AN EXACT METHOD TO FIND ALL K^{TH} BEST SOLUTIONS FOR AN ASSIGNMENT PROBLEM

[1]ALI AL-HASANI, [2]MASAR AL-RABEEAH, [3]SANTOSH KUMAR, [4]ANDREW EBERHARD

[1,2,3,4]School of Mathematical and Geospatial Sciences, RMIT University, Melbourne, Australia.
[1,2]Department of Mathematics, College of Sciences, University of Basrah, Al- Basrah, Iraq.

Abstract - In this paper, we present a new method to find all (n!) solutions for a given n × n assignmentproblem.Thesesolutionscanbearrangedinanincreasingorder, therefore all K ranked solutions can be obtained, $K \geq 2$. The method has been programmed in C language and tested on 60 instances of different sizes, $4 \leq n \leq 20$. Later performance quality of the random search method to find K^{th} best solutions is investigated for a given value of K.

Keywords - Ranked solutions, Assignment Problem, K^{th} Best Solution, Exact Method.

I. INTRODUCTION

The assignment problem (AP) is often faced by decision-makers in industrial situations. Many studies dealing with the assignment model and its variations such as, linear, quadratic, multi- index and generalized assignment have been studied[1],wheremorethan100studieshavebeen presented. The most commonly used method to find the optimal solution for an AP is the Hungarian method [10] with complexity order $O(n^4)$, n being the order of the AP. Assignment method has been used in a different context in by Kumar et al. [11, 12]. Complexity of the Hungarian approach has been enhanced to $O(n^3)$ [9], which is known as shortest augmenting path algorithm. In addition, several software and websites have been developed to find an optimal solution to a given AP, such as Cplex, Lingo and Tora[6,18,19]etc.

In many applications, for practical reasons, one needs to find ranked solutions i.e. the second best, third best, etc. For example, when one uses a navigation device to reach a certain address, the device determines the best way to reach the desired destination in fastest time or the shortest route. But for some reasons, such as congestion or road-work on that specified route, one may be forced to find an alternative route. This will be called the second-best solution. Similarly, the decision-makers have reasons to find ranked best solutions in certain situations, therefore ranked-best solutions have an important role in many practical situations. Determination of the ranked best solution in optimization theory, in general, is an interesting but difficult problem that needs more consideration by researchers.

In this study our interest is confined to the ranked best solutions of anAP, therefore were view available methods developed to find the K^{th}best solutions to an AP. Murty[15] developed a method to find the K^{th}best solution of an assignment problem, $K \geq 2$. Murthy's approach was based on blocking, one at a time, the optimal assignment that is if i^{th} job was assigned to j^{th} person, he suggested to set $C_{ij} = \infty$. Hamacher and Queyranne [7] developed an alternative approach for ranking solutions to combinatorial problems. Chegireddy and Hamacher [5] applied [7] to bipartitematchings problems. This process depends on the shortest cycle, which can be obtained by solving at most n shortest path problems, time complexity order is $O(Kn^3)$. Pascoal et al. [16] considered the subsetsinreverseorderandtheywereabletore-optimizethesolutionfromtheprevioussubset consideredandfoundthebestassignmentbysolvingasing leshortestpathproblem,yieldingthe same time complexity as in [4].Also Lu and Rosenbaum [14] have developed method for ranking assignment solutions. Recently, Pedersen et al. [17] have further modified Murthy's approach with some encouragingresults.

In this paper, we developed a new technique to find all K^{th}best solutions for an AP. The method is easy to understand and can be implemented by any programming language. It can find all ranked K^{th}best solutions for moderately large size APs. The paper also investigates performance of the random search method to find K^{th}best solutions [13].

This paper has been organized in 5sections. Mathematical background has been discussed in Section 2, the proposed method is presented in Section 3. The method is illustrated in Section 4, where several instances of different size problems have been presented. In this section, performance of the random search method to find the K^{th} has also been evaluated. Conclusion and future studies are discussed in Section5.

II. MATHEMATICAL BACKGROUND

An assignment is a well-known problem involving n jobs to be assigned to n persons,eachperson is assigned to exactly one job and each job is assigned to a person. The objective function is to minimize the total cost. A mathematical model of an assignment problem is givenby:$\text{MinZ} = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$ (1)

Subject to

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

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

$$x_{ij} = \begin{cases} 1\ if\ \ i^{th}\ job\ is\ assigne\ to\ j^{th}\ person \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases};\quad j = 1,2,\dots,n \quad \text{and} \quad i = 1,2,\dots,n.$$

The following notation have been used in the following discussion.

