking into account unbalanced assignment problem where there are 4 jobs and 3 Operators.

Now we solve this example by proposed algorithm. Here the matrix entries represent the processing times in hours.

**Table 8:** An unbalanced Cost matrix

Job	Operator			
	A	B	C	
1	10	5	13	
2	3	9	18	
3	10	7	2	
4	7	11	9	

### **Solution:**

Here given problem is unbalanced and add 1 new column to convert it into a balance. Since the cost matrix is no longer balanced, we add a dummy column (D) with zero cost entries. The resulting cost matrix is shown in the table below.

**Table 9:** Operation 1 (Balanced cost matrix of example 2)

Job	Operator			
	A	B	C	D
1	10	5	13	0
2	3	9	18	0
3	10	7	2	0
4	7	11	9	0

By applying the proposed algorithm, find their minimum time entry for the particular columns.

This is presented in the table below

**Table 10:** Operation 2

Job	Operator			
	A	B	C	D
1	<del>10</del>	5	<del>13</del>	<del>0</del>
2	3	<del>9</del>	<del>18</del>	<del>0</del>
3	<del>10</del>	<del>7</del>	2	<del>0</del>
4	7	11	9	0

Since, job 1 has a single minimum time for operator B.

Since, job 2 has a single minimum time for operator A.

Since, job 3 has a single minimum time for operator C.

Therefore, optimal solution is

Job 1 is assigned to operator B

Job 2 is assigned to operator A

Job 3 is assigned to operator C

Job 4 is not assigned to operator D

Optimal Cost is =  $(5 + 3 + 2)$  hours.

= **10** hours.

This completes the solution.

### 7.3. Numerical example 3

**Table 11:** A cost matrix

	Operator				
job	1	2	3	4	5
1	13	8	16	18	19
2	9	15	24	9	12
3	12	9	4	4	4
4	6	12	10	8	13
5	15	17	18	12	20

**Solution:**

By applying the proposed algorithm, find their minimum time entry for each column. This is exposed in the table below

**Table 12:** Operation 1

	Operator				
job	1	2	3	4	5
1	<del>13</del>	<b>8</b>	<del>16</del>	<del>18</del>	<del>19</del>
2	9	15	24	9	12
3	12	9	4	4	4
4	<b>6</b>	<del>12</del>	<del>10</del>	<del>8</del>	<del>13</del>
5	15	17	18	12	20

Here, operator 1, 2 has unique job entries at row 4 and 1 and it does not occur again and as such assign operators 1, 2 to jobs 4 and 1 respectively. Consequently, mark column 1, 2 and their corresponding row 4 and 1.

Since, operator 3, 4 & 5 has a single minimum time at job 3. The minimum time difference is now taken for operators 3, 4 and 5.

## AN EFFICIENT ALTERNATIVE APPROACH TO SOLVE AN ASSIGNMENT PROBLEM

Here, minimum time difference for operator 3 is  $(24-4) = 20$

Minimum time difference of operator 4 is  $(9-4) = 5$

Minimum time difference of operator 5 is  $(12-4) = 8$

Since, operator 3 has the maximum time difference, assign operator 3 to job 3.

**Table 13:** Operation 2

	Operator				
job	1	2	3	4	5
1	<del>13</del>	<b>8</b>	<del>16</del>	<del>18</del>	<del>19</del>
2	9	15	24	9	12
3	<del>12</del>	<del>9</del>	<b>4</b>	<del>4</del>	<del>4</del>
4	<b>6</b>	<del>12</del>	<del>10</del>	<del>8</del>	<del>13</del>
5	15	17	18	12	20

Now, operator 4 & 5 has a single minimum time at job 2. The minimum time difference is now taken for operators 4 and 5.

