International Journal of Scientific Research in ______________________________ **Research Paper**
Computer Science and Engineering
Vol.6, Issue.2, pp.76-81, April (2018) **E-ISSN:** 2320-7639

Comparative Study between FA, ACO, and PSO Algorithms for Optimizing Quadratic Assignment Problem

Vanita Jain[1]*, Aarushi Jain[2], Achin Jain[3], Arun Kumar Dubey[4]

[1,3,4]Bharati Vidyapeeth's College of Engineering, New Delhi, India
[2]FPM(Information Systems), IIM, Indore, India

[*]Corresponding Author: vanita.jain@bharatividyapeeth.edu

Available online at: www.isroset.org

Received: 08/Mar/2018, Revised: 20/Mar/2018, Accepted: 31/Mar/2018, Online: 30/Apr/ 2018

Abstract— Many advancements have been made in the development of optimization algorithms which are based on the evolutionary concept of behavior of biotic creatures like fish, ants, birds etc. This paper compares three such nature inspired algorithms; firefly, particle swarm optimization and ant colony optimization algorithm for optimizing Quadratic Assignment Problem. These algorithms are compared on the grounds of their time of computation of result, the no. of iterations required to solve the problem, and the accuracy .

Keywords— Quadratic Assignment Problem; Facility Location Problems; Firefly Algorithm; Ant Colony Optimization Algorithm; Particle Swarm Optimization

I. INTRODUCTION

There are many problems which are related to the positioning of some facilities such that the location for these facilities must be chosen to satisfy the demands of some commodities with minimum cost incurred [1]. The solution to these problems are required to estimate the number of facilities to be implanted in a region and the location of these facilities in that region. To satisfy the demands for the commodity completely, more number of facilities can be established. This will increase the cost of setting up the facilities. To reduce the establishment cost the number of facilities must be reduced, but this will in turn increase the transportation cost [2]. Hence, there are typically two measures that validates the effectiveness of this solution, these are, the service cost and the facility cost. The service cost is the total transportation cost of distribution of commodities between these facilities and from the facilities to the clients' locations. The facility cost is the total cost of establishing and locating all these facilities. These are the economic problems with great practical importance and are known as Facility Location Problems [3] which are encountered while determining the location to set up a manufacturing plant, warehouses, depots, storage facilities, hospitals, libraries, fire stations, base stations for mobile phone services, etc. [4] In such types of problems the customer locations and the alternative facility locations are known to be as discrete points in a plane. The demand for the commodity in these alternative locations is the pre- requisite knowledge for solving these problems. There are many variants to this problem. Sometimes a facility that can serve the number of clients is also fixed. This is called capacity of the facility [5].

Quadratic Assignment Problem (QAP) is a combinatorial assignment problem which finds its applications in concepts of engineering, operations research and economics [6]. The basic interpretation of a QAP can be understood properly by applying its formulations for the economic assignment of the set of activities to the set of locations [7]. Just like the facility location problem, Quadratic Assignment problem is also a NP-Hard problem [8]. Hence there is no such algorithm in polynomial time that can solve this problem [11]. As the size of the problem increases the amount of work that should be done to solve the problem increases exponentially. It was concluded by Burkard in [9] that the problem cannot be applied properly beyond the size of 20 [10]. Practically QAP is used to model many problems in the area like combinatorial data analysis, facility location problem, combinatorial optimization problem like travelling salesman problem, and many more [8].

Metaheuristic algorithms have been significant in the field of optimization algorithms, soft computing, and computational intelligence since they were introduced or generated. Swarm Intelligence (SI) based algorithms are subset of metaheuristic nature inspired algorithms. SI based algorithms are computed based on the swarm intelligence characteristics of birds, fishes, insects and other biological agents [16]. All three-metaheuristic nature inspired algorithms used in this paper are based on Swarm Intelligence.

The first algorithm is Firefly Algorithm inspired by fireflies. A firefly is a unique creature which has about two thousand distinct species in the world. These are hibernating insects which shows up in the spring season. They have a special characteristic of glowing in the dark. The Fireflies flash their light in short and rhythmic pattern either to attract their mating partner or to attract their potential prey. In the firefly algorithm this flashing behavior of these fireflies are studied and exploited to generate an optimization method for the objective function [17].