Notation2.1X^* represents an optimal solution to the given assignment problem. It is an nelementstring. Let the value of the objective function at this point be denoted by Z^*.In this paper, since our interest is to find all solutions and corresponding values of the objective function, the optimal solution and its objective function value will be represented by $X_1$and Z_1respectively. In general, a solution string will be denoted by X_H; $H = 1,2,\dots,n!$ and the value of the objective function at that point will be denoted by $Z^{K^{th}}$; $K \leq n!$ This miss-match arises due to multiple solutions leading to the same value of the objective function.

Notation2.2When more than one distinct solutions resulting in the same value of the objective Z, they are denoted by: X_{H_r}, such that $r \geq 1$.

Notation 2.3 For a given feasible solution from the set X_H, any change in the existing solution will require a change in at least two basic variables.

Notation 2.4From any given feasible solution, consisting of n string elements, at most $n(n-1)/2$ new feasible solutions can be generated. These $n(n-1)/2$ solution strings and their values of the objective functions at that point are denoted by: $X_{H,j}$; and$Z_{\{K,1\}}, Z_{\{K,2\}}, \dots, Z_{\left\{K,\left\{\frac{n(n-1)}{2}\right\}\right\}}$. The solution strings are rearranged in increasing order of the values of the objective function. If there are more than one solution with the same value of the objective function, they are arbitrarily arranged in the list.

Notation2.5When a string has been fully examined for all new feasible solutions, that string is labeled as $X_{H_r}(E)$ and all new solutions are denoted by $X_H$and Z_K.

Notation 2.6 Number of distinct solutions to the given assignment problem is denoted by NA and when NA = n!, the search ends. It may be noted that position of the string X_1 and Z_1 are fixed and the position of the other remain temporary until end of the search.

There are a few interesting observations about the assignment model (1). Proof is provided, where considered necessary.

Observation2.1Foragivenn × nassignment problem, the total number of feasible solutions are given byn!.

Observation2.2 For a given feasible solutiont on ×n assignment problem, the number of non-basic cells is an even number given by $n(n-1)$.

Observation2.3Given a feasible solution to an $n \times n$ assignment problem, at mostn (n-1)/2 numberofdifferentsolutionscanbegeneratedbymakinganallocation,oneatatime,inthe non-basic cell (p,q).

Proof.Consideranon-basiccell(p,q)i.e.thecellinthep[th]rowandtheq[th]column.Sincethe non-basic cell corresponds to a current feasible solution to the given AP, there must be an assignment in the p row and the q column. Let this assignment be denoted by (p, q_1), note $q_1 \neq q$ and similarly the column q must have an assignment in some other row. Let this row be denoted by (p_1, q), $p_1 \neq p$. This is illustrated in Table 1. It may be noted that the cell (p_1, q_1) is also a non-basic cell with respect to the current feasible solution. Holding on to the remaining (n -2) assignmentsintheircurrentpositions,anewfeasiblesolutioncanbegeneratedbymovingthe currentallocationsinthecells$(p_1,q)$and(p,q_1)tocells(p,q) and (p_1,q_1),respectively. This process will reverse if we commence with the non-basic cell (p_1, q_1). Therefore, one can conclude that from the n(n-1) non-basic cells, at most n(n-1)/2 new feasible assignment solutions can begenerated.

III. EXACT METHOD FOR RANKED SOLUTIONS

Finding the all K[th]best solution for any optimization problem becomes more and more demanding for large size problems. Here, the authors have attempted to find all ranked K[th]best solutions for a given AP, $2 \leq n \leq 20$, which for the model 1 is given by n!. The method is comprised of the following steps.

Step 1: Initial process

Step 1.1 Using the Hungarian method of assignment, find the optimal solution $X_1 = \left\{x_{1s_1}, x_{2s_2}, \dots, x_{ps_p}, \dots, x_{ns_n}\right\}$, herex$_{ps_p}$ indicatesanassignmentinthep[th]rows$_p$thColumn.

$P \setminus J$	J_1	J_2	⋯		J_q	⋯	J_{q_1}	⋯	J_n
P_1	x_{11}	x_{11}	⋯		x_{1q}	⋯	x_{1q_1}	⋯	x_{1n}
P_2	x_{21}	x_{22}	⋯		x_{2q}	⋯	x_{2q_1}	⋯	x_{2n}
.	.	.	⋯		.	⋯	.	⋯	.
.	.	.	⋯		.	⋯	.	⋯	.
.	.	.	⋯		.	⋯	.	⋯	.
P_p	x_{p1}	x_{p2}	⋯		x_{pq}	⋯	x_{pq_1}	⋯	x_{1n}
.	.	.	⋯		.	⋯	.	⋯	.
.	.	.	⋯		.	⋯	.	⋯	.
.	.	.	⋯		.	⋯	.	⋯	.
P_{p_1}	x_{p_11}	x_{p_12}	⋯		x_{p_1q}	⋯	$x_{p_1q_1}$	⋯	x_{1n}
.	.	.	⋯		.	⋯	.	⋯	.
.	.	.	⋯		.	⋯	.	⋯	.
.	.	.	⋯		.	⋯	.	⋯	.
P_n	x_{n1}	x_{n2}	⋯		x_{nq}	⋯	x_{nq_1}	⋯	x_{nn}

Table 1: Assignment Optimal solution

Step 1.2 From the solution obtainedfrom Step 1.1, develop two lists of basic and non-basic variables as follows:

List 1:$\{x_{1s_1}, x_{2s_2}, \dots, x_{ps_p}, \dots, x_{ns_n}\} = X_1$.

