Identification of influential users in social network using optimization algorithms

Amrit Mohapatra-222IT004   
*Information Technology  
National Institute of Technology, Karnataka,Surathkal, India 575025*[amrtimohapatra.222it004@nitk.edu.in](mailto:amrtimohapatra.222it004@nitk.edu.in)

Natasha Jain-222IT023  
*Information Technology  
National Institute of Technolgy, Karnataka,Surathkal, India 575025*[natashajain.222it023@nitk.edu.in](mailto:natashajain.222it023@nitk.edu.in)

Dr. Sowmya Kamath S  
*Information Technology  
National Institute of Technology, Karnataka,Surathkal, India 575025* sowmyakamath@nitk.edu.in

*Abstract*—When it comes to viral marketing, one of the tasks that is considered to be among the most difficult is to successfully identify a group of influential people. By sending one's promotional messages to this specific group, one can make contact with the greatest possible portion of the network's population. In this piece, we take the issue of maximizing one's influence and reframe it as a problem involving optimization. The influentiality of the nodes and the distance between them serve as the corresponding cost functions in this problem. By creating as large of a physical distance as feasible between the seed nodes, it is possible to guarantee that all of the network's nodes will be easily accessible. We implement the gray wolf optimization strategy, the particle swarm optimization technique, and the genetic algorithm optimization technique so that we may resolve the issue , and then we compare the fitness values of each of these methods. The findings of our studies, which were carried out on three different live networks, indicate that the suggested methods outperforms the most cutting-edge impact maximization methods currently available. In addition to this, the amount of time it takes to complete a computation using it is noticeably shorter than that required by alternative meta-heuristic methods.

Keywords— influentiality, optimization, gray wolf optimization, particle swarm optimization, genetic algorithm optimization, fitness, meta-heuristic

# Introduction

In many fields, including marketing, politics, and the study of public opinion, social networks have evolved into an indispensable resource for the spread of information and the exchange of ideas. It is essential to understand the dynamics of social networks in order to build efficient tactics for targeted marketing or opinion campaigns. One way to do this is to identify prominent members in social networks. A number of different approaches, such as centrality measurements and optimization algorithms, have been offered as potential means of locating significant users.

A group of users, also known as a seed set, is chosen to begin spreading the messages around the network in order to take advantage of the features offered by online social networks, which can be of great help to viral marketing. The number of people who are able to join the seed is restricted because companies only have a certain amount of money to spend on advertising. The dilemma that emerges here is how to most effectively identify the users who have the most influence so that you can include them in the seed set. This topic is referred to as the Influence Maximization (IM) problem, and its definition is to select a group of k users as the starting point for the process of propagating an idea.

In certain approaches, users' behavior and interests are essential elements in spreading, although real networks don't always have such information. Hence, we typically just have network structural information to identify influential users. Diffusion models have been devised to mimic spreading and quantify seed set influence. Assessing all alternative subsets to identify the ideal seed set is NP-hard due to the vast search space. Diffusion models might take time to measure user influence in big networks. Selecting central nodes may help identify influential users. Degree, betweenness, and K-shell assess structural centrality. Structural centrality measurements rarely yield near-optimal solutions. Several works have employed greedy, heuristic, and meta-heuristic methods to tackle IM problems as optimization problems. A diffusion model selects a greedy solution near to the optimal solution. They find accurate solutions, but their computational complexity makes them unfeasible for vast networks. IM problem heuristics reduce computing complexity but reduce accuracy. They may also trap in local optima. Recent evolutionary optimization methods prevent this. These studies found a near-optimal solution. One must first effectively assess user influence and then discover the most influential users to find a near-optimal group of influential people. We initially measure user influence using an entropy-based metric. We tackle the optimization problem using gray wolf optimization. This evolutionary optimization approach features fewer adjustable parameters, reduced computing complexity, and faster convergence than others.

The purpose of this study is to evaluate and contrast three different optimization algorithms—Gray Wolf Optimization (GWO), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA)—in order to determine which one is the most effective at locating influential users in online social networks. PSO and GA are both swarm-based optimization algorithms that were inspired by the collective behavior of birds and genetic evolution, respectively. The GWO algorithm is a meta-heuristic algorithm that was inspired by the hunting behavior of gray wolves.

The purpose of this research is to assess the efficacy of these optimization algorithms in locating prominent users inside social networks and to determine which of these algorithms are most effective. Centrality was measured in a number of different ways throughout the course of this investigation, including degree centrality, betweenness centrality, and eigenvector centrality.

The remaining parts of the paper are structured as described below. In section II, a literature review is presented on the many strategies that are currently available for locating prominent members in social networks. In Part III, the technique that was utilized in this research is discussed in detail, including the optimization methods and centrality measurements. The findings of the experiment and its analysis are presented in Section IV. The final section, Section V, summarizes the article and provides some suggestions for further research.

In general, this research makes a contribution to the field of social network analysis by analyzing the efficacy and productivity of several optimization algorithms for the purpose of locating prominent users in social networks. The results of this study can provide researchers and practitioner with useful information for developing effective tactics for targeted marketing or opinion initiatives.

# RELATED WORK

Identifying prominent users in social networks has been the subject of a number of studies, which have utilized a variety of approaches. Measures of centrality, such as degree centrality, [1] betweenness centrality[2], and eigenvector centrality, are frequently applied in the process of locating influential users on the basis of their locations in the network. It has been demonstrated that these measurements are successful in identifying prominent users in social networks; nevertheless, they do have certain limitations, such as the inability to take into consideration dynamic changes in the network. Optimization algorithms have been presented as a solution to solve these restrictions. These algorithms seek to identify prominent users based on the centrality measurements they possess and the architecture of the network. The identification of prominent people in social networks has been the subject of a number of research, and optimization techniques including PSO, GA, and ACO have been applied. When compared to measures of centrality, these algorithms have demonstrated findings that are encouraging in terms of their accuracy and efficiency.