The pattern in which the ants behave in their colonies is the inspiring source of development of the second algorithm, that is, the Ant Colony Optimization Algorithm or ACO Algorithm. The ants use pheromones to communicate with other members of their colony. This pheromone is analogues to the information or details about the solution to the problem in question. The approach through which the ants finally computes the correct solution to their search forms the basis of ACO[18].

Particle Swarm Optimization(PSO) Algorithm is based on the iterative behavior of swarms of birds, insects or animals. Every living creature on this planet is constantly trying to get themselves a better life by being at a better place. They are always looking for a better solution to their problems. This pattern in which the complete swarm computes the best solutions out of their conditions is the basis Particle Swarm Optimization Algorithm [19].

The paper is organized in eight sections. Section II explains the Quadratic Assignment problem which is the problem used for the comparative study. Section III deals with brief description of the Firefly Algorithm. The Ant-Colony-Optimization Algorithm has been discussed in section IV. Section V elaborates the Particle-Swarm-Optimization Algorithm. Section VI discusses the methodology used in this paper. The results of the experimental work are shown in section VII and then finally section VIII concludes the paper.

II. QUADRATIC ASSIGNMENT PROBLEM

QAP is used to model multi row layout problem which has a fixed number of facilities for services of equal areas. A few examples of multi row layout problems are an office layout problem, a machine layout problem in an automated manufacturing system, and facility location problem. Travelling Salesman Problem can also be considered as the special case of QAP if it is assumed that all the facilities are connected by a single ring of flow. Similarly, many other combinatorial optimization problems can be written in the form of QAP with slight modifications. [11] The Quadratic Assignment Problem was first explained by Koomans and Beckman in 1957 for locating certain facilities for the flow of commodities [7]. In a facility location problem there are n locations and n facilities. This problem, to be formulated by a QAP, requires an equal number of locations and facilities. If the number of locations are more than the number of facilities, these extra locations are assigned the dummy facilities and the flow between these facilities is taken to be zero. If the number of locations are less than the number of facilities then this problem is not valid for QAP [12]. Mathematically, QAP is modeled using three n × n matrices.

- A→ [aik]→ the matrix showing distance between location i and k.
- B→ [bjl]→ the matrix showing the flow between facility j and l.
- C→ [cij]→ the cost of assignment of facility j to location i.

Normally, A and B are integer valued matrices. The assignment cost C between facility j to location I is considered negligible as it does not make any significant contribution to the complexity of the problem. For this situation a permutation π is used in which j = π (i) means a facility j is assigned to location i (i = 1, 2, …., n). The cost of transferring the data or the commodity can be represented as the product of the distance between the locations where the terms flow between two facilities. [13] QAP aims to minimize this total cost. Therefore, QAP is represented in its quadratic form as follows:

$$\text{Min } z = \sum_{i=1}^{N} \sum_{k=1}^{N} aik\ b\pi(i)\ \pi(k) \quad \pi\text{: permutation of } \{1 \ldots N\}$$

Here, N different permutations ensure that each condition is tested for the optimization [13]. Although the NP-Hard problems can have solutions for large dataset, QAP is an exception in this case. QAP instances of considerable size cannot be computed or calculated accurately. They will always get an approximate solution [14].

III. FIREFLY ALGORITHM

In late 2007-08 at Cambridge University. Firefly Algorithm was formulated by Xin-She Yang [16]. This algorithm is based on the behavior of fireflies and the rhythmic pattern of the lights emitted by them. Fireflies flashes their light through a process called bioluminescence. [20] The reason of different patterns of their light is used by them is not known truly but it is assumed that it is either used to attract a potential prey or to mate with their counter parts. It can also be used to warn the predators or the enemies about the bitter taste of the fireflies [21]. The Firefly Algorithm has three idealized rules or assumptions, which are:

- The fireflies are unisex, that is, every firefly is attracted to every other firefly.
- The brightness of a firefly is considered proportional to the attractiveness of that firefly. The less bright firefly is attracted towards the brighter firefly. If both the fireflies are of equal brightness than none of them is attracted to the other one. They will simply move in their random pattern.
- The brightness of a firefly is associated with its objective function [16].