List 2: $\{x_{ij_1}, x_{ij_2}, \dots, x_{ij_i}, \dots, x_{ij_n}; (j_i = 1,2, \dots, n; j_i \neq s_i), i = 1,2, \dots, n.\}$

The List1 is comprised of number of basic cells and List2 will haven$(n-1)$numberofnon- basic cells. Develop two sets, the first set X is made of the assignment strings and the second set is for corresponding values of the objective function, denoted by Z for those assignment solutions. Let X_K= $\{X_1\}$ andZ_K= $\{Z^1\}$.

Step 1.3 Initially $\{X = X_1 = X^*\}$,$Z = \{Z_1 = Z^*\}$,NA = 1

Step 2:

Step 2.1 Findalln(n-1)/2 distinct feasible solutions from the selected solution X_1 and corresponding values of the objectivefunction, as explained in Observation 2.3 and Table 1. Let the distinct number of solutions obtained be denoted by 'na'.

Step 2.2 Rearrange all solutions in an increasing order of the value of the objective function. Note the position of the string X_1 will not change. The number of feasible solutions NA = NA+na.

Step 2.3 Attach a label E to the solution which has been fully searched for feasible solutions.

Step2.4Selecttheunlabeledsolutionthatcorrespondstotheminimumvalueoftheobjective function.

Step 3 Repeat the Step 2

Step 4: If NA $\leq$ n!, return to Step 2 else Stop.

Theprocedurehas been developed as Algorithm 1, which willbeexplainedbygivinganillustrative4X4 example.

It may be noted that generally all possible solutions to a given AP problem are seldom required, however, when optimal solution is not acceptable for any reason, a need for alternative solutions arises. These solutions are often near the optimal solution and they are known as the K[th]best solutions, K $\geq$2. Earlier an attempt was made to find the K[th]best using the random search method [8, 14]. The results obtained by the random search method are evaluated in this paper.

Algorithm 1 An exact method to find all k-best solution

Step1.Findtheoptimalsolution$X_1 \rightarrow X$ and $Z_1 \rightarrow Z$

Step 2.Input

2.1 Number of assignment solution NA =n!

2.2 NumberofsearchesNS=n(n−1)/2.

2.3 $S_1 = r_1$; // $\{r_1$: number of the solutions with the same value of the objective function$\}$.

Step 3.

3.1. S_2 =0

3.2. $S_2 = S_2 + 1$.

3.3. Choose non-basic cell$(x_{ij})_{S_2}$ in X_1.

3.4. $(x_{ij})_{S_2} \rightarrow (x_{ij})^*_{S_2}$(basic cell).

3.5. Find X_{1,S_2} and Z_{1,S_2}.

3.6. If $X_{1,S_2} \notin X$then Add$X_{1,S_2} \rightarrow X$ and Add $Z_{1,S_2} \rightarrow Z$

3.7. $S_2 = S_2 + 1$. and $S_1 = S_1 + 1$.

3.8. If $S_2 < NS$ GOTO 3.1.

3.9. Find Min $\{Z/Z_1\}=Z_2$ with X_2

3.10. $X_1 = X_2$

3.11. If $S_1 < NA$ GOTO 3.1.

Step 4.Ranked Z.

Step 5. Stop.

A. An illustrative example

Consider the assignment problem 2, where we find all solutions NA = 4! = 24.

assignment problem

	J_1			
P_1	8	4	9	4
P_2	5	5	7	8
P_3	3	1	6	3
P_4	3	6	1	8

optimal assignment solution

	J_1			
	4	0	5	0
	0	0	2	3
	2	0	5	2
	2	5	0	7

Step 1.