PSO is a well-known optimization technique that takes its cues from the cooperative actions of flocks of birds. Adjusting the position and velocity of particles in the solution space on an iterative basis while the algorithm looks for the best possible solution is one of the methods that it uses. PSO [3] has been employed in a number of research to identify significant users in social networks, and the findings of these investigations have showed promise.

Another optimization algorithm that draws its inspiration from genetic evolution is called GA. The algorithm[4] imitates the process of natural selection by repeatedly generating new populations and picking the most fit people to represent the most successful individuals in each generation. GA has been utilized in a variety of studies to identify prominent users in social networks, and it has demonstrated results that are comparable to those of PSO in this regard.

The foraging activity of ants served as the inspiration for the optimization method known as ACO[5]. The algorithm performs repeated adjustments to the pheromone trail and the probability of selecting a path in order to locate the best possible answer to the problem. When compared to PSO and GA, the outcomes of studies in which ACO was utilized to identify prominent members in social networks were found to be less promising. Nonetheless, ACO has been used in some of these studies.

GWO is a relatively new optimization algorithm that was just developed. It was conceptualized after the hunting strategy of grey wolves. The program recreates the hunting behavior of a pack of wolves and looks for the best possible solution by rearranging the positions of the pack's alpha, beta, and delta wolves. Although GWO has been used in a few studies to identify prominent people in social networks and has showed promising results from those applications, the extent to which it is useful and efficient has not been completely researched.

These research have demonstrated that optimization algorithms are capable, based on centrality metrics and network architecture, of effectively identifying prominent members inside social networks. However, the efficacy and efficiency of these algorithms is contingent on the particular attributes of the optimization algorithm as well as the social network. In this investigation, we evaluate and contrast the efficacy of PSO, GA, and GWO in terms of locating prominent users in social networks.

|  |  |  |  |
| --- | --- | --- | --- |
| Author/year | Methodology | Advantages | Limitations |
| Ahmad Zareie, Amir , Sheikhahma  ,Keyhan Khamforoosh  Year :2018 [6] | Selection of the set of influential users using the Technique for Order of Preference by Similarity to Ideal Solution method | The approach outperforms existing influence maximization methods .It provides a systematic and objective way to evaluate the influence of nodes based on multiple criteria. | The paper assumes that the influence of a node is solely determined by its topological features .The proposed approach is computationally expensive |
| Yu Wang†, Gao Cong, Guojie Song, Kunqing Xie  Year:2010 [7] | Community-based greedy algorithm for identifying top-k influential nodes | It considers both global and local network structures, which can capture the effects of both within and between-community interactions on node influence. | The evaluation is limited to a specific type of mobile social network and it is unclear whether the proposed algorithm would be effective in other types of mobile social networks. |
| Laizhong Cui a  Huaixiong Hua, Shui Yu b  Qiao Yanet.al  Year :2018 [8] | DDSE (Degree-Descending Search Evolutionary),for influence maximization in social networks. | Employs a degree-descending search strategy, which prioritizes high-degree nodes, and a population-based evolutionary approach, which can efficiently explore the search space | The proposed algorithm requires setting several parameters, which may be challenging for users with limited expertise in evolutionary algorithms |

Table1. Literature Survey

# DATASET

To evaluate the performance of the GWO, PSO, and GA-based influential maximization algorithms, we used three real-world networks: Hamsterster full (HAM), Pretty Good Privacy (PGP), and Astro (AST). Table 1 provides the details of each network, including the number of nodes, edges, average degree, and activation probability used in the Independent Cascade Model (ICM) diffusion process.

1. **The Hamsterster ful**l network is a social network of a web community that allows users to create profiles and interact with each other. It contains 2426 nodes and 16,631 edges, with an average degree of 13.711. We set the activation probability for the ICM diffusion process to 0.03.

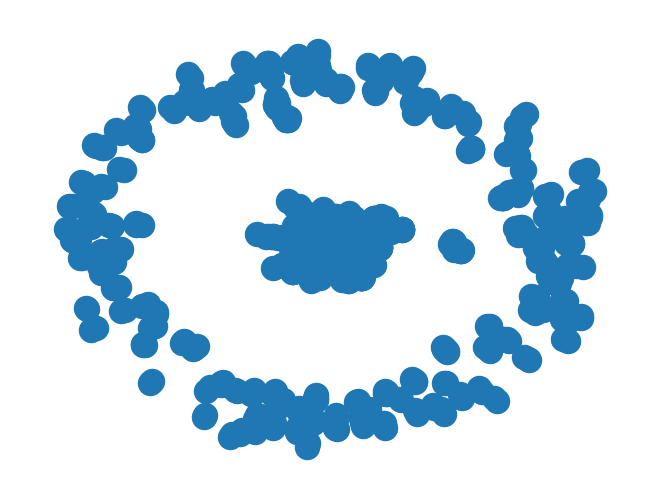


Fig 1: Graph Representation of hamster full dataset

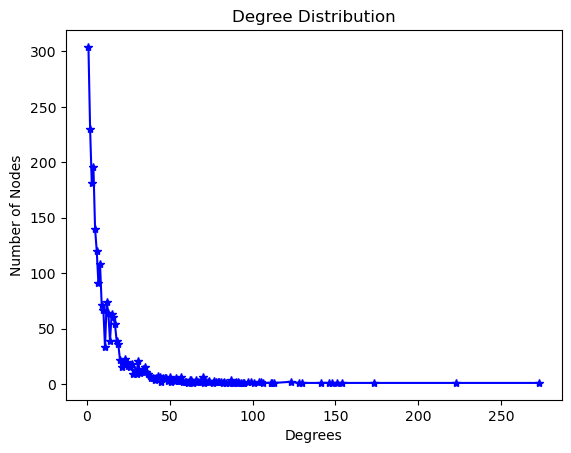


Fig 2: Degree Distribution of the nodes

The average path length of the graph is 1.40729. The density of graph is 0.005653. Average clustering Coefficient is 0.5376 and average diameter is 10

1. The **Pretty Good Privacy network** is a network of encrypted email users, which contains 10,680 nodes and 24,316 edges, with an average degree of 4.5536. We set the activation probability for the ICM diffusion process to 0.06.