The rhythm of the flash of light and the time until which the flashing continues judges whether the signal is for the prey or for mating. The intensity of light at a distance r can be calculated using the inverse square law as the intensity of light follows the law. This is the reason why the intensity of light reduces as the distance increases.
I0 ∝ 1/r2

According to the above equation, the intensity I0 decreases as the distance r increases. This is the reason why the firefly is not seen from the far-off distance. There are two important aspects in the Firefly Algorithm i.e. the attractiveness of the fireflies and the variation of light intensities. At location x the light intensity I(x) is directly proportional to the objective function at x. The attractiveness coefficient β is relative. Hence, to avoid the 0 in denominator error the light intensity and the attractiveness can be represented as [13].

$$I(r) = I0 \quad \text{And} \quad \beta = \beta_0 e^{-\gamma r^2}$$

The firefly which is less attractive moves towards the firefly j which is more attractive or brighter according to the following equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t \text{-} x_i^t\right) + \alpha_t \varepsilon_i^t$$

Here the second term is due to the attraction between the fireflies where β0 is the attractiveness of the firefly at the distance r = 0. The third term is randomization where α is the randomization parameter, and ϙ t i is a vector of random numbers drawn from a Gaussian distribution or uniform distribution at time t. [16, 21].

IV. ANT COLONY OPTIMIZATION

Ant-Colony-Optimization is a problem optimizing technique that uses artificial ants to find solutions to the problems and is based on the behavior of real ants which communicates through a chemical they possess called pheromone. This chemical acts as the information regarding the route an ant follows and the distance it has covered to reach its destination. Hence, this algorithm requires the past data and the distances between locations as a prerequisite to be implemented. When an ant finds food while roaming around randomly in the area it returns to its colony through a specific path leaving the trail of the pheromone behind. When the other ant of the colony discovers this trail, they follow the same path and leave its trail of pheromone of same quantity too on the same spot. But this pheromone evaporated too. Longer the ant will take to return to the same spot the lesser the effect of the trail becomes effective. Hence the shorter path between the colony and the food gets more renewal of trail than the longer path. The path with higher quantity of pheromone becomes more attractive than the route with the lower quantity of pheromone. In this way the shortest path or the optimized path is the one which attracts highest number of ants or which has the largest quantity of trail. [22, 23] The Ant Colony Optimization Algorithm is a meta-heuristic technique which was first applied on the travelling salesmen problem. [23] It is being applied on the Quadratic Assignment in this paper.

In the Ant Colony Optimization Algorithm each artificial agent which acts as an ant has many characteristics:

- When the ant chooses to assign facility j to location i is leaves a trace τij on the pair (i,j).
- It assigns the probability to each trace which is the function of the quality of the trace on the pair (i,j).
- To develop a total change, areas and exercises officially coupled are restrained until the point when the sum of what exercises have been allocated.

This algorithm construct solutions step by step, the use of a population of m agents which assigning an activity to each location[13]. The selection of the new route based on pheromone provides a mechanism to calculate the shortest path. The probability of an ant at node i to move from i to j is given by:

$$p_{ij} = \frac{\Phi_{ij}^{\alpha} d_{ij}^{\beta}}{\sum_{i,j=1}^{n} \Phi_{ij}^{\alpha} d_{ij}^{\beta}}$$

Here α and β are influence parameters, φij denotes the concentration of pheromone on the route between nodes i and j, and dij is the desirability of this route. For the selection of the shortest route some prerequisite knowledge is required, for example the distance between the node i and node j should be known before hand to know the value of the desirability of this route [21].

V. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart in 1995 worked on the Particle Swarm Optimization Algorithm [21]. It is implemented by maintaining a swarm of particles whose movement in the search space is influenced by the improvements discovered by the other particles [24]. Just like the Genetic Algorithms, PSO is also initialized with the population having random solutions. Each particle or the potential solution flows through the problem space with some random velocity assigned to it. Each potential solution or the particle records its coordinates of the best solution it has achieved so far. This coordinate is considered as the personal best or the *pbest* of the particle. Similarly, there is a different type of best which is the best solution of the overall problem. This is the *gbest* or global best of the problem. Now these particles flow towards a different solution with some different velocity provided to them. If this solution is better than the previous one than the new solution becomes the new *pbest.* The acceleration or the new velocity of the particle is decided according to the previous velocity [25].

