"Science Stays True Here"
Journal of Mathematics and Statistical Science (ISSN 2411-2518), Vol.5, 129-145 | Science Signpost Publishing

Different Approaches to Solution of The Assignment Problem Using R Program

Öznur İşçi Güneri, Burcu Durmuş, Dursun Aydın

Department of Statistics, Muğla Sıtkı Koçman University, Kötekli Campus, 48000, Muğla, Turkey.

Abstract

The aim of the current study is to emphasize the importance of an heuristic solution method for the classical assignment problem (AP). In the literature, many different algorithms including both classical and heuristic algorithms have been developed to solve AP. In this study, we introduce newly R program codes for the classical and heuristic algorithms. Note that Brute Force algorithm, Hungarian algorithm, and Linear Programming (LP) algorithm are konown as classical algorithms, while the Greedy is considered as the heuristic algorithm. For this purpose, we made an application based on 4x4 dimensional sample. In addition, four different methods are obtained for different dimensional problems. The outcomes from the study show that both classical methods and the Greedy method provides the optimal or near optimal results.

Keywords: Assignment Problem, Linear Programming (LP), Brute Force Method, Hungarian Algorithm, Greedy Method.

1. Introduction

The assignment problem, one of the fundamental optimisation problems, is a linear programming model which is arranged to match the resources (employee, machine etc.) with varying tasks. The key idea is to get maximum profit or sale and to get minimum cost. Since AP is a special type of transplantation problem, it needs to provide the following expressions:

- Equilibration of supply and demand (square matrix)
- Supply from every supply node is one
- Every demand node has a demand of one
- The optimal result value is an integer, such as 0 or 1

Although all the algorithms developed for transportation problem are suitable for the assignment problem, solving the assignment problem as a transportation problem is complicated and time consuming. Thus, new methods have been developed for the assignment problem.

Some of the classical algorithms developed for AP solution are; Branch Boundary Algorithm, Brute Force Algorithm, and Hungarian Algorithm. There are also heuristic algorithms (genetic algorithms, parelel auction algortihm, penalty method, and greedy method) for solving this problem. Among these, the Greedy algorithm is one of the most preferred methods because it is easy to apply and requires a small number of iterations. There are more than 10 algorithms that are similar in performances (and which one is the best depends on specific situations) in these days, and studies about this topic are still on-going [1]. Classical and heuristic algorithms for assignment problems are given in Table 1.

Table 1: Algortihms for assigment problems

Classical Algorithms	**HeuristicAlgorithms**
Simplex Method (LP)	OACE Algorithm
Transportation Method	Auction Algorithm
Brute Force Algorithm	Greedy Algorithm
The Branch Boundary Algorithm	Genetic Algorithm
Hungarian Algorithm	Harmony Search Algorithm
Lagrangian Relaxation Algorithm	Penalty Method

Among diverse approaches developed for assignment problem, Hungarian method [2,3], linear approaches [4], and the newer one- Auction algorithm [5] and heuristic algorithms are commonly used.

The Branch Boundary Algorithm, reported by Breu and Burdet (1974), Wolsey and Nemhauser (1988), and Rijavec (1992) is the only known algortihm for solving the multidimentional assigment problem optimally [6, 7, 8]. For problems of any size, since the time required for computing the optimal solution grows exponentially with the size of the problem, this algortihm is impratical [9]. Since assignment problem in Operations Research is a main problem, a large number of studies have been conducted about this topic and various algorithms have emerged [10].

Aringhieri et al. (2015) presented a two-stage heuristic algorithm for the assignment problem. The algorithm was tested using real data collected from a state hospital in Genova, Italy. The results show that the proposed method performs well in terms of both solution quality and computation time [11]. When the computer solutions are taken into consideration with the classical solution methods, an increasing number of (n) and increasing processing time due to the increased memory requirement and high degeneration are encountered. Therefore, there are also special solution methods which are different from the classical methods and use their characteristics. But none of these methods is as well known as the Hungarian solution. This method is almost recognized enough to be referred to together with the assignment problem [12].

