ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Scalable Influence Maximization Using Ant Colony Optimization with Attribute-Based Scouting

Mithun Roy[1] and Indrajit Pan[2]

ABSTRACT

Influence Maximization (IM) is a vital problem in social network analysis that focuses on identifying a small subset of influential nodes to maximize the spread of information across a network. Traditional influence maximization algorithms, including greedy and heuristic-based methods, often struggle with scalability and efficiency, especially when applied to large-scale networks. To overcome these limitations, we propose a novel Hybrid Ant Colony Optimization (HybridACO) algorithm that integrates a neighbor scouting strategy based on attribute similarity. This approach utilizes the inherent network structure by combining the global search capability of Ant Colony Optimization (ACO) with a local scouting mechanism that selects nodes based on their neighbors' influence potential and attribute similarity. By integrating attribute-driven scouting, HybridACO ensures that the selected nodes are not only topologically influential but also contextually relevant for the diffusion process. Comprehensive evaluations on both synthetic and real-world benchmark networks show that the proposed algorithm significantly surpasses existing state-of-the-art (SOTA) methods in influence spread, computational efficiency, and robustness.

Article information:

Keywords: Influence Maximization, Social Network Analysis, Ant Colony Optimization (ACO), HybridACO Algorithm, Neighbor's Scouting Strategy (NSS), Attribute Similarity

Article history:
Received: April 21, 2025
Revised: September 18, 2025
Accepted: October 9, 2025
Published: October 31, 2025 (Online)

DOI: 10.37936/ecti-cit.2025194.261748

1. INTRODUCTION

Social networks have emerged as a cornerstone of modern society, encompassing both online and offline platforms that facilitate interaction and information sharing. Online social networks, such as Twitter, Facebook, and Instagram, have amassed billions of users and connections, serving as dynamic communication channels for individuals and organizations. These platforms offer unparalleled opportunities for businesses and advertisers to reach a vast global audience and promote their products, services, or ideas effectively. On the other hand, offline social networks encompass a broader spectrum of relationships, including social, professional, and physical connections that are pivotal for local and regional interactions. These networks include personal ties with family, friends, and acquaintances, as well as professional collaborations and affiliations within specific domains.

Heuristic [4] [7] [8] and meta-heuristic-based [9-11] Influence Maximization (IM) algorithms search for new areas in the solution space and refine reasonable solutions. The update mechanism reinforces the successful solutions and improves the results over iterations. Ant Colony Optimization (ACO) [12] is a meta-heuristic algorithm that is inherently parallel [13]. Researchers can utilize this parallelism to speed up computations, especially in large networks. Hybrid meta-heuristic algorithms [14-17] combine various optimization techniques to solve complex problems across different applications and enhance performance and efficiency. Recent studies [18-20] have discovered the use of metaheuristic algorithms for Influence Maximization (IM) due to their flexibility and effectiveness in navigating complex search spaces. However, despite their advantages, achieving a delicate balance between maximizing the influence spread and minimizing the execution time remains a critical

[1]The author is with the Department of Computer Science and Engineering, Siliguri Institute of Technology, Sukna, Siliguri, Darjeeling, India, Email: mithunroy.cse@sittechno.edu.in
[2]The author is with the Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), RCC Institute of Information Technology, Kolkata, India, Email: indrajit.pan@rcciit.org.in
[2]Corresponding author: indrajit.pan@rcciit.org.in

challenge in developing robust IM algorithms suitable for Real-world Social Networks (RSNs).

Despite its computational intensity, we propose our HybridACO algorithm, which combines the meta-heuristic principles of Ant Colony Optimization (ACO) with a neighbor scouting mechanism based on attribute similarity. Hybrid Ant Colony Optimization (HybridACO) discover the collective intelligence of artificial ants to efficiently explore the search space and employs attribute-driven node selection to improve the quality of the identified seed nodes. The Neighbor Scouting Strategy al-lows the algorithm to prioritize nodes with strong connections and similar attributes, effectively capturing the network's inherent structural and semantic characteristics. This novel approach not only enhances the exploration of solution spaces but also ensures computational scalability and robustness, making it an efficient tool for influence maximization in RSNs. The primary contributions of this study are as follows.

- We propose a hybrid optimization approach that integrates Ant Colony Optimization (ACO) with a neighbor scouting strategy to maximize influence spread.
- This algorithm integrates attribute-based similarity to enhance its capability of identifying influential nodes within heterogeneous networks.
- Our approach demonstrates superior scalability and robustness across diverse network structures, out-performing baseline methods in terms of influence spread and computational efficiency.
- We provide valuable insights into how integrating structural and attribute-based strategies can significantly improve the influence maximization process.
- We conducted extensive experiments on synthetic and real-world networks using the Independent Cascade (IC) and Linear Threshold (LT) models, demonstrating that HybridACO outperforms state-of-the-art (SOTA) methods in effectiveness.

The remainder of this study is structured as follows. Section 2 provides a review of related work on influence maximization. Section 3 presents the preliminaries. The proposed algorithm is given in Section 4. Section 5 presents numerical examples, and Section 6 discusses the results. Finally, Section 7 concludes the study and out-lines potential directions for future research.

2. LITERATURE SURVEY

2.1 Greedy-Based Heuristic Approaches

Kempe *et al.* [6] demonstrated that researchers can model Influence Maximization using information diffusion frameworks such as the Linear Threshold (LT), Independent Cascade (IC), and Weighted Independent Cascade (WIC), emphasizing the computational challenge of identifying the most influential nodes in a network. To address this problem, they proposed a greedy algorithm based on a hill-climbing strategy applied to a submodular function, achieving an approximation factor of $\mathbf{1} - \frac{\mathbf{1}}{e^{-\epsilon}}$, where e is the natural logarithm base that accounts for the Monte Carlo simulation error. However, the algorithm remains computationally intensive owing to its iterative process of selecting k seed nodes and accurately assessing the influence spread, which requires tens of thousands of Monte Carlo simulations.