The original, globally accepted version of PSO is implemented by the following process:

- The array of population or particles are initialized with random velocities and positions for the d that shows problem space dimension.
- The d variable of each particle evaluates desired fitness function.
- The particle's pbest is compared by fitness value. If current calculation is edged over pbest then set pbest value is equal to current and the pbest location equal to the current location in d-dimensional space.
- The overall fitness evaluation is compared with the population's overall previous best. Reset gbest to the current particle's array index and value if the current value is better than gbest, then
- The velocity and the position of the particles are changed according to the following formulae:

$$v_i^{t+1} = v_i^t + \alpha\varepsilon_1[g* - x_i^t] + \beta\varepsilon_2[x*_i - x_i^t]$$
$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

Here ε_1 and ε_2 are two random vectors whose values lie between 0 and 1. Parameters α and β determines acceleration. [21, 25].

VI. METHODOLOGY

Quadratic Assignment problems are on the most difficult problem to solve in the class of NP – Hard problems. One of the most commonly used approach that has been used is Branch and Bound based algorithms, but the main concern in using these algorithms is that we must limit the size of the space that needs to be searched [26]. In this paper we have tried to solve these problems in polynomial time by optimizing the algorithms using nature inspiring algorithms and then find the closest possible solution.

The dataset for the problem has been obtained from [15]. The data describe the distances of 12 different facilities of a hospital and the flow of patients between those facilities. The optimal solutions to this problem has been found by J.Clausen and M.Perregaard using parallel Branch and Bound based algorithms. The optimal solution to the above dataset was found as 289 [15].

In this paper the algorithms are provided with 1000 iterations. They are compared based on three parameters.

a) time taken by the algorithm to compute the correct result

b) percentage error in the result computed by each algorithm, and

c) time taken by these algorithms to complete these 1000 iterations.

We implemented these algorithms and conducted our experiments on MATLAB R2013a. In each of the algorithm population size is taken as 50. The program continues to run till the maximum number of iterations. The experiment has been done for 100 times and results are considered on an average basis.

To solve the problem using Ant Colony optimization, initial pheromone (τ) has been taken as 10 with pheromone exponential weight (α) as 0.3 and evaporation rate (ρ) as 0.1. The pheromone levels were updated with each iteration to ensure that ants follow the path having larger quantity of pheromone.

The parameters used in Firefly algorithm are the Light absorption coefficient (χ) taken as 1, attraction coefficient base value (β) as 2 and mutual coefficient as 0.2.

To solve the problem using PSO, the learning coefficient c1 and c2 should be between 1 and 4, so it was considered as personal learning coefficient (c1) as 1.5 and global learning coefficient (c2) as 2.

VII. RESULTS

Nature inspiring algorithms explores the search space randomly by improving the cost function to the best value, hence they are faster in execution but do not guarantee optimal solution and results in near optimal solutions [21]. Since they are random search based algorithm, thus every time they may give a different answer. The correct answer to the problem statement is 289 [15]. As an example, in one the runs - The Firefly Algorithm started with 334 in its first iteration and then improved the result in very few iterations to obtain the correct result without any gaps. Particle Swarm Optimization started with 345 and on optimization changed its answer to 299 in an early stage but unfortunately didn't reach to the exact answer. The Ant Colony Optimization Algorithm started with 346, improved the cost after every few iterations and finally finished at a best cost of 307. The graph and iterations results obtained are shown in Figure 1 and Table 1 respectively.

Table 1 Iterations results for PSO, ACO, FA

Algorithm	Initial Value	Final value
FA	334	289
PSO	345	299
ACO	346	307

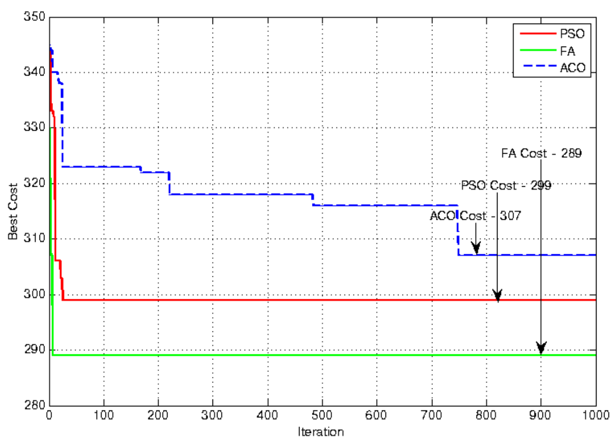

Figure 1. Best Cost vs Iterations results for PSO, ACO and FA

The difference between the cost obtained from these algorithms with the optimized cost is represented as the percentage error, calculated using the below equation and results are given in Table 2.