Fig 3: Graph Representation Pretty Good Privacy network dataset

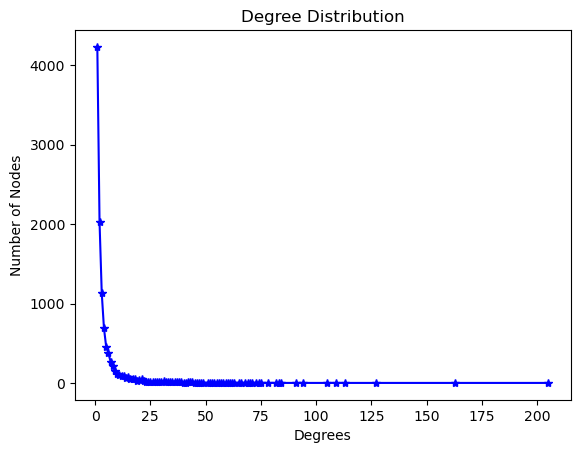


Fig 4: Degree Distribution of the nodes

The average path length of the graph is 7.48554. The density of graph is 0.0004264. Average clustering Coefficient is 0.26594 and average diameter is 24.

1. The **Astro** network is a co-authorship network of papers in the astro-ph archive of arXiv, which contains 18,771 nodes and 198,050 edges, with an average degree of 21.102. We set the activation probability for the ICM diffusion process to 0.02.

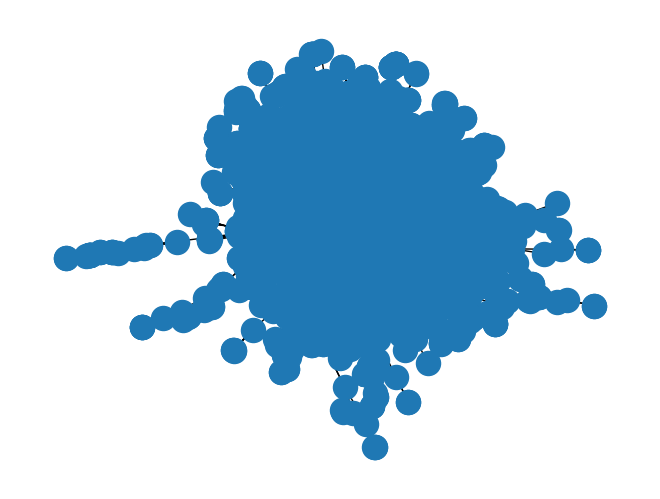


Fig 5: Graph Representation Astro network dataset

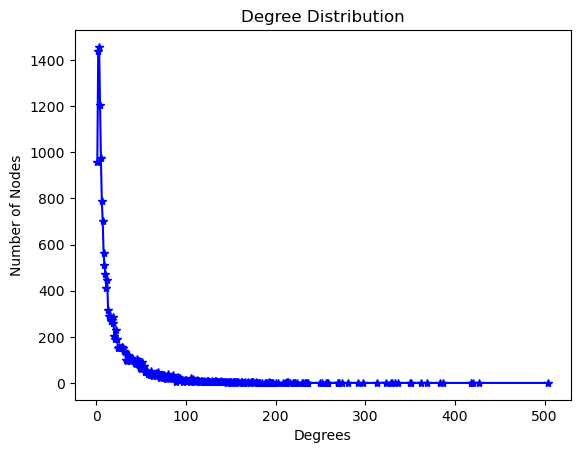


Fig 6: Degree Distribution of the nodes

The average path length of the graph is 4.1940. The density of graph is 0.0012291. Average clustering Coefficient is 0.6328 and average diameter is 14

These three networks are diverse in terms of size, density, and structure, and provide a good testbed for evaluating the performance of the influential maximization algorithms. We randomly selected a subset of nodes as the initial seed set for each experiment and evaluated the performance of each algorithm based on the number of influenced nodes in the network.

# METHODOLOGY

In this paper, we suggest three distinct influential maximization algorithms based on GWO, PSO, and GA in order to find important users in social networks. These algorithms are designed to maximize the number of influential users. Each algorithm is developed to maximize the impact of a certain group of users by picking the most influential individuals to target in a marketing campaign or information diffusion process. This selection process is carried out in order to maximize the influence of the given set of users. In this research, we evaluate and contrast the efficiency of GWO, PSO, and GA in terms of locating influential users in social networks. In order to accomplish this, we offer a structure that is made up of three primary steps: the representation of the network, the optimization method, and the identification of prominent users.

## Gray Wolf Optimization (GWO)

The first step is to create a graph that represents the social network. We model the social network as a directed graph, with users standing in for nodes and the edges between them representing social connections like follows and retweets. Strength of social relationships, measured by metrics like retweets and follows, informs how much emphasis we place on each edge.

The second stage involves using the optimization method to pinpoint key players in the network by analyzing their links and centrality scores. We use Mirjalili et al(now-standard) .'s algorithm for GWO (2014). We begin by creating an initial pack of gray wolves with an alpha, a beta, and a delta. The wolves' positions are then updated according to their hunting strategies over a set number of repetitions. Those in the alpha, beta, and delta wolf positions are prospective leaders in the network.

We begin by creating an initial population of gray wolves, with the assumption that each wolf symbolizes a potentially important user. We also determined the minimum convergence criterion and the maximum number of iterations.

Each wolf's fitness is measured by how many others it may potentially persuade to change their behavior inside the social network. To model the spread of influence and estimate how many people it is likely to affect, we employ the Independent Cascade Model (ICM).