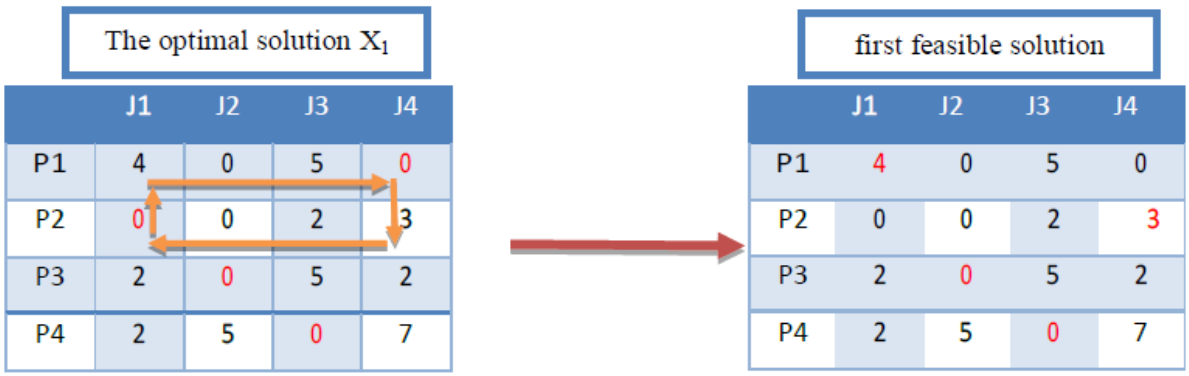
Solving the assignment problem by the Hungarian method, the optimal assignment solution is given by: Thus X_1 = $\{x_{14}, x_{21}, x_{32}, x_{43}\}$ giving Z_1= 11. List1: $\{x_{14},x_{21},x_{32},x_{43}\}$.

List 2: $\{x_{11}, x_{12}, x_{13}; x_{22}, x_{23}, x_{24}; x_{31}, x_{33}, x_{34}; x_{41}, x_{42}, x_{44}\}$.

$X = \{X_1\}$, $Z = \{Z_1\}$, $NA = 1$.

NS=4(4-1)/2 =6

Develop 6 distinct feasible solutions from X_1 and corresponding values of the objective function. Select the non-basic cell $x_{1,1}$ from the List 2. This cell will replace the current basic cells $x_{1,4}$ and $x_{2,1}$ and will be matched by the non-basic cell $x_{2,4}$. Similarly, the remaining 5 other feasible solutions from the solution X_1 and List 2 are generated, as given in Table 2.



K	non-basic cell	corresponding cell	X_{1K}	Z_{1K}
1	x_{11}	x_{24}	$x_{11}, x_{24}, x_{32}, x_{43}$	18
2	x_{12}	x_{34}	$x_{12}, x_{21}, x_{34}, x_{43}$	17
3	x_{13}	x_{44}	$x_{13}, x_{21}, x_{32}, x_{43}$	23
4	x_{22}	x_{31}	$x_{14}, x_{22}, x_{31}, x_{43}$	13
5	x_{23}	x_{41}	$x_{14}, x_{23}, x_{32}, x_{41}$	15
6	x_{33}	x_{42}	$x_{14}, x_{21}, x_{33}, x_{42}$	21

Table 2: Search for alternative solutions from X_1

Rearrange $Z_1, Z_{1,1}, Z_{1,2}, \ldots, Z_{1,6}$solutions in an increasing order of the value of the objective function as shown in Table 3. From Table 3, we have a potential 2[nd]best solution for an exhaustive search for more feasible solutions.

K	X_K	Z_K
1	$(x_{14}, x_{21}, x_{32}, x_{43})_E$	11
2	$x_{14}, x_{22}, x_{31}, x_{43}$	13
3	$x_{14}, x_{23}, x_{32}, x_{41}$	15
4	$x_{12}, x_{21}, x_{34}, x_{43}$	17
5	$x_{11}, x_{24}, x_{32}, x_{43}$	18
6	$x_{14}, x_{21}, x_{33}, x_{42}$	21
7	$x_{13}, x_{21}, x_{32}, x_{43}$	23

Table 3: Feasible solutions after exhaustive search from X_1

The same process is repeated for the solution X_2= x14, x22, x31, x43,Z_2 = 13, which are given in Table 4. It may be noted that the string X_2 generated five new feasible solutions and repeated the optimal solution. Thus,

we have a total number of 12 feasible solutions shown in Table 5.

K	non-basic cell	corresponding cell	X_{2K}	Z_{2K}
1	x_{11}	x_{34}	$x_{11}, x_{22}, x_{34}, x_{43}$	17
2	x_{12}	x_{24}	$x_{12}, x_{24}, x_{31}, x_{43}$	16
3	x_{13}	x_{44}	$x_{13}, x_{22}, x_{31}, x_{44}$	25
4	x_{21}	x_{32}	$x_{14}, x_{21}, x_{32}, x_{43}$	11
5	x_{23}	x_{42}	$x_{14}, x_{23}, x_{31}, x_{42}$	20
6	x_{33}	x_{41}	$x_{14}, x_{22}, x_{33}, x_{41}$	18

