

# Efficient Cost Optimization using Transportation Mechanisms for Cloud Computing Environment

Narander Kumar, Parth Ranjan, Surendra Kumar



**Abstract:** Efficient computations are increasing now a day, so their need is very high in the world. Infrastructure and computation techniques are not as much as efficient in conventionally or in present scenario, therefore the cloud computing is new to deal this type of problems. Sequencing of hardware and software technologies, for giving scalable and low cost computational understandings in cloud computing. The major focus of this research is to diminish the transportation cost of resource allocation along with various virtual machines in cloud computing environment. In this research paper, implementation of Vogel's Approximation Method (VAM) to obtain an Initial Basic Feasible Solution (IBFS) and an algorithm to optimize the cost of resource transportations for cloud service provider (CSP) as well as present an example also to understand the proposed method for total supply values and total demand values. Although the calculation of cost reduction until the iteration still has a non-negative values, and the calculation is done again until the last iteration. A comparison has been shown the cost of the proposed mechanism is much less from other technique.

**Keywords :** Cloud Computing, Cloud Service Provider (CSP), Initial Basic Feasible Solution (IBFS), Resource Allocation, Vogel's Approximation Method (VAM)

## I. INTRODUCTION

Cloud computing is a modern and unique field rather than offering physical products of computing services are offered. Consumers are availed cost-effective services through shared resources. Services of various levels are provided to user for a varied spectrum of domains by cloud computing service providers [1]. Cloud Computing concepts are established on the concepts of parallel computing, distributed, and grid computing techniques. The key factors for popularity of cloud computing is cost-effective services while enhancement in scalability & flexibility. Complexity in cloud computing is increased because of high magnitude of data and high strength of users, along with the maintaining service continuity and optimal reliability. To strengthen cloud reliability, resource allocation mechanism enhances reliability in cloud and

lessens resource cost. Cloud Computing is a popular field of computer science because of its cost-effectiveness. Consequently any recommendation to boost any core metric of cloud computing without paying attention to associated cost diminishes the objective of cloud computing. The approach emphasizes on reducing cost and enhancing reliability within cloud computing [2].

There are multiple parameters associated, that are responsible for decision making of location and time for optimal allotment of data objects as well as services in cloud environment. Automatic placement and scheduling system is taken into narrative that the resource allotment is beneficial for stakeholder. The core hurdle is cost or performance optimization for available set of parameters, usually comprising one among the performance and price that is focused to optimization. The major constraints for enhancing cloud capabilities are Performance, Cost, Locality, Reliability and Continuous Availability. Due to variety of stationing schemes, several admissible parameters, a batch of checks and aimed tasks of potential interest, there are several challenges for the advancement of placement methods that are largely applicable. Primarily, no existence of any generic model to represent different schemes of resource scheduling, essentially when required credentials of user is ambiguous and difficult to encode via computer modeling languages. Large size problem is solved by suitable parameters for various tasks [3].

## II. RELATED WORK

CompVM (complementary VM) is an underlying designation instrument that integrates strength to corresponding VMs by worldly/spatial mindfulness. The term corresponding VMs are defined as the machines whose sheer interest is every resource measurement (within spatial space) almost arrives at the ability of its hosts in the lifetime period of VM (within worldly space). Presence of VM resources uses designs, the instrument foresees, the resources use examples of VMs [4]. Efficient transmission is for remaining tasks at hand of spatial inquiries of distinct sorts into reasonable virtual machines. Utilization of constant geospatial inquiry design learning philosophy in order to attain the framework's efficiency serving questions related to geospatial section. The properties which are application specific and the structures are organized that ought to aim on which kind of the geospatial inquiry to be designated for what sort of VM consequently [5].

Revised Manuscript Received on April 12, 2020.

\* Correspondence Author

**Narander Kumar\***, DCS, Babasaheb Bhimrao Ambedkar University, Lucknow, India. Email: nk\_iet@yahoo.co.in

**Parth Ranjan**, DCS, Babasaheb Bhimrao Ambedkar University, Lucknow, India. Email: parthranjan99@gmail.com

**Surendra Kumar**, DCS, Babasaheb Bhimrao Ambedkar University, Lucknow, India. Email: kumar.surendra1989@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

The system joins two general thoughts: agreement gauge that can dodge advertise control and yields an inexact ideal objective income with the accord gauge innovation. Also Revenue Extraction that can decide the victors and similarly shares the objective income created by agreement gauge among them with a solitary deal cost [6]. A progression of anticipating models for foreseeing interest to VMs in a cloud-based programming utilized as a SaaS. These models are then utilized in an intermittent survey arrangement model which decides what number of VMs ought to be provisioned or de-arrangement at every assessment interim [7]. Multi-occupant design of distributed computing, various applications have various prerequisites and needs. Dissatisfaction of moving information on time may cause a serious presentation fraud of an application in case of system clog inside a server farm. It might bring about a serious disappointment for a basic application that needs a specific degree of QoS fulfillment, in this way, an effective resources provisioning procedures are essential to guarantees moving the high-need information over different deals in arrange stumbling block [8]. The usage of host's handling capability gets enhanced by implementation of VM designation approach that leads to improvement in the span and throughput of cloud frame. The fundamental results are gathered by applying follow based origination in CloudSim 3.0.3 and analysis is done with the help of existing VM Distribution strategies [9]. To shield against such co-living arrangement assaults in Infrastructure as a Service (IaaS) through a secure, left behind task at hand accustomed and vitality productive Virtual Machine (VM) assignment methodology. In particular, model the VM task issue as a streamlining issue where the object is to bound: (1) The security risk presented by the co-living arrangement of VMs from different clients. (2) The general power utilization. (3) The imbalanced remaining load among a variety of physical servers. As this enhancement issue is NP-hard [10].