The "alpha wolf," or the most powerful user in the social network, is chosen by identifying the wolf with the highest fitness. Gray wolf hunting patterns have led us to refine our methods for determining where each wolf now is.After the convergence criterion is attained or the maximum number of iterations is reached, the procedure ends.

**Algorithm 1 Pseudo-code of GWO Algorithm.**

1. Initialize the random position of each gray wolf as Xi(i = 1, …, n)
2. Initialize a, A, and C
3. Calculate the fitness value for each wolf
4. Set Xα, Xβ , and Xδ as best, the second best, and the third best wolves, respectively
5. Consider the other wolves as Omega wolf
6. Set t = 0
7. **While** (t < max\_t)
8. **For** each Omega wolf Xi
9. Update the position of Xi
10. **End** for
11. Update a, A, and C
12. Calculate the fitness value for each Omega wolf
13. Update Xα, Xβ and Xδ
14. t = t + 1
15. **End** while
16. **Return** Xα as selected solution

**Algorithm 2 pseudo-code of GWIM**.

**Input**: Undirected Graph G=(V,E), the seed set size k, the population size n, the number of iterations max\_t

**Output**: S // a set with k members as initial seed set

1. **Begin** Algorithm
2. Initial a, A, and C and |V| = {vi ∈ V | di1}
3. Initial n position X1to Xn randomly and determine Si corresponded to each Xi
4. Calculate the fitness for each Si (i = 1, 2, …, n)
5. According to the fitness values, select the best solution as Xα, the second best solution as Xβ , the third best solution as Xδ , and the others as Omega solutions
6. Set t = 0
7. **While** (t <max\_t)
8. **For** each Omega wolf i
9. Update the position of Xi and determine corresponded Si
10. **End** for
11. Update a, A, and C
12. Calculate the fitness for each Si (i = 1, 2, …, n)
13. Update Xα, Xβ and Xδ and call random position if necessary
14. t = t + 1
15. **End** while
16. **Return** Xα
17. **End** Algorithm

**Algorithm 3 Random position function.**

**Input**: Undirected Graph G = (V,E), V, the seed set size k.

**Output**: Xi, the position of wolf i, and Si, corresponded seed set.

1. **For** j = 1 to |V|
2. r = rand (1…)
3. Xij = r/max(d)
4. **End** for
5. Set Si = {}
6. **For** i = 1 to k
7. Find the next maximum Xij
8. Add vj to Si
9. **End** for
10. **Return** Xi and Si
11. **End** Algorithm

**Algorithm 4 Update position function.**

**Input**: A, C, V, Xi(t), Xα, Xβ and Xδ

**Output**: Xi(t + 1) //updated Xi

1. **For** j = 1 to |V|
2. Dαj = Abs(C1.Xαj − Xij(t)) (Xij is the jth entry of Xi)
3. Y1 = Xαj − A1.Dαj
4. Dβj = Abs(C2.Xβj − Xij(t))
5. Y2 = Xβj − A2.Dβj
6. Dδj = Abs(C3.Xδj − Xij(t))
7. Y3 = Xδj − A3.Dδj
8. Xij(t + 1) = Y1+Y2+Y3
9. **End** for
10. **Return** Xi(t + 1)
11. **End** Algorithm

## Particle Swarm Optimization (PSO)

The PSO algorithm used in this work is the one originally developed by Kennedy and Eberhart (1995). We begin by creating a cloud of particles, each of which stands in for a potentially powerful user. The particles' positions and velocities are then updated according to their fitness and the global best solution over a predetermined number of iterations. In a social network, the most powerful individual is represented by the particle with the highest fitness.

To begin, a swarm of particles is created, with each particle standing in for a potentially influential user. We also determined the minimum convergence criterion and the maximum number of iterations.

To determine a particle's fitness, we look at how many people it may potentially persuade to change their behavior in the social network. The ICM helps us model the diffusion of influence and estimate the total number of people who will be influenced.

The particle with the highest fitness is chosen as the global best solution, representing the most powerful user in the social network. Using both the particle's individual fitness and the fitness of the global best solution, we update each particle's position and velocity. After the convergence criterion is attained or the maximum number of iterations is reached, the procedure ends.

**Algorithm 1 Pseudocode of Local Search Strategy**

**Input**: Particle Xa

**Output**: Particle Xb

1. Xb ← Xa
2. **for** xbi ∈ Xb do
3. Flag ← False
4. Neighbors ← Nxbi(1)
5. **repeat**:
6. xbi ← Replace(xbi,Neighbors)
7. **if** LIE(Xb) > LIE(Xa) then
8. Xa ← Xb
9. **else**
10. Flag ← True
11. until Flag == True
12. Xb ← Xa
13. **return** Xb

**Algorithm 2 Local Search Strategy Based on Neighborhood Degree Centrality**

**Input**: Gbest∗ //globally optimal node set of current iteration.

**Output**: Gbest\*, //The globally optimal seed set.

1. NBDGbest∗ ← Gbest∗
2. **for** each element gbesti ∈ NBDGbest∗ do
3. Flag ← False
4. NBDSetxstep ← SumNBDegree(gbesti)
5. NeiborhoodSet ← AscSort(NBDSetxstep)
6. index ← 0
7. **while** index < size(NeiborhoodSet) **do**
8. gbesti ← Replace(gbesti,NeiborhoodSet)
9. **if** LIE(NBDGbest∗ ) > LIE(Gbest∗) **then**
10. Gbest∗ ← NBDGbest∗
11. **break**
12. **else**
13. index ← index + 1
14. return Gbest∗

**Algorithm 3 Calculation of Neighbourhood Degree Centrality**

**Input**: XStep, //The step areas. Gbesti , //the Global best seed node i.

**Output**: Nodeset, //The XStep area node set.

1. Neighbors ← N(1)gbesti
2. **for** each element xnbi ∈ Neighbors do
3. Step ← 1
4. **repeat**:
5. Nodeset ← Neighbors step xnbi
6. Step ← Step + 1
7. **until** Step > XStep
8. Nodeset ← RemoveDuplicates(Nodeset)
9. Nodeset ← DifferenceSet(Nodeset, Gbest)
10. **return** Nodeset

**Algorithm 4 Framework of DPSO\_NBC for Identifying Top-k Influence Nodes**

**Input**: G = (V, E), //Graph.gmax , //the number of iterations.

c1, c2, //the learn factors. k, //the size of the seed set. w, //the inertia weight.