Table 4 Feasible solutions after exhaustive search from X_2

K	X_K	Z_K
1	$(x_{14}, x_{21}, x_{32}, x_{43})_E$	11
2	$(x_{14}, x_{22}, x_{31}, x_{43})_E$	13
3	$x_{14}, x_{23}, x_{32}, x_{41}$	15
4	$x_{12}, x_{24}, x_{31}, x_{43}$	16
5	$x_{11}, x_{22}, x_{34}, x_{43}$	17
6	$x_{12}, x_{21}, x_{34}, x_{43}$	17
7	$x_{11}, x_{24}, x_{32}, x_{43}$	18
8	$x_{14}, x_{22}, x_{33}, x_{41}$	18
9	$x_{14}, x_{23}, x_{31}, x_{42}$	20
10	$x_{14}, x_{21}, x_{33}, x_{42}$	21
11	$x_{13}, x_{21}, x_{32}, x_{43}$	23
12	$x_{13}, x_{22}, x_{31}, x_{44}$	25

Table 5: Feasible solutions after exhaustive search from X_1 and X_2

This process continues with (X_3 =x14, x23, x32, x41, Z_3=15) until we reach the target number of 24 distinct solutions. These solutions are given in Table 6.

Z^K	Z value	X	N.S
Z^1	11	$x_{14}, x_{21}, x_{32}, x_{43}$	1
Z^{2nd}	13	$x_{14}, x_{22}, x_{31}, x_{43}$	2
	13	$x_{12}, x_{21}, x_{34}, x_{43}$	
Z^{3dr}	15	$x_{14}, x_{23}, x_{32}, x_{41}$	1
Z^{4th}	16	$x_{12}, x_{24}, x_{31}, x_{43}$	1
Z^{5th}	17	$x_{11}, x_{22}, x_{34}, x_{43}$	

	17	$x_{12}, x_{23}, x_{34}, x_{41}$	2
Z^{6th}	18	$x_{14}, x_{22}, x_{33}, x_{41}$	2
	18	$x_{11}, x_{24}, x_{32}, x_{43}$	
Z^{7th}	20	$x_{14}, x_{23}, x_{31}, x_{42}$	2
	20	$x_{13}, x_{22}, x_{34}, x_{41}$	
Z^{8th}	21	$x_{13}, x_{24}, x_{32}, x_{41}$	3
	21	$x_{14}, x_{21}, x_{33}, x_{42}$	
	21	$x_{12}, x_{24}, x_{33}, x_{41}$	
Z^{9th}	22	$x_{12}, x_{23}, x_{31}, x_{44}$	1
Z^{10th}	23	$x_{13}, x_{21}, x_{32}, x_{44}$	3
	23	$x_{13}, x_{21}, x_{34}, x_{42}$	
	23	$x_{12}, x_{21}, x_{33}, x_{44}$	
Z^{11th}	24	$x_{11}, x_{23}, x_{34}, x_{42}$	2
	24	$x_{11}, x_{23}, x_{32}, x_{44}$	
Z^{12th}	25	$x_{13}, x_{22}, x_{31}, x_{44}$	1
Z^{13th}	26	$x_{13}, x_{24}, x_{31}, x_{42}$	1
Z^{14th}	27	$x_{11}, x_{22}, x_{33}, x_{44}$	1
Z^{15th}	28	$x_{11}, x_{24}, x_{33}, x_{42}$	1
Total Solutions			24
CPU time			0.0074

Table 6: Kth best assignment solution for illustrative example 3.1

IV. COMPUTATIONAL STUDY

Inthissection,15 assignment problems of different sizes 4×4, 6×6, 8×8, 10×10 and 20×20wererandomly generated and tested for the effectiveness of the proposed algorithm. The coefficient of the objective function wasalso randomly generated between [0, 20]. The proposed algorithmwas implemented in C, using Cplex 12.6 to read lp files, which represents the assignment models, and run on HP laptop Via VMware Workstation 14 player (Lubuntu 16.04 LTS x64.) with 1 GB of RAMmemory.

4.1. Instances size 4×4

The method was applied for the C_1, C_2, C_3 . These results are given in Tables 7, 8 and 9.

$$C_1 = \begin{bmatrix} 11 & 3\,6 & 5 \\ 13 & 52 & 16 \\ 19 & 211 & 4 \\ 12 & 101 & 1 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 9 & 18\,1 & 9 \\ 10 & 27 & 1 \\ 19 & 915 & 17 \\ 19 & 1013 & 19 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 16 & 1417 & 19 \\ 7 & 137 & 14 \\ 15 & 1818 & 9 \\ 15 & 96 & 11 \end{bmatrix}$$

$Z^{K^{th}}$	Z value	Solution string X_n	N.S.
Z^{1}	16	$x_{11},x_{23},x_{32},x_{44}$	1
Z^{2nd}	21	$x_{11},x_{22},x_{34},x_{43}$	5
		$x_{12},x_{23},x_{34},x_{41}$	
		$x_{12},x_{21},x_{34},x_{43}$	
		$x_{14},x_{23},x_{32},x_{41}$	
		$x_{14},x_{21},x_{32},x_{43}$	
Z^{3rd}	22	$x_{13},x_{21},x_{32},x_{44}$	1
Z^{4th}	25	$x_{12},x_{23},x_{31},x_{44}$	1