Vitality and Resource-Aware VM Placement (ERAP) computation is revealed. This calculation thinks about vitality just as focal handling unit (CPU) use to send the VMs on the servers. CloudSim toolbox is utilized to observe the conduct of the ERAP calculation. The viability of the ERAP calculation is tried on legitimate remaining task at hand hints of Planet Lab [11]. Analyzed the limitation mindful VM situation in varied processing bunches. A model of VM arrangement imperatives is displayed that backs necessities of all varieties among VMs and hosts. Secondly requirements of aware VM arrangement calculations are discussed that enhances the presentation for load adjusting or vitality sparing [12]. Exploring different avenues regarding a dynamic VM arrangement that believe diverse resources types (to be specific, CPU and memory). The planned arrangement make use of a inherited calculation for the dynamic reallocation of the VMs reliant on the real appeal of the individual VMs intending to bound under-use and over-usage situations in the cloud server farm [13]. Debasements of virtual machines (VMs) accessibility in distributed computing because of pernicious assaults. Reenactment result shows how stranded VMs gather after some time due to VM spillage and affected collection of resources. Few methods material is at different parts of the cloud to recognize and dispose of the stranded

VMs [14]. Calculations for the power-mindful designation what's more, movement of virtual machines. Power sparing is accomplished through power effective solidification of virtual machines on a more modest number of servers and by placing inactive hubs in resting mode [15].

Focal workplaces will helpful apportion cloud resources to store important substance also, fulfill client demands for those substances, in order to limit the framework costs regarding capacity, VM reconfiguration, content access idleness and substance relocation [16]. It requires on-request assignment of the physical mutual resources to individual VM according to their dynamic prerequisites to fulfill the SLA among client and cloud supplier. To do these proficiently future resources usage is anticipated utilizing fluffly rationale based forecast. To maintain a strategic distance from underestimation forecast mistakes because of spikes in the remaining task at hand. The anticipated qualities are cushioned with legitimate worth and quickly resources tops are raised [17]. In order to perform analysis dynamic power-sparing resources allocation (DPRA) system is reliant on calculations of molecule swarm expansion. The DPRA focuses on vitality utilization of physical machine (PM) and virtual machine (VM), in addition to it also handles vitality effectiveness proportion of climate control system [18]. The cloud-based satellite correspondence organizes the mechanisms to completely utilize resources in current satellite framework. Vitality Minimization is an essential issue in the C-SCN due to huge vitality cost and complicated calculations [19].

### III. MATHEMATICAL MODEL OF TRANSPORTATION PROBLEM:-

A unique category of linear programming problem is the transportation problem its objective is the minimization of transportation cost for a given commodity, from origin or source (factories, manufacturing units) to the destinations (warehouses, wholesale stores). Each origin has a limited supply (i.e. maximum count of products that can be shipped from it) on the other hand each destination has a demand to be satisfied (i.e. minimum count of products that can be shipped to it).

Notations used:-

$p$  = total count of sources ( $s = 1, 2, \dots, p$ )

$q$  = total count of destinations ( $d = 1, 2, \dots, q$ )

$w_{sd}$  = unit cost of shipping to be shipped from origin  $s$  to destination  $d$

$r_{sd}$  = amount to be shipped from origin  $s$  to destination  $d$

$k_s$  = supply at source  $s$

$l_d$  = demand at destination  $d$

Mathematical formulation of the transportation problem as a linear programming problem as below:

$$\text{Minimize } Y = \sum_{s=1}^p \sum_{d=1}^q w_{sd} r_{sd}$$

Subject to the following:-

$$\sum_{d=1}^q r_{sd} \leq k_s \text{ for } s = 1, 2,$$

.....,  $p$  (supply)

$$\sum_{s=1}^p r_{sd} \leq l_d \text{ for } d = 1, 2, \dots, q \text{ (demand)}$$

$$r_{sd} \geq 0$$

For existence of feasible solution it is important that total capacity equals to the total requirement.

If  $\sum_{s=1}^q k_s = \sum_{d=1}^p l_d$  i.e. if total supply = total demand then

it is considered as a Balanced Transportation Problem otherwise it is known as Unbalanced Transportation Problem.

**IV. PROPOSED ALGORITHM FOR COST OPTIMIZATION:-**

1. Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.
2. Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.
3. Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.
4. If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected.
5. Balance the supply and demand of the table. Cut out the satisfied row or column.
6. Steps 1 to 4 are repeated until the supply and demand of the full table i.e. for all rows and columns, becomes 0.
7. Formulate an IBFS using VAM.
8. From an unoccupied cell create a closed path (loop). The permission to make a 90 degree turn, in this loop is authorized only at two positions i.e. firstly at occupied cells and secondly at original unoccupied cells. Attach (+) and (-) sign at each corner in alternate manner, beginning from the original unoccupied cell.
9. The sum of transportation costs of each cell in the closed path is calculated which is known as Net Cost Change.
10. For every unoccupied cell, repeat steps 8 and 9.
11. If every net cost changes are  $\geq 0$ , an optimal solution is achieved. Terminate the algorithm.
12. If every net cost changes are not  $\geq 0$  then the unoccupied cell having the maximum negative net cost change is selected and then construction of a closed path is done.
13. Choose the least allocated value amongst the negative marked values on the closed path.
14. This value is assigned to the selected unoccupied cell (Now the unoccupied cell becomes an occupied cell).
15. This value is added to occupied cells marked with (+) sign.
16. This value is subtracted from occupied cells marked with (-) sign.
17. Repetition of Step-8 to Step-16 till an optimal solution is achieved. When Net Cost Change  $\geq 0$  for every unoccupied cell, the algorithm terminates.

**Degeneracy:-** A degenerate transportation problem is a problem when transportation table with m rows and n columns, does not satisfy the following condition:-  
 $m + n - 1 = \text{number of allocated cells.}$

**V. WORKING EXAMPLE**

In order to formulate Initial Basic Feasible Solution (IBFS) Vogel's Approximation Method (VAM) is implemented also

applying proposed algorithm for optimal solution. Let's take an example to understand the proposed phenomena for Total supply values are 5 and Total demand values are 5 [20].

**Table 1: Assignment Matrix for Transportation.**

	1	2	3	4	5	SUPPLY
A	3.25	1.2	5	2	4.4	4000
B	1.73	1.4	5.2	2.1	4.2	21455
C	2	1.4	4.8	2.35	4.15	1880
D	1.8	1.2	4.6	2.44	4.12	300
E	1.9	1.3	4.8	2.2	4	2410
DEMAND	9335	2180	2200	16120	210	

From the above Table 1 following data is concluded:

p = total count of sources = 5

q = total count of destinations = 5

$$\sum_{s=1}^q k_s = 30045$$

$$\sum_{d=1}^p l_d = 30045$$

$$\sum_{s=1}^q k_s = \sum_{d=1}^p l_d$$

i.e. Total Sum of Supply = Total Sum of Demand so it is a Balanced Transportation Problem.