Minimum time difference of operator 4 is  $(12-9) = 3$ ;

Minimum time difference of operator 5 is  $(20-12) = 8$

Since, operator 5 has the maximum time difference, assign operator 5 to job 2. And assign the remaining operator 4 to job 5

**Table 14:** Operation 3

	Operator				
job	1	2	3	4	5
1	<del>13</del>	<b>8</b>	<del>16</del>	<del>18</del>	<del>19</del>
2	<del>9</del>	15	<del>24</del>	<del>9</del>	<b>12</b>
3	<del>12</del>	<del>9</del>	<b>4</b>	<del>4</del>	<del>4</del>
4	<b>6</b>	<del>12</del>	<del>10</del>	<del>8</del>	<del>13</del>
5	<del>15</del>	<del>17</del>	<del>18</del>	<b>12</b>	<del>20</del>



Therefore, optimal solution is

Job 1 is assigned to operator 2

Job 2 is assigned to operator 5

Job 3 is assigned to operator 3

Job 4 is assigned to operator 1

Job 5 is assigned to operator 4

Optimal Cost = (8 + 12 + 4 + 6 + 12) hours.

= **42** hours.

This completes the solution.

## 8. OPTIMALITY TEST OF PROPOSED METHOD

To find whether solutions obtained from example 1, 2 and 3 are optimal or not by applying Hungarian Method for the above examples.

### 8.1. For numerical example 1

**Table 15:** Operation 1

	Operator					
job	1	2	3	4	5	<b>Min Value</b>
1	8	10	13	11	6	<b>6</b>
2	5	14	11	9	13	<b>5</b>
3	13	12	7	6	9	<b>6</b>
4	10	8	9	13	12	<b>8</b>
5	6	11	12	8	11	<b>6</b>

(Minimum cost or time entry in each row)

**Table 16:** Operation 2

	Operator				
job	1	2	3	4	5
1	2	4	7	5	0
2	0	9	6	4	8
3	7	6	1	0	3
4	2	0	1	5	4
5	0	5	6	2	5

(Row minimization by subtracting minimum value from each row)

**Table 17:** Operation 3

	Operator				
job	1	2	3	4	5
1	2	4	7	5	0
2	0	9	6	4	8
3	7	6	1	0	3
4	2	0	1	5	4
5	0	5	6	2	5
<b>Min</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>

(Minimum cost or time entry in each column)

**Table 18:** Operation 4

	Operator				
job	1	2	3	4	5
1	2	4	6	5	0
2	0	9	5	4	8
3	7	6	0	0	3
4	2	0	0	5	4
5	0	5	5	2	5

(Column minimization by subtracting minimum value from each column item)

**Table 19:** Operation 5

	Operator				
Job	1	2	3	4	5
1	2	4	6	5	0
2	0	9	5	4	8
3	7	6	0	0	3
4	2	0	0	5	4
5	0	5	5	2	5

[  $5 \neq 4$  (here  $N = 5$ ,  $n = 4$ ), so solution is not optimal]

**Table 20:** Operation 6

	Operator				
Job	1	2	3	4	5
1	2	4	6	5	0
2	0	9	5	4	8
3	7	6	0	0	3
4	2	0	0	5	4
5	0	5	5	2	5

(Minimum of uncovered values, minimum value is 2)

**Table 21:** Operation 7

	Operator				
job	1	2	3	4	5
1	2	2	4	3	0
2	0	7	3	2	8
3	9	6	0	0	5
4	4	0	0	5	6
5	0	3	3	0	5

(By subtracting the minimum value from all the uncovered values and add it to intersecting values)

**Table 22:** Operation 8

	Operator				
Job	1	2	3	4	5
1	2	2	4	3	0
2	0	7	3	2	8
3	9	6	0	0	5
4	4	0	0	5	6
5	0	3	3	0	5

(  $5 = 5$ ; here  $N = 5$ ,  $n = 5$ , So, optimal solution has been reached)

Now start the assignment from the row or column that has minimum of zeroes.