Z^{5th}	27	$x_{13},x_{22},x_{34},x_{41}$	2
		$x_{11},x_{23},x_{34},x_{42}$	
Z^{6th}	28	$x_{11},x_{22},x_{33},x_{44}$	2
		$x_{12},x_{21},x_{33},x_{44}$	
Z^{7th}	30	$x_{14},x_{22},x_{31},x_{43}$	2
		$x_{11},x_{24},x_{32},x_{43}$	
Z^{8th}	31	$x_{13},x_{22},x_{31},x_{44}$	1
Z^{9th}	33	$x_{14},x_{22},x_{33},x_{41}$	2
		$x_{13},x_{21},x_{34},x_{42}$	
Z^{10th}	36	$x_{14},x_{23},x_{31},x_{42}$	2
		$x_{13},x_{24},x_{32},x_{41}$	
Z^{11th}	39	$x_{12},x_{24},x_{31},x_{43}$	2
		$x_{14},x_{21},x_{33},x_{42}$	
Z^{12th}	42	$x_{12},x_{24},x_{33},x_{41}$	1
Z^{13th}	48	$x_{11},x_{24},x_{33},x_{42}$	1
Z^{14th}	51	$x_{13},x_{24},x_{31},x_{42}$	1
Total Solutions			24
CPU time			0.0017

Table 7: K[th]best assignment solution for C_1

Z^K	Z value	Solution string X_n	N.S.
Z^1	30	$x_{13},x_{24},x_{32},x_{41}$	1
Z^{2nd}	31	$x_{13},x_{24},x_{31},x_{42}$	1
Z^{3dr}	32	$x_{11},x_{24},x_{32},x_{43}$	1
Z^{4th}	35	$x_{11},x_{24},x_{33},x_{42}$	1
Z^{5th}	38	$x_{13},x_{21},x_{34},x_{42}$	1
Z^{6th}	39	$x_{13},x_{22},x_{34},x_{41}$	2
		$x_{13},x_{21},x_{32},x_{44}$	
Z^{7th}	41	$x_{11},x_{22},x_{34},x_{43}$	3
		$x_{13},x_{22},x_{31},x_{44}$	
		$x_{14},x_{21},x_{32},x_{43}$	
Z^{8th}	43	$x_{14},x_{22},x_{31},x_{43}$	2
		$x_{11},x_{23},x_{34},x_{42}$	
Z^{9th}	44	$x_{14},x_{21},x_{33},x_{42}$	3
		$x_{14},x_{23},x_{32},x_{41}$	
		$x_{11},x_{23},x_{32},x_{44}$	
Z^{10th}	45	$x_{11},x_{22},x_{33},x_{44}$	3
		$x_{14},x_{22},x_{33},x_{41}$	
		$x_{14},x_{23},x_{31},x_{42}$	
Z^{11th}	51	$x_{12},x_{24},x_{31},x_{43}$	1
Z^{12th}	53	$x_{12},x_{24},x_{33},x_{41}$	1
Z^{13th}	58	$x_{12},x_{21},x_{34},x_{43}$	1
Z^{14th}	61	$x_{12},x_{23},x_{34},x_{41}$	1
Z^{15th}	62	$x_{12},x_{21},x_{33},x_{44}$	1
Z^{16th}	63	$x_{12},x_{23},x_{31},x_{44}$	1
Total Solutions			24
CPU time			0.0092

Table 8: K[th]best assignment solution for the instance C_2

Z^K	Z value	Solution string X_n	N.S.
Z^1	36	$x_{12},x_{21},x_{34},x_{43}$	1
Z^{2nd}	41	$x_{11},x_{23},x_{34},x_{42}$	1
Z^{3dr}	42	$x_{13},x_{21},x_{34},x_{42}$	1
Z^{4th}	44	$x_{11},x_{22},x_{34},x_{43}$	1
Z^{5th}	45	$x_{12},x_{23},x_{34},x_{41}$	1
Z^{6th}	47	$x_{12},x_{23},x_{31},x_{44}$	1
Z^{7th}	49	$x_{12},x_{24},x_{31},x_{43}$	1
Z^{8th}	50	$x_{14},x_{21},x_{32},x_{43}$ $x_{14},x_{23},x_{31},x_{42}$ $x_{12},x_{21},x_{33},x_{44}$	3
Z^{9th}	52	$x_{11},x_{23},x_{32},x_{44}$	1
Z^{10th}	53	$x_{13},x_{21},x_{32},x_{44}$ $x_{14},x_{21},x_{33},x_{42}$ $x_{14},x_{22},x_{31},x_{43}$	3
Z^{11th}	54	$x_{11},x_{24},x_{32},x_{43}$ $x_{13},x_{22},x_{34},x_{41}$	2
Z^{12th}	55	$x_{13},x_{24},x_{31},x_{42}$	1
Z^{13th}	56	$x_{13},x_{22},x_{31},x_{44}$	1
Z^{14th}	57	$x_{11},x_{24},x_{33},x_{42}$	1
Z^{15th}	58	$x_{11},x_{22},x_{33},x_{44}$	1
Z^{16th}	59	$x_{14},x_{23},x_{32},x_{41}$	1
Z^{17th}	61	$x_{12},x_{24},x_{33},x_{41}$	1
Z^{18th}	64	$x_{13},x_{24},x_{32},x_{41}$	1
Z^{19th}	65	$x_{14},x_{22},x_{33},x_{41}$	1
Total Solutions			24
CPU time			0.0055