Aktel et al. (2017) have shown that metaheuristic algorithms for door assignment problems provide good results at a reasonable time for large-scale door assignment problems [13]. Seethalakshmy and Srinivasan (2018) discussed a methodology to determine the transportation problem in their study. In their study, they saw that transport could not be kept away from the assignment, in the end, proposed a new algorithm that combined the assignment in the transport [14].

Reyes et al. (2019) conducted a comprehensive literature review for assignment problems. In this study, they examined 71 representative articles and discussed solution methods. They have found that there are many solution methods for the assignment problem (classical, heuristic, meta-heuristic ...). They classified 71 representative papers according to solution methods, performance measures and constraints or concerns. The authors think that the current review is the most comprehensive and detailed study to date [15]. Various other linear network flow problems can be reformulated as assignment problems. To name a few the classical shortest path problems, there are max-flow problems, linear transportation problems, and linear minimum cost flow problems. Therefore, the solution of assignment problems is important. Comparison of computational complexity for different optimal assignment algorithms is given in Table 2 [16] .

Table 2: Comparison of the computational complexity for different optimal assignment algorithms

Algorithms for AP	**Complexity order**
Brute Force Method	$O(n!)$
OACE Algorithm	$O(n^4)$
Hungarian Algorithm	$O(n^3)$
Auction Algorithm	$O(n^2)$
Greedy Algorithm	$O(n^2 logn)$

In this study, the solution of Brute Force, Hungarian Method, and heuristic Greedy algorithm are discussed.

2. Model and Analysis

2.1 The Assignment Problem

The assignment problem is a special form of general linear programming problems. Suppose we have m workers and n machines. We know the cost of assigning machines to workers. We aimed to make an assignment which minimizes the cost. The optimal assignment is the assignment that minimizes the cost. It is accepted that the number of workers is equal to the number of machines in the assignment problem $(m = n)$ (balanced). If m is greater than n $(m > n)$, the amount of dummy machine is added to the model. If m is smaller than n $(m < n)$, the amount of dummy workers is provided by participating to the equation of $(m = n)$. The cost of the dummy machine and dummy workers (c_{ij}) is zero. Mathematically when m is equal to n $(m = n)$, the linear programming form of AP is as follows:

The objective function is:

$$Min\ Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

Subject to:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\dots,m \quad i = 1,2,\dots,m$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1,2,\dots,m \quad i = 1,2,\dots,n$$

$$x_{ij} = 0\ or\ x_{ij} = 1$$

for example (4x4) the linear programming of the assignment problem is as follows,

$$Min\ Z = c_{11}x_{11} + c_{12}x_{12} + \cdots + c_{44}x_{44}$$

Subject to:

$$x_{11} + x_{12} + x_{13} + x_{14} = 1$$

$$x_{21} + x_{22} + x_{23} + x_{24} = 1$$

$$x_{31} + x_{32} + x_{33} + x_{34} = 1$$

$$x_{41} + x_{42} + x_{43} + x_{44} = 1$$

$$x_{11} + x_{21} + x_{31} + x_{41} = 1$$

$$x_{12} + x_{22} + x_{32} + x_{42} = 1$$

$$x_{13} + x_{23} + x_{33} + x_{43} = 1$$

$$x_{14} + x_{24} + x_{34} + x_{44} = 1$$

$$x_{ij} = 0\ or\ x_{ij} = 1$$

Cost matrix is used to make the resolution of assignment problems more efficient. Let c_{ij} be the cost of assigment the i *th* resource to the j *th* task. We define the cost matrix to be the mxn matrix in Table 3.

Table 3: Assignment matrix

Resources (workers)		**Activities (**machines**)**				**Supply**
		machine 1	machine 2	**…**	machine n	
c_{ij}=	worker 1	c_{11}	c_{12}	…	c_{1n}	1
	worker 2	c_{21}	c_{22}	…	c_{2n}	1
	.	.			.	.
	.	.			.	.
	.	.			.	.
	worker m	c_{m1}	c_{m2}	…	c_{mn}	1
	Demand	1	1	…	1	mxn

AP is often made for minimization problems. However, some assignment problems are for maximization instead of minimization. In this case, the largest value element in the table is extracted from all the elements in the table to make a solution. The other steps are as in the minimization problems.