STEP 1: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 2: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

**Table 2: Calculation of Row and Column Penalties.**

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	3.25	1.2	5	2	4.4	4000	2 - 1.2 = 0.8
B	1.73	1.4	5.2	2.1	4.2	21455	1.73 - 1.4 = 0.33
C	2	1.4	4.8	2.35	4.15	1880	2 - 1.4 = 0.6
D	1.8	1.2	4.6	2.44	4.12	300	1.8 - 1.2 = 0.6
E	1.9	1.3	4.8	2.2	4	2410	1.9 - 1.3 = 0.6
DEMAND	9335	2180	2200	16120	210		
COLUMN PENALTY	1.8 - 1.73 = 0.07	1.2 - 1.2 = 0	4.8 - 4.6 = 0.2	2.1 - 2 = 0.1	4.12 - 4 = 0.12		

STEP 3: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.

STEP 4: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As no draw occurs between the values of penalties, this step is ignored.



STEP 5: Balance the supply and demand of the table. Cut out the satisfied row or column. The above steps are performed and the following results are obtained and are displayed in

Table 3: -

Maximal penalty is 0.8, is present in row A. Minimal  $c_{ij}$  value in this row is  $c_{A2} = 1.2$ . Maximal allocation in cell  $c_{A2}$  is  $\min(4000, 2180) = 2180$ . Demand of column 2 is satisfied and the supply of row A adjusted from 4000 to 1820 ( $4000 - 2180 = 1820$ ).

STEP 6: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 7: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

Table-3: Allocating Value in cell  $c_{A2}$  min (4000, 2180) = 2180 and further calculating penalties.

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	3.25	<del>1.2</del> (2180)	5	2	4.4	1820	$3.25 - 2 = 1.25$
B	1.73	<del>1.4</del>	5.2	2.1	4.2	21455	$2.1 - 1.73 = 0.37$
C	2	<del>1.4</del>	4.8	2.35	4.15	1880	$2.35 - 2 = 0.35$
D	1.8	<del>1.2</del>	4.6	2.44	4.12	300	$2.44 - 1.8 = 0.64$
E	1.9	<del>1.3</del>	4.8	2.2	4	2410	$2.2 - 1.9 = 0.3$
DEMAND	9335	0	2200	16120	210		
COLUMN PENALTY	$1.8 - 1.73 = 0.07$	--	$4.8 - 4.6 = 0.2$	$2.1 - 2 = 0.1$	$4.12 - 4 = 0.12$		

STEP 8: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.

STEP 9: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As no draw occurs between the values of penalties, this step is ignored.

STEP 10: Balance the supply and demand of the table. Cut out the satisfied row or column.

The above steps are performed and the following results are obtained and are displayed in

Table 4: -

Maximal penalty is, 1.25, is present in row A. Minimal  $c_{ij}$  value in this row is  $c_{A4} = 2$ . Maximal allocation in cell  $c_{A4}$  is  $\min(1820, 16120) = 1820$ . Supply of row A is satisfied and the demand of column 4 is adjusted from 16120 to 14300 ( $16120 - 1820 = 14300$ ).

STEP 11: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 12: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

Table-4: Allocating Value in cell  $c_{A4}$  min (1820, 16120) = 1820 and further calculating penalties.

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	<del>3.25</del>	<del>1.2</del> (2180)	<del>5</del>	<del>2</del> (1820)	4.4	0	--
B	1.73	<del>1.4</del>	5.2	2.1	4.2	21455	$2.1 - 1.73 = 0.37$
C	2	<del>1.4</del>	4.8	2.35	4.15	1880	$2.35 - 2 = 0.35$
D	1.8	<del>1.2</del>	4.6	2.44	4.12	300	$2.44 - 1.8 = 0.64$
E	1.9	<del>1.3</del>	4.8	2.2	4	2410	$2.2 - 1.9 = 0.3$
DEMAND	9335	0	2200	14300	210		
COLUMN PENALTY	$1.8 - 1.73 = 0.07$	--	$4.8 - 4.6 = 0.2$	$2.2 - 2.1 = 0.1$	$4.12 - 4 = 0.12$		

STEP 13: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.

STEP 14: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As no draw occurs between the values of penalties, this step is ignored.

STEP 15: Balance the supply and demand of the table. Cut out the satisfied row or column.

The above steps are performed and the following results are obtained and are displayed in

Table 5: -

Maximal penalty is, 0.64, is present in row D. Minimal  $c_{ij}$  value in this row is  $c_{D1} = 1.8$ . Maximal allocation in cell  $c_{D1}$  is  $\min(300, 9335) = 300$ . Supply of row D is satisfied and demand of column 1 is adjusted from 9335 to 9035 ( $9335 - 300 = 9035$ ).

STEP 16: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 17: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

Table-5: Allocating Value in cell  $c_{D1}$  min (300, 9335) = 300 and further calculating penalties.

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	<del>3.25</del>	<del>1.2</del> (2180)	<del>5</del>	<del>2</del> (1820)	4.4	0	--
B	1.73	<del>1.4</del>	5.2	2.1	4.2	21455	$2.1 - 1.73 = 0.37$
C	2	<del>1.4</del>	4.8	2.35	4.15	1880	$2.35 - 2 = 0.35$
D	<del>1.8</del> (300)	<del>1.2</del>	4.6	2.44	4.12	0	--
E	1.9	<del>1.3</del>	4.8	2.2	4	2410	$2.2 - 1.9 = 0.3$
DEMAND	9035	0	2200	14300	210		
COLUMN PENALTY	$1.9 - 1.73 = 0.17$	--	$4.8 - 4.8 = 0$	$2.2 - 2.1 = 0.1$	$4.15 - 4 = 0.15$		

STEP 18: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost.

Perform the maximum allocation in this cell.

STEP 19: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As no draw occurs between the values of penalties, this step is ignored.

STEP 20: Balance the supply and demand of the table. Cut out the satisfied row or column.

The above steps are performed and the following results are obtained and are displayed in

Table 6: -

Maximal penalty is, 0.37, is present in row B. Minimal  $c_{ij}$  value in this row is  $c_{B1} = 1.73$ . Maximal allocation in cell  $c_{B1}$  is  $\min(21455, 9035) = 9035$ . Demand of column 1 is satisfied and the supply of B is adjusted from 21455 to 12420 ( $21455 - 9035 = 12420$ ).