Greedy algorithms struggle with scalability in large social networks because of their high computational cost, making them primarily suitable for smaller networks. To address this issue, Leskovec *et al.* [3] proposed the Cost-Effective Lazy-Forward (CELF) algorithm, which exploits the property of submodularity and uses a priority queue to efficiently manage marginal gains during seed node selection. CELF is approximately 700 times faster than the traditional greedy algorithm because it reduces the Monte Carlo simulation process. Goyal *et al.* [2] further optimized this method by using CELF++. Chen *et al.* [4] introduced the NewGreedy algorithm, which reduces the time complexity by removing non-contributory edges to create a smaller network for more efficient information diffusion. The MixGreedy algorithm further enhances the efficiency by combining concepts from New-Greedy and CELF. Greedy algorithms effectively identify influential seed nodes in small networks; however, their scalability becomes computationally expensive when applied to large networks.

2.2 Attribute-Based Heuristic Approaches

The incorporation of node attributes into influence maximization has attracted considerable attention. Huang *et al.* [21] explored tractable models for information diffusion in social networks, focusing on both the structural and attribute-driven dynamics. Their work addressed the computational challenges of simulating information spread by proposing simplified models that retain predictive accuracy. They introduced heuristic methods that incorporate both network topology and attribute relevance to determine how information propagates through social networks. Liang *et al.* [18] introduced heuristic and metaheuristic approaches for influence maximization that incorporated attribute-based similarity to target heterogeneous networks. Their method integrates the network's structural properties with node attributes, such as preferences and demographic characteristics, to improve the identification of influential nodes. The algorithm computes a similarity score between nodes based on their attributes and integrates this score with traditional influence measures.

2.3 Metaheuristic-Based Approaches

Metaheuristic methods, namely Genetic Algorithms (GAs), Ant Colony Optimization (ACO), and

Particle Swarm Optimization (PSO), have shown promise in solving the influence maximization problem. Jiang *et al.* [17] utilized simulated annealing to optimize the fitness function, achieving significantly higher accuracy and a performance speedup of 100 to 1000 times compared to the greedy algorithm. Bucur *et al.* [22] introduced a hybrid Genetic Algorithm (GA) approach for influence maximization that integrates local search techniques to refine solutions during each iteration. This combination enables the algorithm to efficiently explore the search space and identify influential nodes with greater accuracy. The integration of a GA with local search ensures better convergence and reduces the risk of suboptimal solutions. Roy *et al.* [20] used GAs to iteratively refine a population of candidate solutions by simulating selection, crossover, and mutation processes. Khatri *et al.* [19] introduced a discretized Harris Hawks Optimization (HHO) algorithm combined with a Neighbor's Scout Strategy to enhance the selection of influential nodes in social networks. By leveraging community structures, this method optimizes the node selection process, whereas the Neighbor's Scout Strategy improves exploration, prevents premature convergence, and refines the search for optimal solutions. Singh *et al.* [10] proposed a hybrid Ant Colony Optimization-based Influence Maximization (ACO-IM) algorithm that integrates the Ant Colony Optimization (ACO) framework with a dynamic learning mechanism to optimize the influence spread. The approach simulates pheromone deposition and evaporation to guide the search process, while dynamically learning from the network's structural properties to enhance node selection. The PSO-IM algorithm, developed using particle swarm optimization, applies degree-based heuristic initialization and local search strategies to en-hance performance and accuracy. Gong *et al.* [11] developed a Discrete Particle Swarm Optimization (DPSO) algorithm tailored for influence maximization. The integration of DPSO with local search strategies enables the algorithm to explore the solution space more effectively and prevents premature convergence. By incorporating a discrete representation, this approach effectively modelled the influence maximization problem, thereby improving scalability and performance. Tang *et al.* [23] introduced an Enhanced Discrete Particle Swarm Optimization (EDPSO) algorithm for influence maximization. The hybrid method incorporates community detection to partition the network into smaller sub-networks, which are subsequently optimized independently. Enhancing the traditional PSO algorithm with community-driven exploration. Roy *et al.* [24] proposed a Lazy Forward Differential Evolution (LFDE) algorithm for influence maximization in large-scale social networks. This approach combines differential evolution with a lazy forward selection strategy to optimize the selection of influential nodes while reducing computational overhead.

2.4 Hybrid Metaheuristic Approaches

Hybrid methods combine multiple heuristics or metaheuristics to exploit their advantages. Singh *et al.* [9] proposed LAPSO-IM, a hybrid influence maximization algorithm that integrates Learning Automata (LA) with Particle Swarm Optimization (PSO). The approach considers both the network structure and attribute relevance to identify influential nodes effectively. The algorithm dynamically adapts its strategy by learning from the in-fluence spread achieved in previous iterations, thereby fine-tuning the selection of nodes over time. Incorporating attribute relevance improves the algorithm's ability to identify nodes with the highest potential for influence spread in heterogeneous networks. The hybrid nature of LAPSO-IM balances the exploratory capabilities of PSO with the adaptability of LA, resulting in an improved convergence speed and solution quality. Zhou *et al.* [25] proposed a hybrid approach that integrates Ant Colony Optimization (ACO) with Simulated Annealing (SA) to effectively address the influence maximization problem. The algorithm utilizes ACO's capability to explore the solution space and SA's strength in refining solutions to prevent premature convergence. This combined effect allows the method to achieve a high influence spread with reduced computational cost. The approach was evaluated on multiple real-world datasets, and it consistently outperformed standalone ACO and SA implementations.