%Error = 100 * (Obtained cost - best cost/ Best Cost)

Table 2 Percentage errors for PSO, ACO and FA

Algorithm	Optimized Cost	%Error
FA	289	0%
PSO	299	3.46%
ACO	307	6.22%

The three algorithms are also compared based on the time taken to complete the iterations. PSO was completed in 37.141 seconds, ACO took 78.819 seconds, while FA took just 17.540 seconds which is fastest among the others. The self-time represent the time spent to execute the core algorithm code whereas total time is the summation of self-time and time used to perform MATLAB functions. Results are shown in Table 3.

Table 3 Time analysis of PSO, ACO and FA

Algorithm	Self-Time (sec)	Total time(sec)
FA	10.2	17.3
PSO	20.2	37.1
ACO	40.8	78.8

VIII. CONCLUSION

The three nature inspired algorithms namely Firefly Algorithm, Ant Colony Optimization Algorithm, and Particle Swarm Optimization have been implemented for a Quadratic Assignment problem. Through this implementation of algorithms on a dataset we have done the comparative study of these algorithms on the ground of accuracy, time of computation of result, and the number of iterations required for the accurate result. The Firefly Algorithm came out to be the best algorithm among all three. It calculated the exact result in the minimum time and used minimum iterations. The PSO and the ACO keeps on swapping their positions between average to worst resulting algorithms of the three. PSO uses less number of iteration than ACO and requires less time. ACO might give more accurate result than PSO but takes more time and iterations for the computation of the result.

REFERENCES

[1] Guha, S., & Khuller, S, "*Greedy Strikes Back: Improved Facility Location Algorithms*" Journal of Algorithms, 31(1), 228–248. (1999)

[2] Wolsey, Laurence A,.et. al. "*The Uncapacitated Facility Location Problem*", CM university, Pittsburgh, management science research rep., (1983).

[3] Arya, V., Arya, V., Garg, N., Garg, N., Khandekar, R., Khandekar, R., Pandit, V., "*Local search heuristic for k-median and facility location problems*", Proceedings of the Thirty-Third Annual {ACM} Symposium on Theory of Computing, 33(3), 21–29, (2001)

[4] Vygen, Jens, "*Approximation algorithms facility location problems*" Forschungsinstitut für Diskrete Mathematik, Rheinische Friedrich-Wilhelms-Universität, 1–59, 2005

[5] Verter, Vedat. "*Uncapacitated and capacitated facility location problems*" Foundations of Location Analysis. Springer US, (2011), 25-37.

[6] Choi, Darwin., "*Quadratic Assignment Problems (QAP) and its Size Reduction Method*" Math Journal 4(1), (2003).

[7] P. Pardalos, F. Rendl, and H. Wolkowicz, "*The quadratic assignment problem: A survey and recent developments*" DIMACS Series in Discrete Mathematics and Theoretical Computer Science 16, (1994), 1–42

[8] Eliane Maria,.et..al., "*A survey for the quadratic assignment problem*," European Journal of Operational Research, vol. 176, (2007), pp. 657–690

[9] R.E. Burkard, "*Quadratic Assignment Problems*", EJOR 15, (1984), 283-289.

[10] Taillard, Éric. "*Robust taboo search for the quadratic assignment problem*", Parallel computing 17(4), (1991), pp.443-455.

[11] Azarbonyad, Hosein, and Reza Babazadeh. "*A Genetic Algorithm for solving Quadratic Assignment Problem (QAP)*" ,arXiv preprint arXiv:1405.5050, (2014).

[12] Heragu S., "*Facilities Design*", Boston, PWS Publishing company, (1997).

[13] Maniezzo, Vittorio, and Alberto Colorni. "*The ant system applied to the quadratic assignment problem*" Knowledge and Data Engineering, IEEE Transactions on 11(5), (1999): 769-778.

[14] Gambardella, L. M., ÉD Taillard, and M. Dorigo. "*Ant Colonies for the Quadratic Assignment Problem*", The Journal of the Operational Research Society 50(2), (1999): 167–176.