STEP 21: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 22: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

**Table-6: Allocating Value in cell  $c_{B1}$  min (21455, 9035) = 9035 and further calculating penalties**

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	3.25	1.2(2180)	5	2(1820)	4.4	0	--
B	1.73(9035)	1.4	5.2	2.1	4.2	12420	4.2 - 2.1 = 2.1
C	2	1.4	4.8	2.35	4.15	1880	4.15 - 2.35 = 1.8
D	1.8(300)	1.2	4.6	2.44	4.12	0	--
E	1.9	1.3	4.8	2.2	4	2410	4 - 2.2 = 1.8
DEMAND	0	0	2200	14300	210		
COLUMN PENALTY	--	--	4.8 - 4.8 = 0	2.2 - 2.1 = 0.1	4.15 - 4 = 0.15		

STEP 23: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.

STEP 24: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As no draw occurs between the values of penalties, this step is ignored.

STEP 25: Balance the supply and demand of the table. Cut out the satisfied row or column.

The above steps are performed and the following results are obtained and are displayed in

Table 7: -

Maximal penalty is, 2.1, is present in row B. Minimal  $c_{ij}$  value in this row is  $c_{B4} = 2.1$ . Maximal allocation in cell  $c_{B4}$  is  $\min(12420, 14300) = 12420$ .

Supply of B is satisfied and the demand of 4 is adjusted from 14300 to 1880 ( $14300 - 12420 = 1880$ )

STEP 26: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 27: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

**Table-7: Allocating Value in cell  $c_{B4}$  min (12420, 14300) = 12420 and further calculating penalties.**

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	3.25	1.2(2180)	5	2(1820)	4.4	0	--
B	1.73(9035)	1.4	5.2	2.1(12420)	4.2	0	--
C	2	1.4	4.8	2.35	4.15	1880	4.15 - 2.35 = 1.8
D	1.8(300)	1.2	4.6	2.44	4.12	0	--
E	1.9	1.3	4.8	2.2	4	2410	4 - 2.2 = 1.8
DEMAND	0	0	2200	1880	210		
COLUMN PENALTY	--	--	4.8 - 4.8 = 0	2.35 - 2.2 = 0.15	4.15 - 4 = 0.15		

STEP 28: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.

STEP 29: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As draw occurs between the values of penalties of Row C and Row E, and the maximum allocated values are also same, Row C is chosen, as it is the top-most row.

STEP 30: Balance the supply and demand of the table. Cut out the satisfied row or column.

The above steps are performed and the following results are obtained and are displayed in

Table 8: -

Maximal penalty is, 1.8, is present in row C. Minimal  $c_{ij}$  value in this row is  $c_{C4} = 2.35$ . Maximal allocation in cell  $c_{C4}$  is  $\min(1880, 1880) = 1880$ . Supply of row C and demand of column 4 both are satisfied.

STEP 31: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 32: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

**Table-8: Allocating Value in cell  $c_{C4}$  min (1880, 1880) = 1880 and further calculating penalties.**

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	3.25	1.2(2180)	5	2(1820)	4.4	0	--
B	1.73(9035)	1.4	5.2	2.1(12420)	4.2	0	--
C	2	1.4	4.8	2.35(1880)	4.15	0	--
D	1.8(300)	1.2	4.6	2.44	4.12	0	--
E	1.9	1.3	4.8	2.2	4	2410	4.8 - 4 = 0.8
DEMAND	0	0	2200	0	210		
COLUMN PENALTY	--	--	4.8	--	4		

STEP 33: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.

STEP 34: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As no draw occurs between the values of penalties, this step is ignored.

STEP 35: Balance the supply and demand of the table. Cut out the satisfied row or column.

The above steps are performed and the following results are obtained and are displayed in

Table 9: -

Maximal penalty is, 4.8, is present in column 3. Minimal  $c_{ij}$  value in this column is  $c_{E3} = 4.8$ . Maximal allocation in cell  $c_{E3}$  is  $\min(2410, 2200) = 2200$ . Demand of column 3 is satisfied and the supply of row E is adjusted from 2410 to 210 ( $2410 - 2200 = 210$ ).

STEP 36: Row Operation: Subtract the least row cost from the next least row cost. This difference is known as ROW PENALTY.

STEP 37: Column Operation: Subtract the least column cost from the next least column cost. This difference is known as COLUMN PENALTY.

Table-9: Allocating Value in cell  $c_{E3}$  min (2410, 2200) = 2200 and further calculating penalties.

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	3.25	1.2(2180)	5	2(1820)	4.4	0	--
B	1.73(9035)	1.4	5.2	2.1(12420)	4.2	0	--
C	2	1.4	4.8	2.35(1880)	4.15	0	--
D	1.8(300)	1.2	4.6	2.44	4.12	0	--
E	1.9	1.3	4.8(2200)	2.2	4	210	4
DEMAND	0	0	0	0	210		
COLUMN PENALTY	--	--	--	--	4		

STEP 38: Pick out the highest row or column penalty, corresponding to this highest row or column penalty pick out the cell with minimal cost. Perform the maximum allocation in this cell.

STEP 39: If a draw occurs between the values of penalties, then the cell with the maximum allocation possible is selected. As no draw occurs between the values of penalties, this step is ignored.

STEP 40: Balance the supply and demand of the table. Cut out the satisfied row or column.

Maximal penalty is, 4, is present in row E. Minimal  $c_{ij}$  value in this row is  $c_{E5} = 4$ . Maximal allocation in cell  $c_{E5}$  is  $\min(210, 210) = 210$ . Supply of Row E and demand of Column 5 is satisfied.

Table-10 displays the list of all Row and Column Penalties and the allocations done.

Table-10: IBFS using VAM.

	1	2	3	4	5	SUPPLY	ROW PENALTY
A	3.25	1.2(2180)	5	2(1820)	4.4	4000	0.8   1.25   --   --
B	1.73(9035)	1.4	5.2	2.1(12420)	4.2	21455	0.33   0.37   --   --
C	2	1.4	4.8	2.35(1880)	4.15	1880	0.6   0.35   --   --
D	1.8(300)	1.2	4.6	2.44	4.12	300	0.35   0.35   --   --
E	1.9	1.3	4.8(2200)	2.2	4(210)	2410	1.8   1.8   --   --
DEMAND	9335	2180	2200	16120	210		0.6   0.64   --   --
COLUMN PENALTY	0.07	0	0.2	0.1	0.12		0.5   0.3   0.3
	0.07	--	0.2	0.1	0.12		0.3   1.8   1.8
	0.07	--	0.2	0.1	0.12		0.8   4
	0.17	--	0	0.1	0.15		
	--	--	0	0.1	0.15		
	--	--	0	0.15	0.15		
	--	--	4.8	--	4		
	--	--	--	--	4		

The minimum total transportation cost =  $1.2 \times 2180 + 2 \times 1820 + 1.73 \times 9035 + 2.1 \times 12420 + 2.35 \times 1880 + 1.8 \times 300 + 4.8 \times 2200 + 4 \times 210 = 64326.55$ .