3. PRELIMINARIES

3.1 Notations:

Table 1 presents the notations used for the problem formulations in this study.

Table 1: *Notations Table.*

Notation	Description
$G(V, E)$	A social network with vertex set V and edge set E.
(u, v)	An edge of a network 'G'.
$N(u)$	The neighbor's set of a node 'u'.
k	The number of nodes in a seed set 'S'.
S	Seed set.
$\sigma(S)$	The expected influence spread of 'S'.
α	α is a parameter used to adjust the relative importance of pheromone levels.
β	β is a parameter used to adjust the influence probabilities.
ρ	Evaporation rate.
γ	Attribute similarity function.
A_u	Similarity vector of a node 'u'.
τ_{uv}	Pheromone value of an edge (u, v).
$D(u)$	Degree of a node 'u'.
$P(v)$	Transition Probability of a node 'v'.
$\Xi(u)$	Influence score of a node 'u'.

3.2 Definitions:

Definition 1 *(Social Network)***:** A social network is a structure consisting of individuals (nodes) connected through relationships (edges). Mathematically, a social network can be represented as a graph $G = (V, E)$, where V denotes nodes and E represents edges, which can be directed, undirected, or weighted to reflect the strength or frequency of ties.

Definition 2 *(Neighbors)***:** In a social network, the neighbors of a node u are represented as $N(u)$, including all nodes v for which an edge (u, v) exists in the network. In undirected networks, neighbors share a mutual connection with u, while in directed networks, neighbors are classified as in-neighbors and out-neighbors.

Definition 3 *(Seed Nodes)***:** In a social network, seed nodes are a selected subset of nodes ($S = k$ and $S \subset V$) that serve as the starting points for influence propagation. These nodes were strategically selected to maximize the spread of influence across the networks.

Definition 4 *(Attribute-Based Similarity Function)***:** In social network analysis and influence maximization, attribute-based similarity functions measure the similarity between nodes based on their attributes or features. This function is used to identifies the influential nodes. Let u and v be two nodes in a social network and let A_u and A_v be their attribute vectors. An attribute-based similarity function calculates the similarity between two nodes based on their attribute values using cosine similarity. The function is defined as:

$$\gamma(u, v) = \frac{A_u \cdot A_v}{\|A_u\| \|A_v\|} \tag{1}$$

Where '·' denotes the dot product, and $\|A_u\|$ and $\|A_v\|$ are vector magnitudes.

Definition 5 *(Influence Spread)***:** The influence spread $(\sigma(S))$ is the expected number of nodes activated in a network through a propagation process initiated by a given seed set S.

3.3 Node Attributes:

Node attributes (Fig. 1) play a crucial role in influence maximization, as they shape the dynamics of information flow and adoption within the network. Trust fosters reliable connections and enhances the propagation of influence. The opinion sharing rate (OSR) and opinion accepting rate (OAR) control the spread and receptiveness of information, respectively, with high values indicating key disseminators and receptive nodes. Age provides context, influencing engagement levels, while category identifies homogeneous groups, aiding community-based strategies. Together, these attributes allow algorithms to efficiently identify and target influential nodes by optimizing influence spread across large-scale social networks.

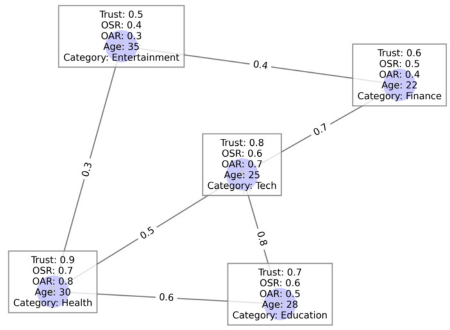

Fig. 1: *Graph with Node Attributes and Edge Weights.*

3.4 Influence Propagation Models:

In a social network, an active node refers to a node that has been influenced or activated during the influence propagation process. Node v becomes active if it satisfies certain conditions defined by the influence propagation model. Influence propagation models [6] provide a mathematical framework for understanding how influence or information spreads in a social network. Two widely used models are Independent Cascade (IC) and Linear Threshold (LT) models.

In the IC model, an active node u attempts to activate its neighbor v with a certain probability P_{uv}, and the activation process continues iteratively until no further nodes become activated. The probability that a node v becomes activated is defined as:

$$P(activation\ of\ v) = 1 - \prod\nolimits_{u\ in\ N(v)} (1 - P_{uv}) \tag{2}$$

Where $N(v)$ represents the set of active neighbors of v. In the LT model, each node v has a threshold $\Theta_v \in [0, 1]$, that represents the minimum cumulative influence required for its activation. Node v becomes active if the sum of the influence weights from its neighbors is satisfied.

$$\sum_{u \in N(v)} w(u, v) \geq \Theta_v \tag{3}$$

Where $N(v)$ is the set of neighbors of v, and $w(u, v)$ is the weight of the influence from neighbors u to v.

Both models capture distinct aspects of influence dynamics: the IC model focuses on independent influence attempts, while the LT model emphasizes cumulative thresholds. Together, they serve as essential frameworks for studying influence propagation in networks.

3.5 Ant Colony Optimization (ACO):

Ant Colony Optimization (ACO) is a bio-inspired optimization algorithm based on the foraging behavior of ants [26]. It is particularly effective for solving combinatorial optimization problems, such as network routing. In ACO, a population of artificial ants explores potential solutions by constructing paths and depositing pheromones, which serve as communication mechanisms. The pheromone levels are updated based on solution quality, guiding subsequent ants toward better paths while maintaining exploration. The algorithm iteratively balances exploration and exploitation, making it robust in identifying near-optimal solutions in large-scale problem spaces. The three key steps in ACO are: (1) solution construction by ants guided by pheromone trails and heuristic values, (2) pheromone updating through evaporation and reinforcement based on solution quality, and (3) iterative optimization to progressively refine solutions across multiple iterations.