2.2 Algorithms for The Assignment Problem

2.2.1 Brute Force Method

The Brute Force Method that calculates the cost of all assignments is quite complex method, especially for large assignments. Suppose we take mxn matrix assignment for the equality of $(m = n)$. There are n option for the first assignment and $n - 1$ option for the second assignment, respectively. Thus, the solution has an exponential run time [17]. There are $n!$ =6 assignments for $n = 3$. This solution may be suitable for small n values. In this context, Table 4 shows the iterator numbers of the Brute Force method:

Table 4: Number of iterations for Brute Force Method

Table Size	**Solution**	**Number of iterations**
3x3	3!	6
4x4	4!	24
5x5	5!	120
6x6	6!	720
7x7	7!	5040
8x8	8!	40320
9x9	9!	362880
10x10	10!	3628800

.	.	.
.	.	.
.	.	.
100x100	100!	≈9.3x10157

2.2.2 Hungarian Method

The given steps are applied to the nxn cost matrix to find the optimal assignment [18].

Step 1: The smallest element in each row is subtracted from all elements in the row.

Step 2: The smallest element in each column is subtracted from all elements in the column.

Step 3: A minimum number of lines are drawn to cover all 0 in the cost matrix.

Step 4: Step 4: Test for Optimality:

i) If the minimum number of lines is n, an optimal assignment and we are finished.

ii) If lines is less than n, an optimal assignment is not yet possible. In that case, go to Step 5.

Step 5: Determine the smallest entry not covered by any line. Subtract this entry from each uncovered row, and then add it to each covered column. Return to Step 3.

If the number of rows and columns in the cost matrix are unequal, the assigment problem is unbalanced. In this case it is not appropriate to use the Hungarian algorithm. Thus any assigment problem should be balanced before it is solved by the Hungarian method [19, 20].

2.2.3 Greedy Method

A Greedy Algorithm essentially makes the best, near-sighted decision at each stage of the problem in hopes of finding a good solution. In this case, the algorithm will choose the lowest cost worker to be assigned to the task as the first assignment, then choose the next lowest cost worker to be assigned to the task, and so on until all tasks have been assigned. The algorithm repeats this procedure until all workers have at least one task [21]. The greedy heuristic continues to cover the row and column of the smallest uncovered entry in the cost matrix until all entries are covered. The resulting set of entries then constitute the assignment of workers to jobs [22]. Greedy algorithm for the Linear Assigment Problem (LAP):

Step 1: Locate the smallest element and delete the row and column where it is located.

Step 2: Find the smallest second element and delete the row and column where it is located.

Step 3: Repeat the process until there are no rows and columns to delete.

Step 4: Selected values give local optimal results.

Heuristics algortihms have the chance to find the optimal answer, but can get trapped in a local optimal solution. Greedy algorithms try to get close to the optimal solution by improving a candidate solution iteratively, with no guarantee that an optimal solution will actually be found. The Greedy Algorithm alone will find a good answer, but randomizing parts of the algorithm will ensure that multiple answers are possible. Modifying Greedy Algorithm such that it sometimes accepts an assignment that temporarily worsens the objective function will succeed in leaving the local optimum and possibly find the global optimal solution.

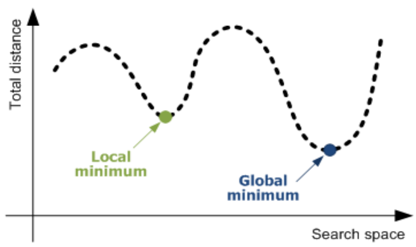

Figure 1: Local minimum and Global minimum for optimal solution

Figure 1 shows that some solutions should be skipped for the most global solution of local search. Thus, the heuristic method (such as Greedy) examines quickly a part of the search field and finds local ends.