**Table 23:** Operation 9

	Operator					
Job	1	2	3	4	5	Minimum number of zero
1	2	2	4	3	0	1
2	0	7	3	2	8	1
3	9	6	0	0	5	2
4	4	0	0	5	6	2
5	0	3	3	0	5	2
Minimum number of zero	2	1	2	2	1	

**Table 24:** Operation 10

Job	Operator					Minimum number of zero
	1	2	3	4	5	
1	<del>2</del>	<del>2</del>	<del>4</del>	<del>3</del>	<del>0</del>	
2	<del>0</del>	<del>7</del>	<del>3</del>	<del>2</del>	<del>8</del>	
3	<del>9</del>	<del>6</del>	<del>0</del>	<del>0</del>	<del>5</del>	2
4	4	0	0	5	6	
5	0	3	3	0	5	1
Minimum number of zero			1	2		

**Table 25:** Operation 11

job	Operator					Minimum number of zero
	1	2	3	4	5	
1	<del>2</del>	<del>2</del>	<del>4</del>	<del>3</del>	<del>0</del>	
2	<del>0</del>	<del>7</del>	<del>3</del>	<del>2</del>	<del>8</del>	
3	<del>9</del>	<del>6</del>	<del>0</del>	<del>0</del>	<del>5</del>	
4	<del>4</del>	<del>0</del>	<del>0</del>	<del>5</del>	<del>6</del>	
5	<del>0</del>	<del>3</del>	<del>3</del>	<del>0</del>	<del>5</del>	1
Minimum number of zero				1		

## AN EFFICIENT ALTERNATIVE APPROACH TO SOLVE AN ASSIGNMENT PROBLEM

Therefore optimal assignment is

Job 1 is assigned to operator 5

Job 2 is assigned to operator 1

Job 3 is assigned to operator 3

Job 4 is assigned to operator 2

Job 5 is assigned to operator 4

Optimal cost is =  $(6+5+7+8+8)$  hours.

= 34 hours.

### 8.2. For numerical example 2:

Here given problem is unbalanced and add 1 new column to convert it into a balance.

**Table 26**

Job	Operator			
	A	B	C	D
1	10	5	13	0
2	3	9	18	0
3	10	7	2	0
4	7	11	9	0

**Table 27: Operation 1**

Job	Operator				Minimum value
	A	B	C	D	
1	10	5	13	0	0
2	3	9	18	0	0
3	10	7	2	0	0
4	7	11	9	0	0

(Minimum cost or time entry in each row)

**Table 28:** Operation 2

Job	Operator			
	A	B	C	D
1	10	5	13	0
2	3	9	18	0
3	10	7	2	0
4	7	11	9	0
Minimum value	3	5	2	0

(Minimum cost or time entry in each column)

**Table 29:** Operation 3

Job	Operator			
	A	B	C	D
1	7	0	11	0
2	0	4	16	0
3	7	2	0	0
4	4	6	7	0

(Column minimization by subtracting minimum value from each column item)

**Table 30:** Operation 4

Job	Operator			
	A	B	C	D
1	<del>7</del>	<del>0</del>	<del>11</del>	<del>0</del>
2	<del>0</del>	<del>4</del>	<del>16</del>	<del>0</del>
3	<del>7</del>	<del>2</del>	<del>0</del>	<del>0</del>
4	4	6	7	0

[We see,  $5 = 5$  (here  $N = 5$ ,  $n = 5$ ), so, optimal solution has been reached]

## AN EFFICIENT ALTERNATIVE APPROACH TO SOLVE AN ASSIGNMENT PROBLEM

Now start the assignment from the row or column that has minimum of zeroes.

**Table 31:** Operation 5

Job	Operator				Minimum number of zero
	A	B	C	D	
1	7	0	11	<del>∞</del>	2
2	0	4	16	<del>∞</del>	2
3	7	2	0	<del>∞</del>	2
4	4	6	7	0	1
Minimum number of zero	1	1	1	4	

Therefore optimal solution is

Job 1 is assigned to person B

Job 2 is assigned to person A

Job 3 is assigned to person C

Nothing is assigned to person D

Optimal cost = (5 + 3 + 2) hours.

= 10 hours.

This completes the solution.