3.6 Problem Definition:

Influence maximization is a fundamental problem in social network analysis that aims to identify a subset of nodes, known as seed nodes, capable of maximizing the spread of influence throughout the network under a specified diffusion model.

The problem can be formally defined as follows:

- **Input**: A social network represented as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each edge e_{ij} has a weight w_{ij}, representing the probability of influencing the propagation between nodes i and j. Additionally, budget k specifies the maximum number of seed nodes that can be selected.
- **Objective**: Select k seed nodes from V such that the expected spread of influence, as defined by an influence propagation model (e.g., Independent Cascade (IC) or Linear Threshold (LT)), is maximized.
- **Constraint**: The number of selected nodes must not exceed k, and the algorithm should operate efficiently, even for large-scale networks.

The objective function $\sigma(S^*)$ is defined as follows:

$$\sigma(S^*) = \max_{S \subset V, |S| \leq k} \mathbb{E}\left[\sum_{t=1}^{R} \sigma(S)_t\right] \tag{4}$$

Eq. (4) aims to identify the optimal seed set S^* of at most k nodes within a set of all nodes V in the network. Here, $\sigma(S)_t$ denotes the influence spread at time step t initiated by seed set S, and R represents the total number of propagation rounds considered. The expectation $\mathbb{E}[\cdot]$ captures the stochastic nature of the diffusion process and reflects uncertainty in the propagation of influence through the network. This formulation seeks to maximize the expected total influence across all rounds, ensuring that the selected seed set produces the highest average spread of influence under the given propagation model Φ.

4. PROPOSED METHODOLOGY:

Influence maximization aims to identify a set of seed nodes (S) in a social network that maximizes the spread of influence. To address the limitations of existing approaches, we propose a Hybrid Ant Colony Optimization (HybridACO) algorithm that integrates attribute-based neighbor selection with a pheromone-guided search. An overview of the HybridACO workflow is presented in **Fig. 2**.

4.1 Proposed Workflow:

Initialization:

In each iteration of the HybridACO algorithm, the process begins by initializing the pheromone values and parameters. All edges in the network are assigned a uniform initial pheromone level (e.g., $\tau_{uv} = 0.5$), and control parameters such as α, β, pheromone evaporation rate ρ, and reinforcement constant $Q(\in [0, 1])$ are set. This ensures a balanced starting point, where no edge is initially favored, and the algorithm can gradually adapt based on experience.

Attribute-Based Neighbor Scouting:

Attribute-Based Neighbor Scouting is a mechanism that evaluates nodes based not only on their structural properties (e.g., degree or connectivity) but also on their attribute similarities (γ). This approach incorporates additional node-specific information, such as trust, opinion sharing rate, opinion acceptance rate, age, and category, to enhance decision-making during the exploration phase of the algorithm. In the context of influence maximization, attribute-based scouting ensures that ants prioritize selecting nodes that are not only influential in terms of network topology but also exhibit high similarity to other nodes based on their attributes.

At this stage, each ant selects nodes probabilistically to form a candidate seed set. A random node u is chosen from the set of nodes V where the degree $D(u)$ is greater than or equal to the average degree $D_{avg}(G)$ of the graph.

$$u = \{u : D(u) \geq D_{avg}(G)\} \tag{5}$$

The probability of choosing neighbor v from node u is given by Eq. (6), where both the pheromone trail intensity and the product of attribute similarity and edge influence probability contribute to the decision.

$$P(v) = \frac{\tau_{uv}^{\alpha} \cdot (\gamma(A_u, A_v) \cdot P_{uv})^{\beta}}{\sum_{v \in N(u)} \tau_{uv}^{\alpha} \cdot (\gamma(A_u, A_v) \cdot P_{uv})^{\beta}} \tag{6}$$

Here, $\gamma(A_u, A_v)$ denotes the attribute similarity function, and p_{uv} represents the edge influence probability. Parameters α and β are used to control the relative importance of the pheromone levels and heuristic information. By tuning these parameters, the algorithm achieves a balance between exploration (driven

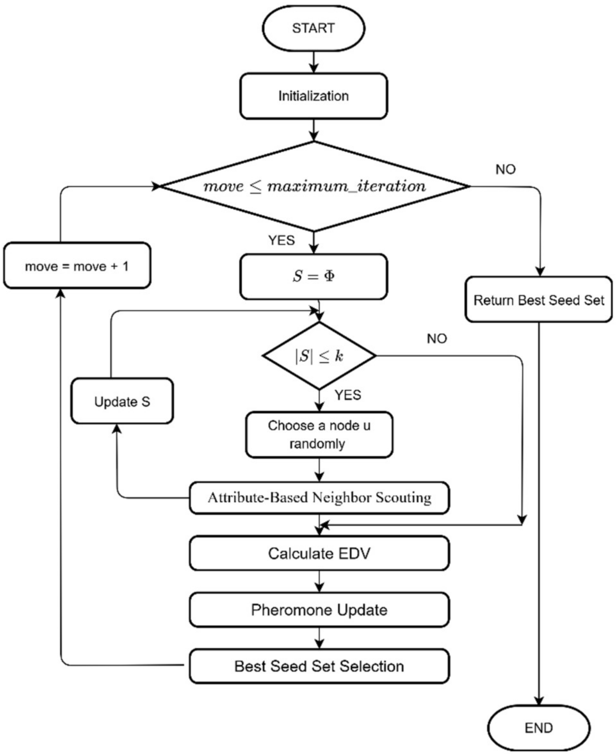

Fig.2: *Flowchart of HybridACO Algorithm.*

by heuristic information) and exploitation (guided by accumulated pheromone knowledge), enabling ants to construct diverse yet high-quality solutions.