3. Analysis and Discussion

3.1 Brute Force Method

This is the original cost matrix (4x4):

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

LP formulation;

$$\text{Min Z} = 14\text{x}_{11} + 5\text{x}_{12} + \cdots + 10\text{x}_{44}$$

Constraints:

$$x_{11} + x_{12} + x_{13} + x_{14} = 1$$

$$x_{21} + x_{22} + x_{23} + x_{24} = 1$$

$$x_{31} + x_{32} + x_{33} + x_{34} = 1$$

$$x_{41} + x_{42} + x_{43} + x_{44} = 1$$
$$x_{11} + x_{21} + x_{31} + x_{41} = 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} = 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} = 1$$
$$x_{ij} = 0 \; or \; x_{ij} = 1, i = 1,2,3,4 \; and \; j = 1,2,3,4$$

Brute Force method;

Colums				**Solution**
P1	P2	P3	P4	14+12+3+10=39
P1	P2	P4	P3	14+12+9+6=41
P1	P4	P3	P2	14+5+3+4=26
P1	P4	P2	P3	14+5+8+6=33
P1	P3	P2	P4	14+6+8+10=38
P1	P3	P4	P2	14+6+9+4=33
P2	P4	P3	P1	5+5+3+2=15* (min)
…				
P4	P1	P3	P2	7+2+3+4=16
P4	P1	P2	P3	7+2+8+6=23

Optimal solution is 15.

3.2 Hungarian Method

This is the original cost matrix:

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

We subtract the row minimum from each row:

9	0	3	2	(-5)
0	10	4	3	(-2)
4	5	0	6	(-3)
0	2	4	8	(-2)

We subtract the column minimum from each column:

9	0	3	0
0	10	4	1
4	5	0	4
0	2	4	6
			(-2)

and

10	0	3	0	
0	9	3	0	(-1)
5	5	0	4	
0	1	3	5	(-1)
(+1)				

and

10	0	3	0
0	9	3	0
5	5	0	4
0	1	3	5

Cover all zeros with a minimum number of lines. There are 4 lines required to cover all zeros. Because there are 4 lines required, the zeros cover an optimal assignment.

10	0	3	0
0	9	3	0
5	5	0	4
0	1	3	5

This corresponds to the following optimal assignment in the original cost matrix:

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

The optimal value is equal to 15.

3.3 Greedy Method

This is the original cost matrix:

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

The smallest element in this matrix is 2. The element (2,1) is taken randomly.

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

Its row and column are closed. The smallest element is 3. (3,3) and its row and column are closed.

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

Repeat step, the smallest element is 4. (4,2) and its row and column are closed.

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

Only the (1,4) element remains.

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

This value is 7. As a result, 2+3+4+7 = 16 is obtained.

4. Conclusions

For solving the assignment problems, classical or heuristic different methods have been developed. Packet programs are widely used for classical solution methods. However, since the number of programs used to solve the assignment problem is limited by the number of dimensions, such as iteration, there is no ready software that can solve the complex regardless of the size of the complex (such as complex, non-mathematically expressed, constraints not stretched or nonlinear problems). Therefore, software development is important. For the AP, more software development solutions were obtained with heuristic methods. In the recent years, it can be used in solving the problems of assigment both classical and heuristic methods by writing code with R package program which is widely used.

In this study, solutions for different methods are shown by using R program. In the scope of the study, Brute Force, Hungarian, Greedy and LP solutions of 3x3, 4x4, 5x5, 6x6, 7x7, 8x8, 9x9 and 10x10 dimensions are given in Table 5 (R codes are given in Appendix 1). The Brute Force approach is a simple and easy solution when the number of dimensions is small. However, as the number of dimensions increases, the number of operations is long. Assignment problem can be written and solved in LP form. However, as the number of dimensions increases, the number of constraints and variables increases. Therefore, the solution is extended as in my Brute Force approach.

The most commonly used iterative solution is the solution to the problems by Hungarian algorithms assignment. When compared to Burte Force and LP method, it provides optimal results in less time. From the heuristic methods, the Greedy method is one of the preferred methods because the algorithm is simple. However, this method does not guarantee optimal results, local solution is obtained. However, since the process load will increase as the number of dimensions increases, the Greedy method will find a solution in a shorter time, so it can be recommended to use in complex or large scale problems.