Table 9: K[th]best assignment solution for the instanceC_3

Z^K	C1	C2	C3
Z^1	24	21	31
Z^{2nd}	25	23	36
Z^{3dr}	28	24	37
Z^{4th}	29	26	38
Z^{5th}	31	28	39
. . .	. . .	. . .	. . .
$Z^{6!th}$	90	87	97
NS	6!	6!	6!
CPU	0.3286	0.8	0.72

Table 10: K[th]best assignment solution for instances size 36

Z^K	C1	C2	C3
Z^1	33	27	37
Z^{2nd}	35	28	38
Z^{3dr}	36	29	39

$Z^{4^{th}}$	38	30	40
$Z^{5^{th}}$	39	31	41
. . .	. . .	. . .	. . .
$Z^{8!^{th}}$	131	135	127
NS	8!	8!	8!
CPU	5.072	6.023	6.052

Table 11: K[th]best assignment solution for instances size 64

4.2. More computational results

A few more results on instances ofsize6×6,8×8,10×10 and 20 × 20, were tested and results are given in Tables 10,11,12 and 13, respectively.

Z^K	C1	C2	C3
Z^1	35	37	26
Z^{2nd}	36	39	29
Z^{3dr}	37	40	30
$Z^{4^{th}}$	38	41	31
$Z^{5^{th}}$	39	42	32
. .	. .	. .	. .
$Z^{100!^{th}}$	177	178	156
NS	10!	10!	10!
CPU	15.086	14.128	13.087

Table 12: K[th]best assignment solution for instances size 100

Z^K	C1	C2	C3
Z^1	35	39	38
Z^{2nd}	36	40	40
Z^{3dr}	37	42	41
$Z^{4^{th}}$	38	43	42
$Z^{5^{th}}$	39	44	43
. . .	. . .	. . .	. . .
$Z^{20!^{th}}$	352	358	357
NS	20!	20!	20!
CPU	21.0234	22.199	22.186

Table 13: K[th]best assignment solution for instances size 400

4.3. Comparison Exact method and Random searchmethod [14]

Although, exact method will provide the $K^{th}\ best\ , K \geq 2$. In many practical situations, instead of all solutions, interest generally arises in only a few solutions, close to the optimal solution in meaningful time. One such method developed by the authors is known as the Random search method [13], where guarantee of correctness is governed by a probability factor. In context of a mathematical programming problem, the random search method was discussed by [8].

In this section, performance of this random search method [14] is evaluated for an assurance factor of 0.80. It may be noted that three instances of sizes 4×4, 6×6, 8×8, 10×10 and 20×20 for K =1, 2, 3, 4 and 5 were tested by using the exact and the random search methods as given in Table 11. Out of 75 values, the random search method has missed out on only 5 values which is 6.2% but is able to handle

much larger size problems. It may be noted from Table 11 that the CPU time for instances of size 4×4 for the exact method are less than the Random search method. However, it starts increasing for instances of higher sizes, as the exact method finds all solutions.

V. CONCLUSION

A new method for determination of all solutions to an assignment problem has been developed. It is a known fact that determination of all solutions will quickly become time consuming, hence it is desirable to compare performance of the random search method [13], which can tackle larger problems in realistic time and provide required answers. A need for such K^{th} best solutions has been pointed by the authors in search of non-dominated solution set for a multi-objective integer problem [1, 2, 3].

REFERENCES

[1] Al-Hasani, Ali, Al-Rabeeah, Masar, Kumar, Santosh, and Eberhard, Andrew. "Finding all non- dominated points for a bi-objective generalized assignment problem". In: Journal of Physics: ConferenceSeries.Vol.1218.1.IOPPublishing.2019,p.012004.

[2] Al-Hasani, A., Al-Rabeeah, M., Kumar, S. and Eberhard, E., 2018. An improved cpu time in triangle splitting method for solving a biobjective mixed integer program. Int J Math EngManag Sci, 3(4), pp.351-364.

[3] Al-Rabeeah,Masar,Kumar,Santosh,Al-Hasani,Ali,Munapo,Elias,andEberhard,Andrew."Bi-objective integer programming analysis based on the characteristic equation". In: International Journal of System Assurance Engineering and Management(2019), pp. 1–8.