The ant selects the neighbor with the highest transition probability that is not already part of the seed set S. This method reduces the computational complexity by limiting the search space to local neighborhoods while effectively capturing the immediate influence dynamics of seed nodes.

Evaluation of Effective Diffusion Value (EDV):

The EDV is a metric used in influence maximization to quantify the overall impact or influence spread achieved by a specific set of seed nodes within a network. It accounts for both direct and indirect influences, where direct influence represents the immediate activation of nodes directly connected to the seed set, and indirect influence captures the activation of nodes further away as the influence cascades through the network.

Once a candidate seed set S is formed, its quality is evaluated using the Effective Diffusion Value (EDV). For each seed $u \in S$, the EDV is calculated as:

$$\Xi_{EDV}(u) = 1 + \sum\nolimits_{v \in N(u)} P(u,v) \tag{7}$$

The total influence spread of the set is then given by:

$$\sigma(S) = \sum\nolimits_{u \in S} \Xi_{EDV}(u) \tag{8}$$

This value serves as a performance indicator of the constructed seed set and directly influences the pheromone reinforcement.

Pheromone Update:

After evaluating $\sigma(S)$, pheromone levels are updated in two steps.

First, Pheromone values are reduced by a factor of $(1-\rho)$,

$$\tau_{uv} = (1-\rho) \cdot \tau_{uv} \tag{9}$$

Which restricts excessive pheromone accumulation and encourages continued exploration.

The pheromone values were reinforced in proportion to the quality of the solution, calculated as $\frac{Q}{\sigma(S)}$.

Reinforcement is then applied to the edges whose source nodes belong to the chosen seed set:

$$\tau_{uv} = \tau_{uv} + \frac{Q}{\sigma(S)}. \tag{10}$$

This reinforcement encourages ants to explore paths that lead to high-influence nodes in subsequent iterations. Q represents a constant that controls the amount of pheromone deposited.

Termination:

At the end of each iteration, the algorithm evaluates whether the termination condition is satisfied. The process terminates when either the maximum number of iterations is reached or convergence is achieved.

The best-performing seed set across all iterations is returned as the final solution.

4.1.1 Proposed Algorithm: HybridACO (seed selection algorithm)

Input: Graph G // Either Directed or Undirected Graph
Output: S_{best} // Best Influential Seed Set

1. *//Initialize Parameters:*
   $\alpha = [0.5, 1.0], \beta = [0.7, 1.0], \rho = [0.6, 0.8], MaxEDV = 0$
2. *//Set pheromone values*
   $\tau_{uv} = 0.5, \forall (u,v) \in E$
3. *Set the Node Attributes randomly* A_v, $\forall v \in V$
4. *While* $move \leq I_{Max}$:
5. $S = \{\varnothing\}$
6. *For* $ant = 1$ *to* k:
7. *//Select a random node u:*
   $D(u) \leq D_{avg}(G)$
8. *//Identify one-hop neighbors:*
   $N(u) = \{v \in V : (u,v) \in E\}$
9. $max_{P(v)} = 0$
10. $temp = None$
11. *For each* $v \in N(u)$:
12. *//Compute transition probability:*
    $P(v) = \frac{\tau_{uv}^{\alpha} \cdot (\gamma(A_u, A_v) \cdot P_{uv})^{\beta}}{\sum_{v \in N(u)} \tau_{uv}^{\alpha} \cdot (\gamma(A_u, A_v) \cdot P_{uv})^{\beta}}$
13. *If* $v \in S$ *and* $P(v) > max_{P(v)}$
14. $temp = v$
15. $max_{P(v)} = P(v)$.
16. *End If*
17. *End For*
18. *//Add the best-scored neighbor to the seed set:*
    $S = S \cup temp$
19. *End For*
20. $\Xi_{EDV}(u) = 1 + \sum_{v \in N(u)} P(u,v), \forall u \in S$

21. *//Compute Influence Spread:*
22. $\sigma(S) = \sum_{u \in S} \Xi_{EDV}(u)$
23. *For each edge* $(u, v) \in E$ *and* $u \notin S$:
    //Update pheromone evaporation:
24. $\quad T_{uv} = (1 - \rho) \cdot \tau_{uv}$
25. *End For*
26. *For each* $u \in S$:
    //Reinforce pheromone:
27. $\quad T_{uv} = T_{uv} + \frac{Q}{\sigma(S)}$
28. *End For*
29. *If* $\sigma(S) > MaxEDV$:
30. $\quad MaxEDV = \sigma(S)$
31. $\quad S_{best} = S$
32. *End If*
33. $move = move + 1$.
34. *End While*
35. *return* S_{best}