Checking Degeneracy:-

Amount of allocated cells = 8, and applying formula  $m + n - 1 = 5 + 5 - 1 = 9$  (where  $m$ = total rows,  $n$ = total columns).

As  $m+n-1 =$  number of allocated cells is not satisfied i.e.  $8 < 9$

∴ it is a degenerated solution

for fixation of degeneracy, an artificial quantity (d) is used d is designated to vacant cell, with the least transportation cost.

Table-11: The quantity d is designated to E2, with the least transportation cost = 1.3

	1	2	3	4	5	SUPPLY
A	3.25	1.2(2180)	5	2(1820)	4.4	4000
B	1.73(9035)	1.4	5.2	2.1(12420)	4.2	21455
C	2	1.4	4.8	2.35(1880)	4.15	1880
D	1.8(300)	1.2	4.6	2.44	4.12	300
E	1.9	1.3(d)	4.8(2200)	2.2	4(210)	2410
DEMAND	9335	2180	2200	16120	210	

Applying proposed algorithm Method for optimality test.

Table-12: Applying proposed method on Allocation matrix.

	1	2	3	4	5	SUPPLY
A	3.25	1.2(2180)	5	2(1820)	4.4	4000
B	1.73(9035)	1.4	5.2	2.1(12420)	4.2	21455
C	2	1.4	4.8	2.35(1880)	4.15	1880
D	1.8(300)	1.2	4.6	2.44	4.12	300
E	1.9	1.3(d)	4.8(2200)	2.2	4(210)	2410
DEMAND	9335	2180	2200	16120	210	

STEP 41: From an unoccupied cell create a closed path (loop). The permission to make a 90 degree turn, in this loop is authorized only at two positions i.e. firstly at occupied cells and secondly at original unoccupied cells. Attach (+) and (-) sign at each corner in alternate manner, beginning from the original unoccupied cell.

STEP 42: In the closed path the sum of transportation costs of each cell is calculated and is known as Net Cost Change.

STEP 43: Repetition of STEP 41 and 42 is done for all unoccupied cells.

Table-13: Formation of closed loop for vacant cells.





VACANT CELL	LOOP	NET COST CHANGE
A1	A1→A4→B4→B1	3.25 - 2 + 2.1 - 1.73 = 1.62
A3	A3→A2→E2→E3	5 - 1.2 + 1.3 - 4.8 = 0.3
A5	A5→A2→E2→E5	4.4 - 1.2 + 1.3 - 4 = 0.5
B2	B2→B4→A4→A2	1.4 - 2.1 + 2 - 1.2 = 0.1
B3	B3→B4→A4→A2→E2→E3	5.2 - 2.1 + 2 - 1.2 + 1.3 - 4.8 = 0.4
B5	B5→B4→A4→A2→E2→E5	4.2 - 2.1 + 2 - 1.2 + 1.3 - 4 = 0.2
C1	C1→C4→B4→B1	2 - 2.35 + 2.1 - 1.73 = 0.02
C2	C2→C4→A4→A2	1.4 - 2.35 + 2 - 1.2 = -0.15
C3	C3→C4→A4→A2→E2→E3	4.8 - 2.35 + 2 - 1.2 + 1.3 - 4.8 = -0.25
C5	C5→C4→A4→A2→E2→E5	4.15 - 2.35 + 2 - 1.2 + 1.3 - 4 = -0.1
D2	D2→D1→B1→B4→A4→A2	1.2 - 1.8 + 1.73 - 2.1 + 2 - 1.2 = -0.17
D3	D3→D1→B1→B4→A4→A2→E2→E3	4.6 - 1.8 + 1.73 - 2.1 + 2 - 1.2 + 1.3 - 4.8 = -0.27
D4	D4→D1→B1→B4	2.44 - 1.8 + 1.73 - 2.1 = 0.27
D5	D5→D1→B1→B4→A4→A2→E2→E5	4.12 - 1.8 + 1.73 - 2.1 + 2 - 1.2 + 1.3 - 4 = 0.05
E1	E1→E2→A2→A4→B4→B1	1.9 - 1.3 + 1.2 - 2 + 2.1 - 1.73 = 0.17
E4	E4→E2→A2→A4	2.2 - 1.3 + 1.2 - 2 = 0.1

STEP 44: If every net cost changes are  $\geq 0$ , an optimal solution is achieved. Terminate the algorithm. As the net cost change for cells C2, C3, C5, D2 and D3 are negative, so the algorithm does not terminate.

STEP 45: The selection of an unallocated cell with maximum (-ve) net cost change is made and then construction of a closed path is done.

The selection of an unallocated cell with maximum (-ve) net cost change is made i.e. cell D3 = - 0.27 and form a closed path from D3.

Constructed closed path is  
D3→D1→B1→B4→A4→A2→E2→E3.

**Table-14: Closed path and (+ve) and (-ve) sign allocation for current vacant cell D3.**

	1	2	3	4	5	SUPPLY
A	3.25[1.62]	1.2 (2180) (-)	5[0.3]	2 (1820) (+)	4.4[0.5]	4000
B	1.73 (9035) (+)	1.4 [0.1]	5.2 [0.4]	2.1 (12420) (-)	4.2 [0.2]	21455
C	2 [0.02]	1.4 [-0.15]	4.8 [-0.25]	2.35 (1880)	4.15 [-0.1]	1880
D	1.8 (300) (-)	1.2 [-0.17]	4.6 [-0.27] (+)	2.44 [0.27]	4.12 [0.05]	300
E	1.9 [0.17]	1.3 (d) (+)	4.8 (2200) (-)	2.2 [0.1]	4 (210)	2410
DEMAND	9335	2180	2200	16120	210	

STEP 46: Choose the least allocated value amongst the negative marked values on the closed path.