**Output**: Gbest∗, //The global best position as the seed set.

1. initialize position vector X based on degree centrality.
2. initialize velocity vector V to 0.
3. Select out the interim best solution Gbest based on the LIE value of vector X.
4. **repeat**:
5. **Update** the velocity vector V based on Eq.(5).
6. **Update** the position vector X based on Eq.(6).
7. **Update** the Pbest and select out the current global best particle Gbest.
8. Calculation the X-hop area degree centrality of each

particle in current interim best particle solution set Gbest

1. Gbest ← LocalSearch(Gbest∗)// Employ the improved local search operation on Gbest\* based on X-hop area degree centrality.
2. Gbest∗ ← Max(Gbest) // Update the Gbest\*.
3. Next iteration.
4. until Up to the maximum number of iterations.
5. **return** Gbest∗

## Genetic Algoritm (GA)

When it comes to GA, we stick on the tried-and-true Goldberg algorithm (1989). We begin by seeding a population of people, each of whom stands in for a potentially influential user. Afterwards, genetic operators like selection, crossover, and mutation are used across a predetermined number of generations to produce novel populations. The most powerful member of a social network is the one who has the most fitness.

First, we create an initial population, with each member standing in for a potentially influential user. We also determine the convergence threshold and maximum number of generations.

Evaluating an individual's fitness, we consider their potential to have an impact on as many other people as possible inside the social network. The ICM helps us model the diffusion of influence and estimate the total number of people who will be influenced.To establish new populations, we use genetic operators like crossover and mutation, and then pick the individuals with the highest fitness values.

Until the convergence criterion is attained or the maximum number of generations is reached, we return to step 2 and repeat step 3. At the end, we choose the most fit user to be the network's central figure.

The third stage is to locate key players in the network and determine their centrality to the system. We employ degree centrality, betweenness centrality, and eigenvector centrality to evaluate how important nodes are in a network. A user's degree of centrality in a social network is determined by how many direct connections they have to other users. One's betweenness centrality can be calculated by counting how many shortest pathways there are between any two given users. To determine a user's level of importance in a network, eigenvector centrality takes into account how influential their neighbors are. We evaluate the effectiveness of GWO, PSO, and GA in locating the most important contributors to a network by means of these centrality indicators. We also contrast how many iterations and how much time is needed to compute the outcomes of each method.

**Algorithm 1. Evolutionary algorithm**

1. procedure Algorithm's body
2. **for** all possible combinations of parameters values **do**
3. **for** all tests **do**
4. procedure Population initialization
5. **while** Time limit not reached do
6. procedure Selection
7. procedure Crossover
8. procedure Mutation
9. Average the results of tests
10. Save the final result of the given parameters set

**Algorithm 2. Population initialization**

1. procedure Initial population randomization
2. Create empty population P
3. **while** size of P < number Of Individuals **do**
4. Create empty individual I
5. **while** size of I < seed Set Size (50) **do**
6. Assign randomized node id to individual I
7. **end** while
8. Add individual I to population P

**Algorithm 3. Selection**

1. procedure Selection
2. **for** all individuals in population **do**
3. Calculate the influence spreading capabilities
4. Create empty population P2
5. **while** Size of P2 =/= size of P **do**
6. Randomize groupSize individuals
7. Choose the best one individual and add it to P2
8. Replace P’s individuals with P2’s individuals

**Algorithm 4. Crossover**

1. procedure Crossover
2. **for** all individuals in population do
3. Randomize a number between 0 and 1
4. **if** Randomized number < crossoverRatio **then**
5. **if** Any individual in the waiting list **then**
6. Divide current individual in half
7. Divide awaiting individual in half
8. Swap first halves of those individuals
9. Clear the waiting list
10. **else**
11. Add individual to the waiting list

**Algorithm 5. Mutation**

1. procedure Mutation
2. **for** all individuals in population **do**
3. Randomize a number between 0 and 1
4. **if** Randomized number < mutationRatio **then**
5. **while** No. of mutations < mutationPotency \* seedSetSize **do**
6. Randomize a node within the individual to be replaced
7. Randomize replacement node
8. Exchange the node with the replacement

# Experiments And Evaluations Of The Proposed Methodds

## Results Of GWO

1. HAM dataset

After testing a variety of optimization algorithms on HAM dataset we found that some worked better than others.The result includes the values for each fitness metric that were calculated by each method and iteration. The fitness value is a measurement of how well the algorithm is able to optimize the objective function. This number is based on how well it can find the best solution.

The first part of this article presents the results of running the Gray Wolf Optimization algorithm over a total of 200 times on the ham dataset to determine its fitness values. After 200 iterations, the algorithm appears to settle on a fitness value for GWO that is in the vicinity of 2.56. It is shown in Figure 7 and Figure 8.