Table 5: R program solution for different examples optimal assignment algorithms

Table Size	**Algorithms**			
	LP Method	**Brute Force** $n!$	**Hungarian** n^3	**Greedy** $n^2 \log n$
3x3	9	9	9	9
4x4	40	15	15	16
5x5	31	31	31	31
6x6	142	142	150	167-174
7x7	38	38	38	38
8x8	41	41	41	51
9x9	72	72	72	77
10x10	58	58	58	58

References

[1].Xin Jin, Parallel Auction Algorithm for Linear Assignment Problem, Project Report, 2017. https://stanford.edu/~rezab/classes/cme323/S16/projects_reports/jin.pdf

[2].Harold W. Kuhn, The Hungarian Method for The Assignment Problem, Naval Research Logistics, Vol:2, 83-97, 1955.

[3].James Munkres, Algorithms for the Assignment and Transportation Problems, Journal of the Society of Industrial and Applied Mathematics, Vol: 5, 32-38, 1957.

[4].George B. Dantzig, Application of The Simplex Method to a Transportation Problem, in: Activity Analysis of Production and Allocation (Proceedings Conference on Linear Programming, Chicago, Illinois, 1949; Tj. C. Koopmans, ed.), Wiley, New York, 359–373, 195

[5].Dimitri P. Bertsekas, A New Algorithm for The Assignment Problem, Mathematical Programming, Vol: 21, 152-171, 1981.1.

[6].Raymond Breu, Claude-Alain Burdet, Branch and Bound Evperiments in 0-1 Programming, in Baliski, M.L., ed., Mathematical Programming Study 2: Approaches to Integer Programming, North Holland Publishing Company, Amsterdam, 1974.

[7].Laurence A. Wolsey, George L. Nemhauser, Integer and Comninational Optimization, John Wiley and Sons, New York, 1988.

[8].Nenad Rijavec, A Lagrangian Relaxation Algortihm for Some Multidimentional Assigment Problems, Ph.D. Thesis, Colorado State University, Fort Collins, 1992.

[9].Aubrey B. Poore and Nenad Rijavec, A Numerical Study of Some Data Association Problems Arising in Multitarget Tracking, 1994.

[10]. Daniella Goldfarb, Efficient Dual Simplex Methods for The Assignment Problem, Mathematical Programming, Vol:33, 187-203, 1985.

[11]. Roberto Aringhieri, Paola Landa, Patrick Soriano, Elena Tanfani and Angela Testi, A Two Level Metaheuristic for the Operating Room Scheduling and Assignment Problem, Computers and Operations Research, Vol:54, 21-34, 2015.

[12]. Adalet Öner, Füsun Ülengin, Atama Problemi İçin Yeni Bir Çözüm Yaklaşımı, itüdergisi/d Mühendislik, Vol:2, 73-79, 2003.

[13]. Abdullah Aktel, Betül Yagmahan, Tuncay Özcan, M. Mutlu Yenisey, Engin Sansarcı, The Comparison of the Metaheuristic Algorithms Performances on Airport Gate Assignmnet Problem, Transportation Research Procedia, Vol:22, 469-478, 2017.

[14]. A. Seethalakshmy and N. Srinivasan, A Heuristic Approach to Obtain an Optimal Solution for Transportation Problem Using Assignment, Taga Journal, Vol: 14, 1862-1870, 2018.

[15]. Juan Jose Rojas Reyes, Elyn Lizeth Solano-Charris and Jairo Rafael Montoya-Torres, The Storage Location Assignment Problem: A Literature Review, International Journal of Industrial Engineering Computations, Vol: 10, 199-224, 2019.

[16]. Frans-Willem Vermaak, The Optimal Assignment Problem: An Investigation Into Current Solutions, New Approaches and the Doubly Stochastic Polytope, Faculty of Engineering, University of the Witwatersand, Master Thesis, Johannesburg, 2010.