Amongst all the negative position (-ve) on closed path the least allocated value is = 300

STEP 47: This value is assigned to the selected unoccupied cell (The unoccupied cell is now an occupied cell).

The value 300 is allocated to cell D3. Cell D3 is now an occupied cell.

STEP 48: This value is added to occupied cells marked with (+) sign.

STEP 49: This value is subtracted from occupied cells marked with (-) sign.

**Table-15: Subtract 300 from each (-ve) marked cell and add 300 to each (+ve) marked cell.**

	1	2	3	4	5	SUPPLY
A	3.25	1.2 (1880)	5	2 (2120)	4.4	4000
B	1.73 (9335)	1.4	5.2	2.1 (12120)	4.2	21455
C	2	1.4	4.8	2.35 (1880)	4.15	1880
D	1.8	1.2	4.6 (300)	2.44	4.12	300
E	1.9	1.3 (300)	4.8 (1900)	2.2	4 (210)	2410
DEMAND	9335	2180	2200	16120	210	

Checking Degeneracy:-

Amount of allocated cells = 9, and applied formula is  $m + n - 1 = 5 + 5 - 1 = 9$

(Where m= total rows, n= total columns).

As  $m + n - 1 =$  number of allocated cells is satisfied i.e.  $9 = 9$

∴ it is a non-degenerated solution.

STEP 50: From an unoccupied cell construct a closed path (loop). The permission to make a 90 degree turn, in this loop is authorized only at two positions i.e. firstly at occupied cells and secondly at original unoccupied cells. Attach (+) and (-) sign at each corner in alternate manner, beginning from the original unoccupied cell.

STEP 51: In the closed path the sum of transportation costs of each cell is calculated and is known as Net Cost Change.

STEP 52: Repetition of STEP 50 and 51 is done for all unoccupied cells.

**Table-16: Formation of closed loop for vacant cells.**

VACANT CELL	LOOP	NET COST CHANGE
A1	A1→A4→B4→B1	3.25 - 2 + 2.1 - 1.73 = 1.62
A3	A3→A2→E2→E3	5 - 1.2 + 1.3 - 4.8 = 0.3
A5	A5→A2→E2→E5	4.4 - 1.2 + 1.3 - 4 = 0.5
B2	B2→B4→A4→A2	1.4 - 2.1 + 2 - 1.2 = 0.1
B3	B3→B4→A4→A2→E2→E3	5.2 - 2.1 + 2 - 1.2 + 1.3 - 4.8 = 0.4
B5	B5→B4→A4→A2→E2→E5	4.2 - 2.1 + 2 - 1.2 + 1.3 - 4 = 0.2
C1	C1→C4→B4→B1	2 - 2.35 + 2.1 - 1.73 = 0.02
C2	C2→C4→A4→A2	1.4 - 2.35 + 2 - 1.2 = -0.15
C3	C3→C4→A4→A2→E2→E3	4.8 - 2.35 + 2 - 1.2 + 1.3 - 4.8 = -0.25
C5	C5→C4→A4→A2→E2→E5	4.15 - 2.35 + 2 - 1.2 + 1.3 - 4 = -0.1
D1	D1→D3→E3→E2→A2→A4→B4→B1	1.8 - 4.6 + 4.8 - 1.3 + 1.2 - 2 + 2.1 - 1.73 = 0.27
D2	D2→D3→E3→E2	1.2 - 4.6 + 4.8 - 1.3 = 0.1
D4	D4→D3→E3→E2→A2→A4	2.44 - 4.6 + 4.8 - 1.3 + 1.2 - 2 = 0.54
D5	D5→D3→E3→E5	4.12 - 4.6 + 4.8 - 4 = 0.32
E1	E1→E2→A2→A4→B4→B1	1.9 - 1.3 + 1.2 - 2 + 2.1 - 1.73 = 0.17
E4	E4→E2→A2→A4	2.2 - 1.3 + 1.2 - 2 = 0.1

STEP 53: If every net cost changes are  $\geq 0$ , an optimal solution is achieved. Terminate the algorithm. As for the cells C2, C3 and C5 the net cost change is negative, so the algorithm does not terminate.

STEP 54: The selection of an unallocated cell with maximum (-ve) net cost change is made and then construction of a closed path is done.

The cell which is vacant and with highest negative net cost change is selected i.e. cell C3 = - 0.25 and form a closed path from C3.

Constructed closed path is C3→C4→A4→A2→E2→E3

**Table-17: Closed path and (+ve) and (-ve) sign allocation for current vacant cell C3**

	1	2	3	4	5	SUPPLY
A	3.25 [1.62]	1.2 (1880) (-)	5 [0.3]	2 (2120) (+)	4.4[0.5]	4000
B	1.73 (9335)	1.4[0.1]	5.2 [0.4]	2.1 (12120)	4.2[0.2]	21455
C	2[0.02]	1.4 [-0.15]	4.8 [-0.25] (+)	2.35 (1880) (-)	4.15 [-0.1]	1880
D	1.8 [0.27]	1.2 [0.1]	4.6 (300)	2.44 [0.54]	4.12 [0.32]	300
E	1.9 [0.17]	1.3 (300) (+)	4.8 (1900) (-)	2.2 [0.1]	4 (210)	2410
DEMAND	9335	2180	2200	16120	210	

STEP 55: Choose the least allocated value amongst the negative marked values on the closed path.

Amongst all the negative position (-ve) on closed path the least allocated value is = 1880

STEP 56: This value is assigned to the selected unoccupied cell (The unoccupied cell is now an occupied cell).

The value 1880 is allocated to cell C3. Cell C3 is now an occupied cell.

STEP 57: This value is added to occupied cells marked with (+) sign.

STEP 58: This value is subtracted from occupied cells marked with (-) sign.

**Table-18: Subtract 1880 from each (-ve) marked cell and Add 1880 to each (+ve) marked cell**

	1	2	3	4	5	SUPPLY
A	3.25	1.2	5	2 (4000)	4.4	4000
B	1.73 (9335)	1.4	5.2	2.1 (12120)	4.2	21455
C	2	1.4	4.8 (1880)	2.35	4.15	1880
D	1.8	1.2	4.6 (300)	2.44	4.12	300
E	1.9	1.3 (2180)	4.8 (20)	2.2	4 (210)	2410
DEMAND	9335	2180	2200	16120	210	

Checking Degeneracy:-

Amount of allocated cells = 8, and applied formula is  $m + n$

$$- 1 = 5 + 5 - 1 = 9$$

(Where  $m$ = total rows,  $n$ = total columns).