**8.3. For numerical example 3:****Table 32: Operation 1**

	Operator					
job	1	2	3	4	5	Minimum value
1	13	8	16	18	19	8
2	9	15	24	9	12	9
3	12	9	4	4	4	4
4	6	12	10	8	13	6
5	15	17	18	12	20	12

(Minimum cost or time entry in each row)

**Table 33: Operation 2**

	Operator				
job	1	2	3	4	5
1	5	0	8	10	11
2	0	6	15	0	3
3	8	5	0	0	0
4	0	6	4	2	7
5	3	5	6	0	8

(Row minimization by subtracting minimum value from each row item)

**Table 34:** Operation 3

	Operator				
job	1	2	3	4	5
1	5	0	8	10	11
2	0	6	15	0	3
3	8	5	0	0	0
4	0	6	4	2	7
5	3	5	6	0	8
Minimum value	0	0	0	0	0

(Minimum cost or time entry in each column)

**Table 35:** Operation 4

	Operator				
job	1	2	3	4	5
1	5	0	8	10	11
2	0	6	15	0	3
3	8	5	0	0	0
4	0	6	4	2	7
5	3	5	6	0	8

(Column minimization by subtracting minimum value from each column item)

**Table 36:** Operation 5

	Operator				
job	1	2	3	4	5
1	<del>5</del>	0	8	10	11
2	0	6	15	0	3
3	<del>8</del>	5	0	0	0
4	0	6	4	2	7
5	3	5	6	0	8

( We get  $5 \neq 4$ ; here  $N = 5$ ,  $n = 4$ ; so solution is not optimal)

Then go to next step

**Table 37: Operation 6**

	Operator				
job	1	2	3	4	5
1	<del>5</del>	<del>0</del>	<del>8</del>	<del>10</del>	<del>11</del>
2	<del>0</del>	6	15	<del>0</del>	3
3	<del>8</del>	<del>5</del>	<del>0</del>	<del>0</del>	<del>0</del>
4	<del>0</del>	6	4	2	7
5	3	5	6	0	8

(Minimum of uncovered values, minimum value is 3)

**Table 38: Operation 7**

	Operator				
job	1	2	3	4	5
1	<del>8</del>	<del>0</del>	<del>8</del>	<del>13</del>	<del>11</del>
2	<del>0</del>	3	12	<del>0</del>	0
3	<del>11</del>	<del>5</del>	<del>0</del>	<del>3</del>	<del>0</del>
4	<del>0</del>	3	1	2	4
5	3	2	3	0	5

(Subtracting the minimum value from all the uncovered values and add it to intersecting values)

**Table 39: Operation 8**

	Operator				
job	1	2	3	4	5
1	<del>8</del>	<del>0</del>	<del>8</del>	<del>13</del>	<del>11</del>
2	<del>0</del>	3	12	<del>0</del>	<del>0</del>
3	<del>11</del>	<del>5</del>	<del>0</del>	<del>3</del>	<del>0</del>
4	<del>0</del>	3	1	2	4
5	3	2	3	0	5

(In the above,  $5 = 5$ ; here  $N = 5$ ,  $n = 5$ , So, optimal solution has been reached)

## AN EFFICIENT ALTERNATIVE APPROACH TO SOLVE AN ASSIGNMENT PROBLEM

Now start the assignment from the row or column that has minimum of zeroes.

**Table 40: Operation 9**

	Operator					
job	1	2	3	4	5	Minimum number of zero
1	<del>8</del>	<del>0</del>	<del>8</del>	<del>13</del>	<del>11</del>	1
2	<del>0</del>	<del>3</del>	<del>12</del>	<del>0</del>	<del>0</del>	3
3	<del>11</del>	<del>5</del>	<del>0</del>	<del>3</del>	<del>0</del>	2
4	<del>0</del>	<del>3</del>	<del>1</del>	<del>2</del>	<del>4</del>	1
5	<del>3</del>	<del>2</del>	<del>3</del>	<del>0</del>	<del>5</del>	1
Minimum number of zero	2	1	1	2	2	

**Table 41: Operation 10**

	Operator					
job	1	2	3	4	5	Minimum number of zero
1	<del>8</del>	<del>0</del>	<del>8</del>	<del>13</del>	<del>11</del>	
2	<del>0</del>	<del>3</del>	<del>12</del>	<del>0</del>	<del>0</del>	1
3	<del>11</del>	<del>5</del>	<del>0</del>	<del>3</del>	<del>0</del>	
4	<del>0</del>	<del>3</del>	<del>1</del>	<del>2</del>	<del>4</del>	
5	<del>3</del>	<del>2</del>	<del>3</del>	<del>0</del>	<del>5</del>	
Minimum number of zero					1	

Therefore optimal solution is

Job 1 is assigned to operator 2

Job 2 is assigned to operator 5

Job 3 is assigned to operator 3

Job 4 is assigned to operator 1

Job 5 is assigned to operator 4

Optimal Cost is =  $(8 + 12 + 4 + 6 + 12)$  hours.