[15] Burkard, Rainer E., Stefan E. Karisch, and Franz Rendl. "*QAPLIB–a quadratic assignment problem library."* ,Journal of Global optimization 10.4 (1997): 391-403.

[16] Yang, Xin-She, and Xingshi He., "*Firefly algorithm: recent advances and applications*", International Journal of Swarm Intelligence 1(1), (2013): 36-50.

[17] Yang, Xin-She. "*Firefly algorithm, Levy flights and global optimization*", Research and development in intelligent systems XXVI. Springer London, (2010), 209-218.

[18] Dorigo, M., & Stu¨tzle, T., "*The ant colony optimization metaheuristic: Algorithms, applications and advances*", In F. Glover & G. Kochenberger (Eds.), Handbook of Metaheuristics, vol. 57 (2002).

[19] Venter, Gerhard, and Jaroslaw Sobieszczanski-Sobieski. "*Particle swarm optimization*" American Institute of Aeronautics and Astronautics Journal ,41(8), (2003): 1583-1589.

[20] Bharat Bhushan and Sarath S. Pillai, "*Particle Swarm Optimization and Firefly Algorithm: Performance Analysis*", IEEE International Advance Computing Conference, (2013).

[21] X.-S. Yang, "*Analysis of Algorithms*," in Nature-Inspired Optimization Algorithms, 1sted. London, Elsevier, (2014), pp.15-16

[22] S. Agarwal, "*Edge Detection in blurred images using Ant Colony Optimization Techniques*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.2, pp.21-24, 2013

[23] J. E. Bell and P. R. McMullen, "*Ant colony optimization techniques for the vehicle routing problem*," Adv. Eng. Informatics, 18(1), (2004), pp. 41–48.

[24] Pedersen, Magnus Erik Hvass, and Andrew J. Chipperfield. "*Simplifying particle swarm optimization*" Applied Soft Computing 10(2), (2010): 618-628.

[25] Russell C. Eberhart and Yuhui Shi, "*Particle swarm optimization: Developments, applications and resources*," in Proc. IEEE Int. Conf. Evolutionary Computation, 01, (2001) pp. 81-86.

[26] Jens Clausen, Michael Perregaard, "*Solving Large Quadratic Assignment Problems in Parallel*", Computational Optimization and Applications, September 1997, 8(2), pp 111-127

Authors Profile

Vanita Jain received Ph.D. from V.J.T.I.,Mumbai, India , M.Tech. from National Institute of Technology, Kurukshetra and B.E. degreen from Punjab Engineering College, Chandigarh, India. From 1998 to 1999, she worked at National Institute of Technology, Kurukshetra as a lecturer. She joined Thadomal Shahani Engineering College, Bandra, Mumbai and served in the Department of Electronics & Telecommunication Engineering for 15 years. Currently, she is Professor and Dean (Academics) at Bharati Vidyapeeth's College of Engineering, New Delhi. She is having more than 28 years of teaching experience. Her field of interest includes Soft Computing, Control, Optimization Techniques and System Engineering and is actively involved in the research in these areas.

Aarushi Jain is currently pursuing Ph.D. from IIM Indore in the discipline of Information Systems. She has completed B.Tech. in Computer Science and Engineering in 2015. She has worked as Software Development Engineer, Associate for 2 years in Fiserv India Pvt. Ltd (A US financial services MNC). Her area of interests is Optimization Techniques, Social Media Analytics, Search Engine Optimization.

Achin Jain, is working as Assistant Professor in Bharati Vidyapeeth College of Engineering, New Delhi in the Information Technology Department. He received M.Tech. (Computer Science and Technology) & B.Tech(Information Technology) from the Guru Gobind Singh Indraprastha University, Delhi. He has rich experience in teaching B.Tech students and has published more than 6 Research Papers in International Journals and Conferences. His area of interest includes Web Usage Mining, Web Attacks.

Arun Kumar Dubey is working as an Assistant Professor in Bharati Vidyapeeth College of Engineering, New Delhi in the Information Technology Department. He has done B.Tech in Computer Science and Engineering , M.Tech in Computer Technology and Application and currently pursuing Ph.D from the Guru Gobind Singh Indraprastha University, Delhi. He has rich experience in teaching B.Tech students and has published more than 6 Research Papers in International Journals and Conferences. His area of interest Big Data,Image and Video Data Analysis.