As  $m+n-1$  = number of allocated cells is not satisfied i.e.  $8 < 9$

∴ it is a degenerated solution.

For fixation of degeneracy, an artificial quantity (d) is used d is designated to vacant cell, with the least transportation cost.

**Table-19: The quantity d is designated to A2, with the minimum transportation cost = 1.2**

	1	2	3	4	5	SUPPLY
A	3.25	1.2 (d)	5	2 (4000)	4.4	4000
B	1.73 (9335)	1.4	5.2	2.1 (12120)	4.2	21455
C	2	1.4	4.8 (1880)	2.35	4.15	1880
D	1.8	1.2	4.6 (300)	2.44	4.12	300
E	1.9	1.3 (2180)	4.8 (20)	2.2	4 (210)	2410
DEMAND	9335	2180	2200	16120	210	

STEP 59: From an unoccupied cell create a closed path (loop). The permission to make a 90 degree turn, in this loop is authorized only at two positions i.e. firstly at occupied cells and secondly at original unoccupied cells. Attach (+) and (-) sign at each corner in alternate manner, beginning from the original unoccupied cell.

STEP 60: In the closed path the sum of transportation costs of each cell is calculated and is known as Net Cost Change.

STEP 61: Repetition of STEP 59 and 60 is done for all unoccupied cells.

**Table-20: Formation of closed loop for vacant cells.**

VACANT CELL	LOOP	NET COST CHANGE
A1	A1→A4→B4→B1	3.25 - 2 + 2.1 - 1.73 = 1.62
A3	A3→A2→E2→E3	5 - 1.2 + 1.3 - 4.8 = 0.3
A5	A5→A2→E2→E5	4.4 - 1.2 + 1.3 - 4 = 0.5
B2	B2→B4→A4→A2	1.4 - 2.1 + 2 - 1.2 = 0.1
B3	B3→B4→A4→A2→E2→E3	5.2 - 2.1 + 2 - 1.2 + 1.3 - 4.8 = 0.4
B5	B5→B4→A4→A2→E2→E5	4.2 - 2.1 + 2 - 1.2 + 1.3 - 4 = 0.2
C1	C1→C3→E3→E2→A2→A4→B4→B1	2 - 4.8 + 4.8 - 1.3 + 1.2 - 2 + 2.1 - 1.73 = 0.27
C2	C2→C3→E3→E2	1.4 - 4.8 + 4.8 - 1.3 = 0.1
C4	C4→C3→E3→E2→A2→A4	2.35 - 4.8 + 4.8 - 1.3 + 1.2 - 2 = 0.25
C5	C5→C3→E3→E5	4.15 - 4.8 + 4.8 - 4 = 0.15
D1	D1→D3→E3→E2→A2→A4→B4→B1	1.8 - 4.6 + 4.8 - 1.3 + 1.2 - 2 + 2.1 - 1.73 = 0.27
D2	D2→D3→E3→E2	1.2 - 4.6 + 4.8 - 1.3 = 0.1
D4	D4→D3→E3→E2→A2→A4	2.44 - 4.6 + 4.8 - 1.3 + 1.2 - 2 = 0.54
D5	D5→D3→E3→E5	4.12 - 4.6 + 4.8 - 4 = 0.32
E1	E1→E2→A2→A4→B4→B1	1.9 - 1.3 + 1.2 - 2 + 2.1 - 1.73 = 0.17
E4	E4→E2→A2→A4	2.2 - 1.3 + 1.2 - 2 = 0.1



STEP 62: If every net cost changes are  $\geq 0$ , an optimal solution is achieved. Terminate the algorithm. As, entire net cost change  $\geq 0$ , Optimal solution is achieved.

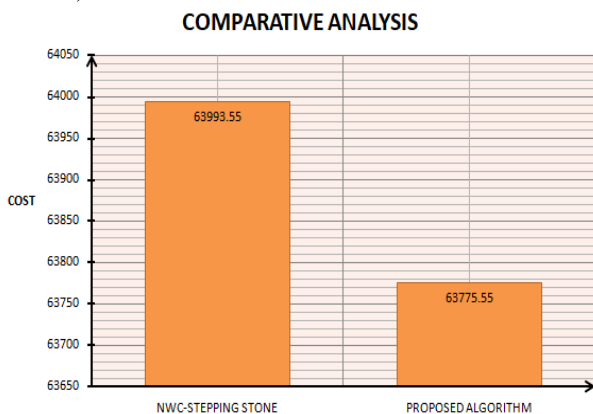
**Table-21: Final Optimal Solution Table.**

	1	2	3	4	5	SUPPLY
A	3.25	1.2 (d)	5	2 (4000)	4.4	4000
B	1.73 (9335)	1.4	5.2	2.1 (12120)	4.2	21455
C	2	1.4	4.8 (1880)	2.35	4.15	1880
D	1.8	1.2	4.6 (300)	2.44	4.12	300
E	1.9	1.3 (2180)	4.8 (20)	2.2	4 (210)	2410
DEMAND	9335	2180	2200	16120	210	

The minimum total transportation cost =  $2 \times 4000 + 1.73 \times 9335 + 2.1 \times 12120 + 4.8 \times 1880 + 4.6 \times 300 + 1.3 \times 2180 + 4.8 \times 20 + 4 \times 210 = 63775.55$

## VI. RESULT AND DISCUSSION

The study comprises of cost optimization and comparative analysis with a combined implementation of Vogel's Approximation Method and Stepping Stone method along with proposed algorithm. Vogel's Approximation Method is implemented to find the initial feasible solution and with the help of Stepping Stone method, the proposed algorithm to optimize the obtained initial feasible solution. After the implementation (as working example) of VAM along with proposed algorithm, the result/graph as displays in Figure 1, there is less transportation cost (i. e. 63775.55) in comparison with costs already obtained in NWC-Stepping Stone (i. e. 63993.55).



**Figure 1: Comparison of NWC-Stepping Stone and Proposed Algorithm**

## VII. CONCLUSION

The cloud computing is enable us to solve large and complex computational problems efficiently. In this context the concept of sharable resources and resource scheduling is adopted for working more efficiently with the cloud resources. When the cloud server resources are overloaded then these techniques are helps to minimize the processing cost and time by distributing the assigned load uniformly to all the set of resources. This will also help for the issues in which the resources are not managed properly. In this research paper we are determining the efficient cost under the transportation problem and the cost of purchase, delivery of services be optimized by using proposed approach. Although the

calculation of cost reduction until the iteration still has a non-negative values, and the calculation is done again until the last iteration obtained a total cost of \$63775.55 which is much less than any other algorithm from surveyed in literature.