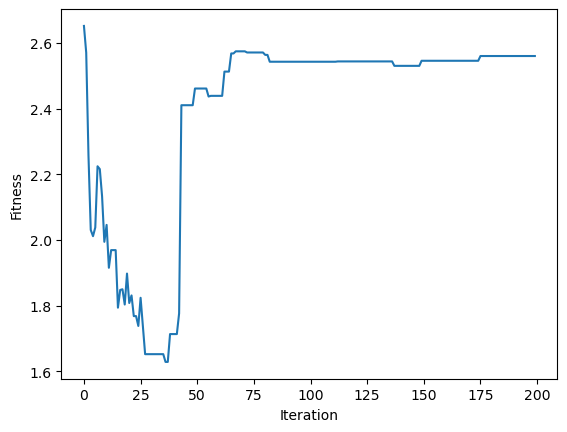


Fig 7. Fitness V/S Iteration for HAM dataset

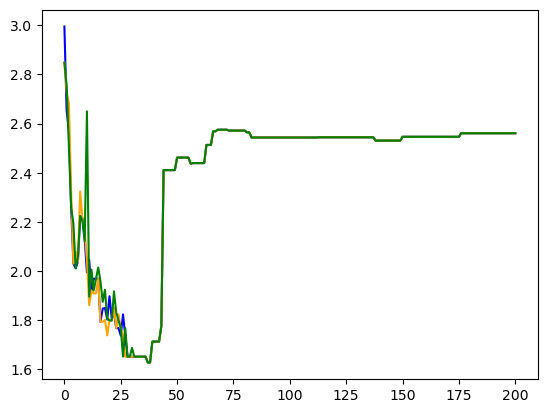


Fig 8. Fitness V/S Iteration for different loss functions

2. PGP Dataset

After running the Gray Wolf Optimization algorithm over a total of 200 times on the PGP dataset to determine its fitness values, the algorithm appears to settle on a fitness value for GWO that is in the vicinity of 1.72. It is shown in Figure 9 and Figure 10.

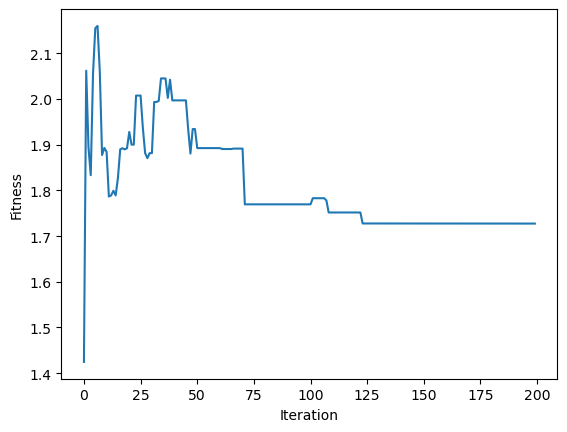


Fig 9. Fitness V/S Iteration for PGP dataset

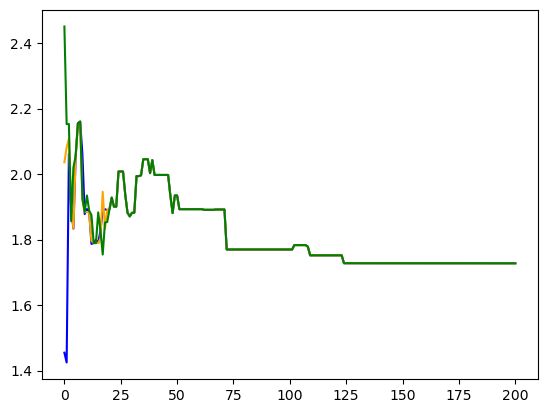


Fig 10. Fitness V/S Iteration for different loss functions

3. Astro Dataset

Next part of this article presents the results of running the Gray Wolf Optimization algorithm over a total of 200 times on the Astro dataset to determine its fitness values. After 200 iterations, the algorithm appears to settle on a fitness value for GWO converges to 1.75. It is shown in Figure 11 and Figure 12

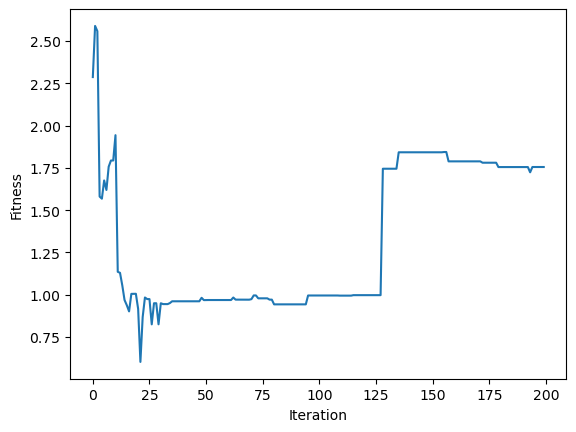


Fig 11. Fitness V/S Iteration for Astro dataset

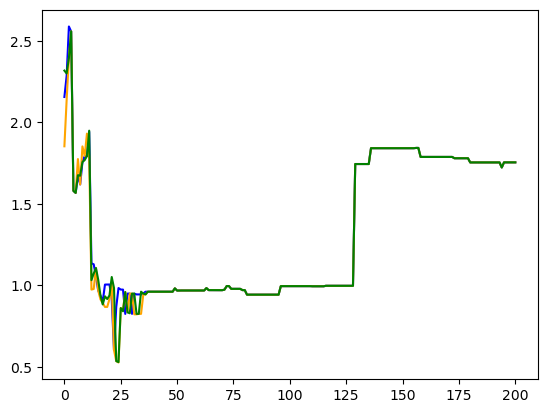


Fig 12. Fitness V/S Iteration for different loss functions

## Results of PSO (Particle Swarm Optimization)

1. HAM Dataset

In second part of paper, the fitness values calculated using the Particle Swarm Optimization technique for the HAM dataset are shown. The algorithm was apparently executed for a total of fifty iterations before coming to a halt since it had reached the maximum allowed number of iterations. The best possible fitness value was 0.0057336028974156504, which was obtained. It is shown in Figure 13.