= 42 hours.

**8.4.** Comparisons of optimal values between proposed method and Hungarian method are shown in the table below:

**Table 42:** Comparison of optimal values

Example	Hungarian Method	Proposed Method	Optimum
1	34	34	
2	10	10	
3	42	42	

## 9. RESULTS AND DISCUSSION

Table 42 showed us that the results of my proposed method and the Hungarian method were the same. It could be argued that our approach provides the best assignment alternatives in fewer steps than the Hungarian method. Since this approach saves time and is easy to comprehend and implement. With fewer calculated steps, the program will run faster. As a result, decision-makers would find it extremely useful. This approach can also be used to solve problems of imbalanced assignment.

## 10. CONCLUSION

We developed a new alternative approach for solving an assignment problem, and it has been shown that this method also provides the best solution. Furthermore, our approach is the same as the Hungarian method's optimal solution. Future research may include attempting to develop an algorithm that produces better results than the Hungarian method.

## FUNDING

No funds were received.

## CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

## Reference

- [1] D.F. Votaw, A. Orden, The Personnel Assignment Problem, Symposium on Linear Inequalities and Programming, Hq. U. S. Air Force, (1952) pp. 155-163.
- [2] S. Singh, A Comparative Analysis of Assignment Problem, IOSR J. Eng. 2(8) (2012), 1–15.
- [3] E. Çela, Assignment Problems, in Handbook of Applied Optimization, Part II-Applications, vol. 6, pp. 667–678, 2002.
- [4] K.M. Jaworski, Podstawy Organizacji Budowy, PWN, Warszawa, 2011.
- [5] H. Turabieh, S. Abdullah, An integrated hybrid approach to the examination timetabling problem, Omega. 39 (2011), 598–607.
- [6] H.W. Kuhn, The Hungarian method for the assignment problem, Naval Res. Log. 2 (1955), 83–97.
- [7] D. König, Über Graphen and ihre Anwendung auf Determinantentheorie und Mengenlehre, Math. Ann. 77 (1916), 453-465.
- [8] J. Egerváry, Combinatorial Properties of Matrices, ONR Logistics Project, Priencton, 1953.
- [9] K.M. Jaworski, Metodologia projektowania realizacji budowy, Wydawnictwo Naukowe PWN, Warszawa 2009, 159-162.
- [10] H.W. Kuhn, A tale of three eras: The discovery and rediscovery of the Hungarian Method, Eur. J. Oper. Res. 219 (2012), 641–651.
- [11] [www.math.harvard.edu/archive/20\\_spring\\_05/handouts/assignment\\_overheads.pdf](http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf)

- [12] A. Ahamed, A. Ahmad, A New Method for Finding an Optimal Solution of Assignment Problem, *Int. J. Mod. Math. Sci.* 12(1) (2014), 10-15.
- [13] G.B. Dantzig, Maximization of a linear function of variables subject to linear inequalities, in: Tj.C. Koopmans, ed., *Activity Analysis of Production and Allocation*, Wiley, New York, 1951, pp. 339–347.
- [14] G.B. Dantzig, *Computational Algorithm of the Simplex Method*, the RAND Corporation, Paper 394, 1953.
- [15] R.G. Kedia, A new variant of simplex method. *Int. J. Eng. Manage. Res.* 3(6) (2013), 73-75.
- [16] B.A. Cipra, The best of the 20th century: Editors name top 10 algorithms. *SIAM news*, 33(4) (2000), 1-2.
- [17] F.L. Hitchcock, The Distribution of a Product from Several Sources to Numerous Localities, *J. Math. Phys.* 20 (1941), 224–230.
- [18] M.M. Ahmed, A.R. Khan, Md.S. Uddin, F. Ahmed, A New Approach to Solve Transportation Problems, *Open J. Optim.* 5 (2016), 22–30.