The Influential Seed Set Selection Algorithm Using HybridACO employs Ant Colony Optimization (ACO) to identify influential nodes in a graph for influence maximization. In lines 1-3, the algorithm initializes key parameters that control the influence of pheromone ($0.5 \leq \alpha \leq 1.0$), heuristic factors ($0.7 \leq \beta \leq 1.0$), and evaporation rate ($0.6 \leq \rho \leq 0.8$), respectively, and a variable $MaxEDV$ is initialized to track the maximum influence spread across iterations. The initial pheromone levels (τ_{uv}) for all edges were set to 0.5, and node attributes (A_v) were set for all vertices $v \in V$. In lines 6-19, the algorithm executes a predefined number of iterations. At the beginning of each iteration, the seed set S is reset to empty. For every ant (up to k ants), a starting node u is randomly selected from nodes whose degree (D(u)) is greater than the average degree ($D_{avg}(G)$) of the graph. The one-hop neighbors of u, denoted $N(u)$, were identified. The transition probabilities for selecting neighbors are computed using τ_{uv}^{α} (pheromone influence) and $(\gamma(A_u, A_v) \cdot p_{uv})^{\beta})$ (heuristic influence). The node with the highest transition probability that is not already in S is selected and added to S. This process was repeated for all ants, thereby expanding seed set S. Once seed set S is formed, the expected diffusion value (EDV), and influence spread $\sigma(S)$ are calculated by summing the contributions from the nodes in S and their neighbors. The pheromone levels on the edges are updated: evaporation reduces the pheromone on all edges by $(1 - \rho) \cdot \tau_{uv}$ (line 24), and reinforcement increases the pheromone levels on the edges connected to the seed nodes by a reward factor, $\frac{Q}{\varsigma(S)}$, proportional to the influence spread (line 27). In lines 20-32, if the influence spread of the current seed set exceeds the maximum recorded value $MaxEDV$, both the best seed set (S_{best}) and $MaxEDV$ are updated accordingly. This process is repeated across multiple iterations, ensuring that the algorithm explores various paths and progressively refines the seed set selection. Finally, the algorithm returns the best seed set,S_{best}, providing an optimized set of influential nodes for the graph.

4.1.2 Premature Convergence Avoidance

Practical exploration of the solution space helps prevent premature convergence in HybridACO by monitoring the pheromone trail diversity during the optimization process. HybridACO integrates the adaptive parameter tuning and pheromone diversification strategies. By balancing exploration (Eq. 9) with exploitation (Eq. 10), the algorithm avoids getting trapped in local optima. Specifically, the use of dynamic α, β, and ρ based on network structure ensures sustained search diversity, leading to more robust and globally optimal influence spread solutions across different networks. This adaptive mechanism balances exploration and exploitation, thereby improving the performance of HybridACO in influence maximization tasks.

4.1.3 Stagnation Prevention

Stagnation occurs when most ants repeatedly generate identical solutions, often because of inappropriate parameter settings. In HybridACO, this is mitigated by setting the parameters to ρ, α and β to optimal values, which effectively balances pheromone evaporation, heuristic influeence, and pheromone influence. A moderate ρ value reduces the dominance of past solutions by enabling controlled pheromone evaporation, thereby encouraging the exploration of new regions in the search space. Similarly, the specified ranges for α and β parameters ensure an optimal trade-off between heuristic guidance and pheromone influence, preventing either factor from dominating the solution generation process. This parameter configuration enables HybridACO to maintain a robust balance between exploration and exploitation, foster solution diversity, reduce the likelihood of stagnation, and enhance its effectiveness in influence maximization tasks.

4.1.4 Time Complexity

The time complexity of the HybridACO algorithm for influential seed set selection is determined by its main computational steps. The outer loop, which runs for I_{Max} iterations, governs the overall execution. Within each iteration, k ants are processed, where k is the colony's size. For each ant, selecting a node involves calculating its degree $O(|V|)$ and identifying one-hop neighbors $O(\Delta)$, where Δ is the maximum degree of the graph. The influence score for a node is computed by summing over its neighbors $O(\Delta)$, and the transition probabilities for all neighbors are calculated in $O(\Delta)$. Updating pheromone levels requires evaporation and reinforcement for all edges $O(|E|)$ at each iteration, whereas the influence spread calculation for the seed set involves $O(k \cdot \Delta)$. By combining these steps, the overall time complexity of the algorithm becomes $O(I_{Max} \cdot (k \cdot |V| + k \cdot \Delta^2 + |E|))$. For dense graphs where $|E| = O(|V|^2)$, the complexity simplifies to

$O(I_{Max} \cdot k \cdot |V|^2)$. Hence, the time complexity of the algorithm depends on the graph density, the number of ants, and the number of iterations.

5. NUMERICAL ILLUSTRATION

Consider a graph G (**Fig. 3**) with four nodes (A, B, C, D) and edges $(A \to B, A \to C, B \to D, C \to D, B \to A)$ with associated influence probabilities $p_{AB} = 0.7$, $p_{AC} = 0.8$, $p_{BD} = 0.6$, $p_{CD} = 0.9$ and similarity attributes $\gamma(A, B) = 0.9$, $\gamma(A, C) = 0.8$, $\gamma(B, D) = 0.7$, $\gamma(C, D) = 0.85$. The algorithm begins with parameter initialization $\alpha = 1$, $\beta = 1$, $\rho = 0.8$ and sets the initial pheromone values $\tau_{uv} = 0.5$ for all edges.

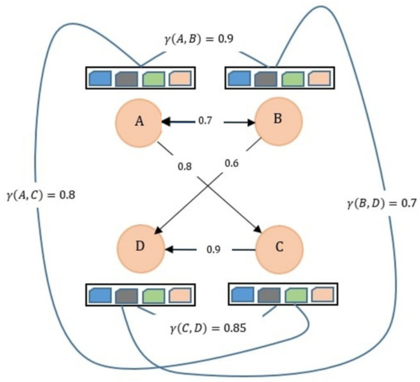

Fig.3: *Toy Network-I.*

Table 2: *Transition Probabilities.*

u	$u \to v$	$\gamma(u,v)$	P_{uv}	τ_{int}	$\tau.\gamma.P$	$P(v)$
A	$A \to B$	0.90	0.70	0.5	0.315	0.4961
	$A \to C$	0.80	0.80	0.5	0.320	0.5039
B	$B \to A$	0.90	0.70	0.5	0.315	0.6000
	$B \to D$	0.70	0.60	0.5	0.210	0.4000
C	$C \to D$	0.85	0.90	0.5	0.383	1.0000

Transition probabilities are computed as $P(B) = 0.496$ and $P(C) = 0.504$, and node C, which has the highest probability, is added to seed set S. Ant 2 selects B, identifies neighbors (A, D). Transition probabilities $P(A) = 0.6$, $P(D) = 0.4$ lead to the selection of A, which is added to S, resulting in $S = \{C, A\}$. Next, the algorithm calculates the EDV using Eq. (7) ($\Xi(A) = 1 + P(A) + P(B) = 2.0$ and $\Xi(C) = 1 + P(C) = 2.0$). Finally, the influence spread is calculated using Eq. (8) ($\sigma(S)$) as 2.0 for C and 2.0 for A, yielding $\sigma(S) = 4.0$. Because $\sigma(S) > MaxEDV$, the best seed set is updated ($S_{best} = \{C, A\}$).