## REFERENCES

- Rengasamy, R., and M. Chidambaram. "A Novel Predictive Resource Allocation Framework for Cloud Computing." 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 118-122. IEEE, 2019.
- Alam, ABM Bodrul, Mohammad Zulkernine, and Anwar Haque. "A reliability-based resource allocation approach for cloud computing." IEEE 7th International Symposium on Cloud and Service Computing (SC2), pp. 249-252. IEEE, 2017.
- Kumaraswamy, S. "Study of Virtual Machine Placement, its Parameters, Challenges, and State of the Art in Cloud Computing." International Journal of Advanced Computer Science and Technology, Vol. 6, No. 1, pp. 1-12, 2016.
- Shen, Haiying, and Liuhua Chen. "Compvm: A complementary vm allocation mechanism for cloud systems." IEEE/ACM Transactions on Networking (TON) 26, no. 3, pp- 1348-1361, 2018.
- Das, Jaydeep, Arindam Dasgupta, Soumya K. Ghosh, and Rajkumar Buyya. "A Learning Technique for VM Allocation to Resolve Geospatial Queries." In Recent Findings in Intelligent Computing Techniques, pp. 577-584. Springer, Singapore, 2019.
- Yang, Bo, Zhiyong Li, Shilong Jiang, and Keqin Li. "Envy-free auction mechanism for VM pricing and allocation in clouds." Future Generation Computer Systems Vol. 86, pp- 680-693, 2018.
- Bouterse, Brian, and Harry Perros. "Dynamic VM allocation in a SaaS environment." Annals of Telecommunications Vol. 73, no. 3-4, pp- 205-218, 2018.
- Son, Jungmin, and Rajkumar Buyya. "Priority-aware VM allocation and network bandwidth provisioning in software-defined networking (SDN)-enabled clouds." IEEE Transactions on Sustainable Computing, Vol. 4, no. 1, pp- 17-28, 2018.
- Banerjee, Sourav, Riman Mandal, and Utpal Biswas. "An Approach Towards Amelioration of an Efficient VM Allocation Policy in Cloud Computing Domain." Wireless Personal Communications, Vol. 98, no. 2, pp- 1799-1820, 2018.
- Liu, Yuhong, Xiaojun Ruan, Songjie Cai, Ruiwen Li, and Hanxiao He. "An optimized VM Allocation Strategy to Make a Secure and Energy-Efficient Cloud Against Co-residence Attack." IEEE International Conference on Computing, Networking and Communications (ICNC), pp. 349-353, 2018.
- Garg, Neha, Damanpreet Singh, and Major Singh Goraya. "Power and Resource-Aware VM Placement in Cloud Environment." IEEE 8th International Advance Computing Conference (IACC), pp. 113-118. IEEE, 2018.
- Kim, Seontae, and Young-ri Choi. "Effects of VM Placement Constraints in Heterogeneous Virtual Clusters." IEEE 3rd International Workshops on Foundations and Applications of Self\* Systems (FAS\* W), pp. 30-36, 2018.
- Mosa, Abdelkhalik, and Rizos Sakellariou. "Dynamic Virtual Machine Placement Considering CPU and Memory Resource Requirements." IEEE 12th International Conference on Cloud Computing (CLOUD), pp. 196-198, 2019.
- Bhan, Rati, Mir Shah Nawaz Ahmad, Mohit Jain, Aditya Singh, Rajendra Pamula, and Parvez Faruki. "VM Availability in Presence of Malicious Attacks in Open-Source Cloud." IEEE 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 26-30, 2019.
- Ajmera, Kashav, and Tribhuwan Kumar Tewari. "Greening the Cloud Through Power-Aware Virtual Machine Allocation." IEEE Eleventh International Conference on Contemporary Computing (IC3), pp. 1-6, 2018.
- Pu, Lingjun, Lei Jiao, Xu Chen, Lin Wang, Qinyi Xie, and Jingdong Xu. "Online resource allocation, content placement and request routing for cost-efficient edge caching in cloud radio access networks." IEEE Journal on Selected Areas in Communications, Vol. 36, no. 8, pp- 1751-1767, 2018.

17. Raghunath, Bane Raman, and B. Annappa. "Dynamic Resource Allocation Using Fuzzy Prediction System." IEEE 3rd International Conference for Convergence in Technology (I2CT), pp. 1-6., 2018.
18. Chou, Li-Der, Hui-Fan Chen, Fan-Hsun Tseng, Han-Chieh Chao, and Yao-Jen Chang. "DPRA: dynamic power-saving resource allocation for cloud data center using particle swarm optimization." IEEE Systems Journal, Vol. 12, no. 2, pp- 1554-1565, 2016.
19. Deng, Boyu, Chunxiao Jiang, and Song Guo. "Energy Minimization of Resource Allocation in Cloud-based Satellite Communication Networks." IEEE Communications Letters, Vol. 23, no. 12, pp- 2353-2356, 2019.
20. Ary, Maxi, And Asep Herman. "Basis Tree Approach and Nwc-Stepping Stone Method For Solving Transportation Problem With Fuzzy Cost", International Seminar on Scientific Issues and Trends, pp. 72-82, 2013.

## AUTHORS PROFILE



**Narander Kumar** received his Post Graduate degree and Ph. D. in CS & IT, from the Department of Computer Science and Information Technology, Faculty of Engineering and Technology, M.J.P. Rohilkhand University, Bareilly, Uttar Pradesh, INDIA in 2002 and 2009 respectively. His research interest includes Quality of Service (QoS), Computer Networks, resource management mechanism, in the networks for multimedia applications, performance evaluation.



**Parth Ranjan** received his B.Tech (Computer Engineering) degree from UIET, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow in 2018. Currently, He is Pursuing M.Tech (CS) degree in the Department of Computer Science at Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, UP, India. His research interest includes Cloud Computing, and Resource Management mechanism.



**Surendra Kumar** received his Master of Computer Application (M. C. A.) from Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India in 2013. Currently, He is a research scholar in the department of Computer Science at Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, UP, India. His research area is resource management in cloud computing environment.