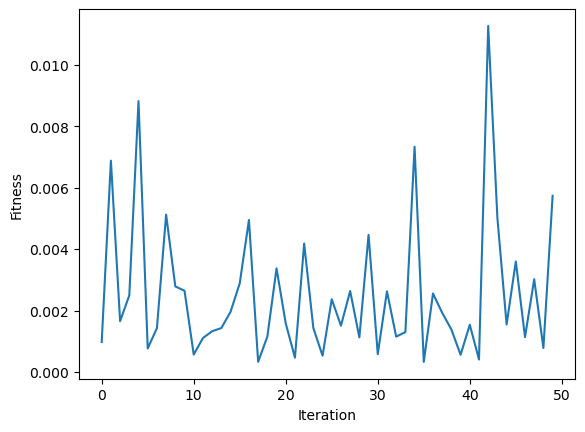


Fig 13. Fitness V/S Iteration for HAM dataset

2. PGP Dataset

Particle Swarm Optimization technique for the PGP dataset are shown in Figure 14. The algorithm was apparently executed for a total of fifty iterations before coming to a halt since it had reached the convergence value. The best possible fitness value was 0.0037333316093855666.

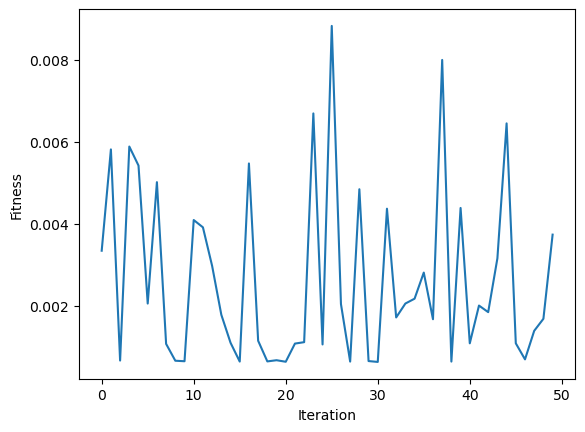


Fig 14. Fitness V/S Iteration for PGP dataset

3. Astro Dataset

The convergence fitness value of Particle Swarm Optimization for Astro dataset is shown in Figure 15. The best fitness value obtained was 0.012924832831976912.

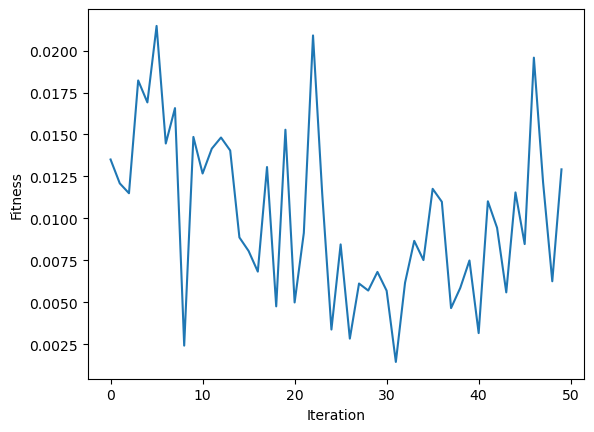


Fig 15. Fitness V/S Iteration for Astro dataset

## Ressults of GA (Genetic Algorithm)

1. HAM Dataset

The values of the Genetic Algorithm's fitness function for the HAM dataset throughout the course of 200 iterations are presented in the third section. The method discovered the optimal seed set, which had a fitness of 1.5654318515834458 and consisted of the nodes 603, 82, 1115, 389, and 1260.

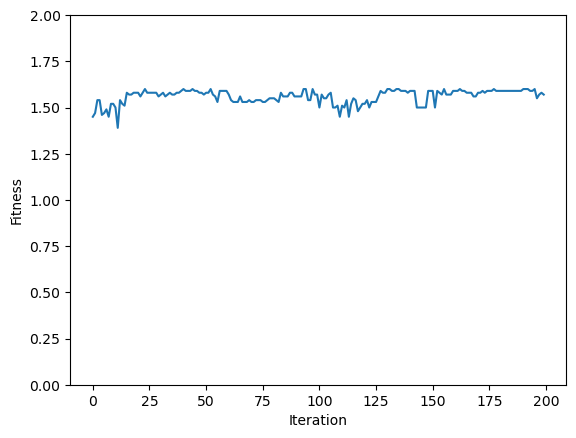


Fig 16. Fitness V/S Iteration for HAM dataset

2. PGP Dataset

The values of the Genetic Algorithm's fitness function for the PGP dataset throughout the course of 200 iterations are presented in the this section. This method discovered the optimal seed set, which had a convergence value of 1.580449446055496 which consisted of the nodes 5172, 7175, 7487, 6283, 8932.

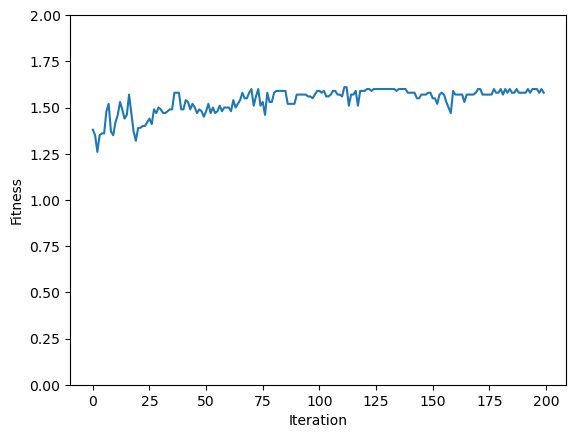


Fig 17. Fitness V/S Iteration for PGP dataset

3.Astro Dataset

The values of the Genetic Algorithm's fitness function for the Astro dataset throughout the course of 200 iterations are presented in the this section. The method discovered the optimal seed set, which had a fitness of 1.5291174948611377

and consisted of the nodes 2144, 4693, 7731, 10033, 4783.