After evaporation, all the pheromones were reduced to 0.1 using Eq. (9). Reinforcement is added to the edges whose source node is in the chosen seed set S. The reinforcement increment was $\frac{Q}{\sigma(S)}$ ($Q = 0.7692$ chosen randomly). Thus, the final pheromone values were $AB, AC, CD := 0.2923$ (using Eq. (10)), $BA, BD := 0.1000$. The process repeats for subsequent iterations, refining the seed set and maximizing the influence spread, eventually converging to the most influential nodes in the network.

Impact of Attribute Similarity in Influence Spread

The influence spread in a network is significantly influenced by the attribute similarity between nodes, because individuals with similar attributes are more likely to influence each other due to the tendency of similar nodes to interact more frequently. Attribute similarity, measured through shared characteristics such as interests, opinions, trust, and behaviors, directly influences the probability of influence propagation.

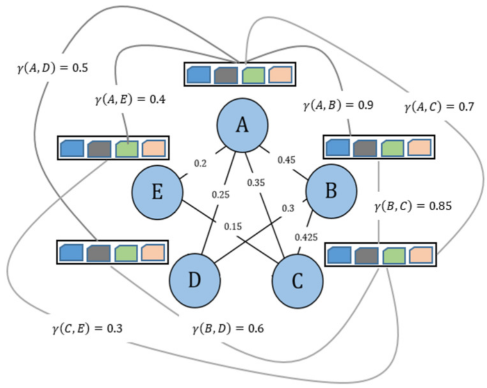

Fig.4: *Impact on Attribute Similarity in Influence Spread on a Toy Network-II.*

Consider a network (**Fig. 4**) with five nodes (A, B, C, D, E) and their pairwise attribute similarity scores measured on a scale of *0* to *1*. The table below illustrates similarity scores and their transition probabilities.

Table 3: *Transition Probabilities.*

$u \to v$	$\gamma(u,v)$	P_{uv}	$P(u \to v)$
$A \to B$	0.9	0.45	0.473684
$A \to C$	0.7	0.35	0.286550
$A \to D$	0.5	0.25	0.146199
$A \to E$	0.4	0.20	0.093567
$B \to C$	0.85	0.425	0.667436
$B \to D$	0.6	0.30	0.332564
$C \to E$	0.3	0.15	1.000000

This analysis highlights the significant role of attribute similarity $\gamma(u, v)$ and the influence probability (p_{uv}) of an edge (u, v), which determines the spread of influence across a network. Node pairs with higher similarity values exhibit greater influence probabilities owing to their stronger affinity, which aligns with the principle of homophily the tendency of similar nodes to interact more frequently. For example, the pair $A \to B$ has the highest similarity $\gamma(A, B) = 0.9$ among node A's neighbors, resulting in the highest influence probability $P(A \rightarrow B) = 0.473684$. Similarly, $B \rightarrow C$, with $\gamma(B,C) = 0.85$ and $p_{uv} = 0.425$, achieves the highest influence probability $P(B \rightarrow C) = 0.667436$ among node B's connections. Conversely, node pairs with lower similarities, such as $A \rightarrow E$, $\gamma(A,E) = 0.4$ had reduced influence probabilities $P(A \rightarrow E) = 0.093567$. For nodes with a single outgoing connection, such as $C \rightarrow E$, all influences are directed toward one neighbor, resulting in a deterministic influence probability $P(C \rightarrow E) = 1.0$. This analysis underscores the importance of considering both attribute similarity and p_{uv} in influence maximization strategies. Nodes with high similarity and p_{uv} facilitate rapid propagation within homogeneous clusters, whereas those with low similarity may require targeted interventions to effectively enhance influence spread.

6. RESULTS AND DISCUSSION

This section begins by outlining the experimental setup, including details of the dataset and probability distribution used for the diffusion models. Next, parameter tuning and convergence speed are analyzed across the three types of synthetic networks and their impact on large-scale networks is discussed. Subsequently, the performance evaluation focused on influence spread and runtime analyses. Additionally, a statistical test was conducted to highlight the significant differences between the proposed method and the compared approaches. Finally, the conclusions are summarized in **Table 7**.

6.1 Experimental Setup:

The proposed HybridACO algorithm is implemented and evaluated on four large-scale social networks. The experiments used two propagation models: the Linear Threshold (LT) model and the Independent Cascade (IC) model. Propagation probabilities were assigned uniformly across all edges, whereas activation probabilities were sampled uniformly from the range [0,1] for each node. The experiments were conducted on a high-performance Linux server equipped with the latest hardware, including a multi-core Intel Xeon Gold processor running at 3.0 GHz and 128 GB of RAM, running on Ubuntu 22.04 LTS. The implementation was developed in Python 3.8, using the NetworkX 2.0 library for graph simulation and analysis and NumPy 1.3 for numerical computations, ensuring efficient execution and consistent influence propagation simulation.

6.2 Impact Analysis on Synthetic Networks:

To evaluate the correct parameters and convergence speed of the HybridACO algorithm, experiments were conducted on synthetic networks that offered controlled environments to simulate different structural properties. Synthetic networks were generated with varying topologies, including scale-free and small-world models, to mirror real-world social network characteristics. We used three synthetic graphs (**Fig. 5**): Erdos-Renyi (ER), Barabasi-Albert (BA), and Watts-Strogatz (WS). These graphs, which have the same number of vertices, differ in structure and connectivity.