[4] Burkard, Rainer E, Dell'Amico, Mauro, and Martello, Silvano. "Assignment problems". Springer, 2009.

[5] Chegireddy, Chandra R and Hamacher, Horst W. "Algorithms for finding k-best perfect matchings". In: Discrete applied mathematics 18.2 (1987), pp.155–165.

[6] CPLEX, IBM ILOG. "12.6". In: CPLEX User's Manual (2014).

[7] Hamacher, Horst W and Queyranne, Maurice. "K best solutions to combinatorial optimization problems". In: Annals of Operations Research4.1 (1985), pp. 123–143.

[8] Huynh, H-N and Kumar, Santosh. "A random search method for extreme point mathematical programming."In:Asia-PacificJournalofOperationalResearch7.1(1990),pp.30–45.

[9] Jonker, Roy and Volgenant, Anton. "A shortest augmenting path algorithm for dense and sparse linear assignment problems". In: Computing38.4 (1987), pp. 325–340.

[10] Kuhn, Harold W. "The Hungarian method for the assignment problem". In: Naval researchlogistics quarterly 2.1-2 (1955), pp.83–97.

[11] Kumar, S, Luhandjula, M, Munapo, E, and Jones, B. "Fifty years of integer programming: a reviewofthesolutionapproaches".In:AsiaPacificBusinessReview6.3(2010),pp.5–15.

[12] Kumar Santosh, Munapo Elias, LesaoanaMaseka, NyamugurePhilimon. "Some innovations in or methodology: linear optimization", LAP LAMBERT Academic Publishing, 2018.

[13] Kumar, Santosh, Al-Hasani, Ali, Al-Rabeeah, Masar, and Eberhard, Andrew. "A random search methodforfinding'K2'numberofrankedoptimalsolutiontoa nassignmentproblem". Manuscript submitted for publication in the Proceeding of International Conference on Mathematical: Pure, Applied and Computation (ICoMPAC) 2019.

[14] Lu,BandRosenbaum,PR."AnAlgorithmforRankingalltheAssignmentsinOrderofIncreasing Cost".In:JournalofComputationalandGraphicalStatistics13.2(2004),pp.422–434.

[15] Murty, Katta G. "Letter to the editor: An algorithm for ranking all the assignments in order of increasing cost". In: Operations research 16.3 (1968), pp.682–687.

[16] Pascoal,Marta, Captivo,MEugénia,andClímaco,João."Anoteonanewvariantof Murtysranking assignments algorithm". In: Quarterly Journal of the Belgian, French and Italian Operations Research Societies 1.3 (2003), pp.243–255.

[17] Pedersen, Christian Roed, Nielsen, Lars Relund, and Andersen, Kim Allan. "An algorithm for rankingassignmentsusingreoptimization".In:Computers&OperationsResearch35.11(2008), pp.3714–3726.

[18] Schrage,LinusEandLINDOSystems,Inc. "OptimizationmodelingwithLINGO".DuxburyPress CA,1997.

[19] Taha, Hamdy A. Operations research: An introduction (for VTU). Pearson Education India, 2005.

Size Class	No. Class	Exact method CPU time(Sec.)	Z^1	Z^{2nd}	Z^{3rd}	Z^{4th}	Z^{5th}	Random search method CPU (Sec.)	Z^1	Z^{2nd}	Z^{3rd}	Z^{4th}	Z^{5th}
C16	C1	0.0078	16	21	22	25	28	0.0106	16	21	22	28	30
	C2	0.0012	30	31	32	35	38	0.0082	30	31	32	35	38
	C3	0.0052	36	41	42	44	45	0.0083	36	41	43	44	45
C36	C1	0.3286	24	25	28	29	31	0.0049	24	25	28	29	31
	C2	0.8	21	23	24	26	28	0.0056	21	23	24	26	28
	C3	0.72	31	36	37	38	39	0.0058	31	36	37	38	39
C64	C1	5.072	33	35	36	38	39	0.0061	33	35	36	38	39
	C2	6.023	27	28	29	30	31	0.0073	27	28	29	30	31
	C3	6.052	37	38	39	40	41	0.0084	37	38	39	41	42
C100	C1	15.086	35	36	37	38	39	0.0064	35	36	38	39	40
	C2	14.128	37	39	40	41	42	0.0071	37	39	40	42	43
	C3	13.087	26	29	30	31	32	0.0067	26	29	30	31	32
C200	C1	21.0234	35	36	37	38	39	0.0142	35	36	37	38	39
	C2	22.199	39	40	41	42	43	0.0135	39	40	41	42	43
	C3	22.186	38	40	41	42	43	0.0117	38	40	42	43	44

Table 14: Comparison Exact method and Random search method [13]

★★★