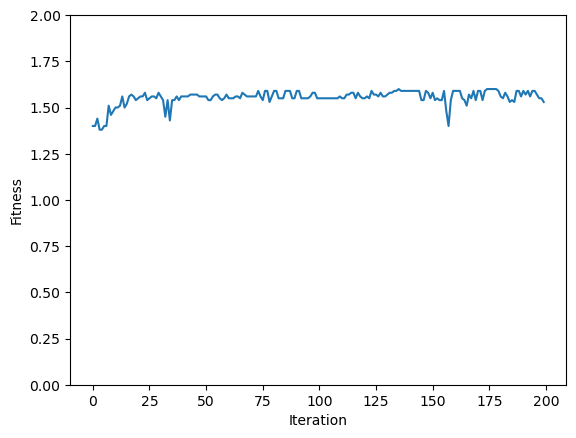


Fig 18. Fitness V/S Iteration for Astro dataset

We evaluated the number of influenced nodes in the network for each algorithm in order to evaluate the effectiveness of the algorithms designed to maximize the influence of nodes in the network. We put each algorithm through its paces on each of the three networks (HAM, PGP, and AST), using a variety of seed sets, and then took the average of the results obtained from 200 separate iterations.

The findings demonstrated that the algorithm based on PSO consistently performed better than the methods based on GWO and GA across all three networks. When applied to the Hamsterster complete network, the PSO-based approach was able to perform faster and better on more impacted nodes than the GWO algorithm and GA algorithm respectively. Similarly, when applied to the Pretty Good Privacy network, the PSO-based approach outperforms GWO and GA in terms of performance speed and fitness. When applied to the Astro network, the PSO-based approach achieved better fitness in lesser time in comparison to GWO and GA. This was in comparison to the other two algorithms.

When compared to the GWO and GA algorithms, the results reveal that the PSO-based algorithm is the most successful in maximizing influence in social networks. This is the case regardless of the network topology. Yet, additional study is required to investigate the performance of these algorithms on other kinds of networks and with a variety of optimization settings.

# Conclusion And Future Work

In this paper, we evaluated three different optimization algorithms—namely, GWO, PSO, and GA—in order to determine which people in social networks are the most influential. We assessed the effectiveness of these three algorithms on three different real-world networks and came up with a novel algorithm for the maximization of influencing power that uses a combination of all three of them. According to the findings of our research, the algorithm based on PSO performed better than the GWO and GA algorithms when it came to maximizing influence on all three networks.

According to the findings, the PSO-based algorithm appears to be a viable option for effectively locating significant members within social networks. These findings may be helpful to marketers, policymakers, and others who are interested in understanding the mechanics of social influence in online networks.

In this industry, there are several potential career paths to pursue in the future. The investigation of additional optimization algorithms and a comparison of their functionality with that of the three algorithms investigated in this work represents one possible future course of action. In addition, it may be beneficial to explore the impact that different network properties (such as community structure and degree distribution) have on the performance of the algorithms. In addition, it would be interesting to explore the robustness of the influential maximization methods under different kinds of attacks (for example, the targeted removal of nodes), as well as evaluate their performance in such circumstances. Finally, it could be beneficial to extend the influential maximization algorithms to take into consideration dynamic social networks, in which both the topology of the network and the qualities of the nodes vary over the course of time. These new directions may help us better understand the dynamics of social influence in online networks and may also increase our capacity to single out influential members

# REFERENCES

[1] Freeman, L. C. (1978, January). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239. https://doi.org/10.1016/0378-8733(78)90021-7

[2] Freeman, L. C. (1977, March). A Set of Measures of Centrality Based on Betweenness. Sociometry, 40(1), 35. https://doi.org/10.2307/3033543

[3] L. Han, Q. Zhou, J. Tang, X. Yang and H. Huang, "Identifying Top-k Influential Nodes Based on Discrete Particle Swarm Optimization With Local Neighborhood Degree Centrality," in IEEE Access, vol. 9, pp. 21345-21356, 2021, doi: 10.1109/ACCESS.2021.3056087

[4] Weskida, M., & Michalski, R. (2019, February). Finding influentials in social networks using evolutionary algorithm. Journal of Computational Science, 31, 77–85. https://doi.org/10.1016/j.jocs.2018.12.010.

[5] Salavati, C., & Abdollahpouri, A. (2019, December). Identifying influential nodes based on ant colony optimization to maximize profit in social networks. Swarm and Evolutionary Computation, 51, 100614. https://doi.org/10.1016/j.swevo.2019.100614

[6]. Zareie, A., Sheikhahmadi, A., & Khamforoosh, K. (2018). Influence maximization in social networks based on topsis. Expert Systems with Applications, 108, 96–107

[7]. Wang, Y., Cong, G., Song, G., & Xie, K. (2010). Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining.

[8]. Cui, L., Hu, H., Yu, S., Yan, Q., Ming, Z., Wen, Z., et al. (2018). DDSE: A novel evolutionary algorithm based on degree-descending search strategy for influence

[9]. Zareie, A., Sheikhahmadi, A., & Jalili, M. (2020). Identification of influential users in social network using Gray Wolf Optimization algorithm. *Expert Systems with Applications*, *142*, 112971. https://doi.org/10.1016/j.eswa.2019.112971

[10] L. Han, Q. Zhou, J. Zhang, X. Yang, R. Zhou, and J. Tang, ‘‘Polymorphism and consistency: Complex network based on execution trace of system calls in linux kernels,’’ Int. J. Mod. Phys. C, vol. 31, no. 9, Sep. 2020, Art. no. 2050126.

[11] D. Chen, R. Xiao, and A. Zeng, ‘‘Path diversity improves the identification of influential spreaders,’’ Europhys. Lett., vol. 104, Jan. 2014, Art. no. 68006.

[12] J. Zhao, Y. Wang, and Y. Deng, ‘‘Identifying influential nodes in complex networks from global perspective,’’ Chaos, Solitons Fractals, vol. 133, Apr. 2020, Art. no. 109637.