[17]. Longin Jan Latecki, Assignment Problem: Hungarian Algorithm and Linear Programming, 2019. https://cis.temple.edu/~latecki/Courses/AI.../MatchingV1.ppt

[18]. Samuel Kwame Amponsah, Dominic Otoo, Said Salhi, E. Quayson, Proposed Heuristic Method for Solving Assignment Problems, American Journal of Operations Research, Vol:6, 436-441, 2016.

[19]. Wayne L. Winston, Operation Research Applications and Algortihms, Thomson Learning Academic Resource Center, Fourth Edition, 2004.

[20]. Billy E. Gillett, Introduction to Operation Research, McGarw-Hill, New York, 1976.

[21]. Allison M. Douglas, A Modified Greedy Algorithm for The Task Assignment Problem, Master Thesis, Department of Industrial Engineering, University of Louisville, 2007.

[22]. Jacob Lewellen, A Greedy Approximation Algorithm for the Linear Assignment Problem, 2017.

[23]. Sam Buttrey, Integer Programming for The Assignment Problem, 2015.
https://cran.r-project.org/web/packages/lpSolve/lpSolve.pdf

Appendix-1. R Program Codes

R program codes for the 4x4 sample problem are given below;

LP Method R Codes; (lp.assign)

lp.assingn for R program [23].

```
> assign.costs <- matrix (c(14,5,8,7,2,12,6,5,7,8,3,9,2,4,6,10), 4, 4)
> assign.costs
> lp.assign (assign.costs)$solution
     [,1] [,2] [,3] [,4]
[1,]   14    2    7    2
[2,]    5   12    8    4
[3,]    8    6    3    6
[4,]    7    5    9   10

     [,1] [,2] [,3] [,4]
[1,]    0    0    0    1
[2,]    1    0    0    0
[3,]    0    0    1    0
[4,]    0    1    0    0
```

The objective function is 15.

Brute Force Method R Codes;

```
> x <- t(matrix(c(14,5,8,7,2,12,6,5,7,8,3,9,2,4,6,10), nrow = 4))
> p <- permn(1:4)
> r <- nrow(x)
> y <- 0   > t <- 1
> i <- 1   > k <- 1
> array1 <- c(y)      > array2 <- c(y)
> while ( i <= length(p))
+ { while (k <= r)
+    { y = y + x[,p[[i]][k]][k]
+       array1[k] <- y
+       k <- k+1 }
+    array2[i] <- array1[r]
+       i <- i+1  k <- 1 }
> j <- 2
```

```
> array3 <- c(y)
> array3[1] <- array2[1]
> while(j <= length(p))
+ {array3[j] <- array2[j]-array2[j-1]
+    j <- j+1 }
> print(array3)
 [1] 39 41 33 23 16 26 33 38 28 23 24 24 23 23 31 37 28 22 15 24 32 23 22 20
> print(paste("Z=", min(array3)))
[1] "Z= 15"
```

Hungarian Method R Codes; (solve_LSAP)

```
> x <- matrix(c(14,5,8,7,2,12,6,5,7,8,3,9,2,4,6,10), nrow = 4)
> solve_LSAP(x)
> y <- solve_LSAP(x)
> sum(x[cbind(seq_along(y), y)])
Optimal assignment:
1 => 4, 2 => 1, 3 => 3, 4 => 2
[1] 15
```

Greedy Method R Codes;

```
> assignment_Greedy <- function(objection)
+ {   row <- sqrt(length(objection))
+    table <- t(matrix(objection, nrow = row))
+    x <- 1
+    array <- c(x)
+    while (length(table) > 1)
+    {
+       array[x] <- table[which.min(table)]
+       table[which.min(table)] <- NA
+       table <- table[complete.cases(table),complete.cases(t(table))]
+       x <- x+1
+    }
+    assign <- c(array, table)
+    result <- sum(matrix(assign)[,1])
+    print(assign)
+    print(paste("Z=", result)) }

> objection <- c(14,5,8,7,2,12,6,5,7,8,3,9,2,4,6,10)
```

```
> assignment_Greedy(objection)
[1] 2 3 4 7
[1] "Z= 16"
```

The Greedy method randomly assigned the first step. Therefore, the result is not optimal but close to the optimal.

Published: Volume 2019, Issue 6 / June 25, 2019