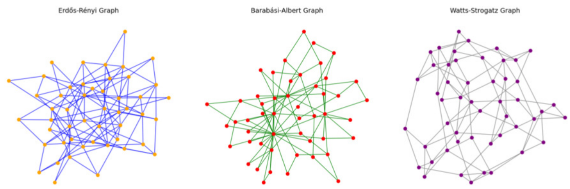

Fig.5: *Three Synthetic Network Models (ER, BA, and WS) with an Equal Number of Vertices.*

(a) Parameters Tuning:

To optimize the parameters for influence maximization using the HybridACO algorithm on synthetic networks, it is essential to align parameter selection with the structural properties of the graph. **Fig.6** illustrates an analysis of parameter tuning for the HybridACO algorithm across three synthetic graphs ER, BA, and WS, demonstrating the significant impact of graph characteristics on parameter effectiveness. For scale-free BA networks dominated by hub nodes, optimal parameters $\alpha = 0.56$, $\beta = 0.72$, and $\rho = 0.8$ effectively exploit hub influence while avoiding over-exploitation. In small-world WS networks, moderate values $\alpha = 0.5$, $\beta = 0.83$, and $\rho = 0.67$ utilize clustering and tightly connected communities. For random ER networks with uniform connectivity, setting balanced parameters $\alpha = 0.89$, $\beta = 0.83$, and $\rho = 0.73$ ensure a robust and consistent influence spread. For large-scale real-world net-works, which often contain a mix of these structural properties, optimal performance can be achieved by tuning the parameters within the empirically validated ranges: $\alpha \in [0.5, 1.0]$, $\beta \in [0.7, 1.0]$, and $\rho \in [0.6, 0.8]$. These ranges provide a balanced trade-off between exploration and exploitation, ensuring a robust and scalable influence maximization across diverse network structures.

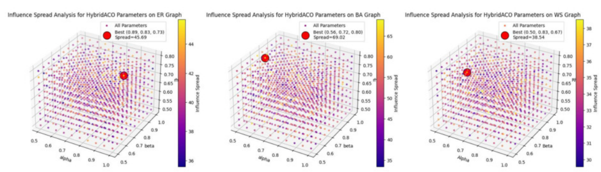

Fig.6: *Optimized Parameters for Influence Maximization across Three Types of Synthetic Graphs.*

(b) Convergence Speed:

The convergence plot (**Fig. 7**) demonstrates the influence spread achieved by the HybridACO algorithm across iterations, revealing distinct behaviors for the three synthetic graphs. The BA graph exhibits the fastest convergence, achieving approximately 62 influence spreads within the first 25 iterations and stabilizing near its maximum value of 69 by iteration 50, highlighting the significant role of hub nodes in accelerating influence propagation. The ER graph demonstrates a moderate rate of convergence, with the spread increasing steadily and plateauing at approximately 45 after 70-80 iterations, reflecting the impact of its random connectivity. In contrast, the WS graph shows the slowest convergence, reaching approximately 32 influence spreads within the first 20 iterations, their high clustering and limited long-range connections, and gradually soothing at 39 after 80-90 iterations.

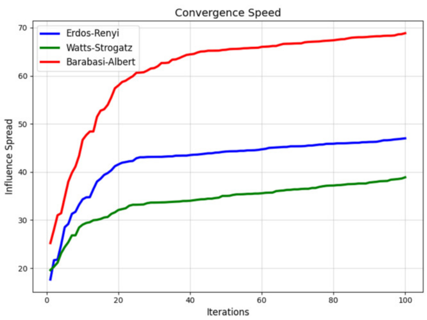

Fig. 7: *Convergence Speed of the HybridACO Algorithm on ER, BA, and WS Graphs.*

The slower convergence in WS networks can be attributed to the hampering of rapid spread across the network. These findings highlight that real-world social networks exhibit a hybrid structure combining scale-free and small-world properties.

Scale-free features account for the presence of hubs and influencers, whereas small-world characteristics ensure high clustering and short average path lengths.

The proposed HybridACO algorithm must adapt to these properties by utilizing hubs for rapid influence spread in scale-free networks and effectively utilizing clustering and a short average path length. For large-scale networks, the convergence speed of the HybridACO algorithm depends significantly on the structural properties of the network.

6.3 Performance Analysis on Real-world Social Networks:

(a) Datasets Used:

Table 4: *Network Information.*

Dataset	Nodes	Edges	Avg_Degree
NetScience	1589	2742	3.45
GRQC	5242	14496	5.53
NetHept	15233	31398	4.12
Gnutella30	36682	88328	4.81

Four benchmark real-world social networks were used: NetScience, NetHept, GRQC, and Gnutella30. NetScience represents collaborations in network theory, NetHept focuses on co-authorships in high-energy physics theory, GRQC captures collaborations in General Relativity and Quantum Cosmology, and Gnutella30 reflects peer-to-peer file-sharing networks. These datasets (**Table 4**) were obtained from the Stanford Large Network Dataset Collection (SNAP) (`https://snap.stanford.edu/data/`).

(b) Result Analysis:

The information spread is used in influence maximization because it directly measures the effectiveness of selecting seed nodes that can propagate information across the network. Maximizing information spread ensures that the chosen seeds influence as many nodes as possible, providing a clear, quantitative way to evaluate and compare the ability of different algorithms to maximize the reach or impact of information diffusion across diverse social networks.

The performance of six algorithms Degree Discount (DD) [4], CELF++ [2], Particle Swarm Optimization (PSO) [11], Discrete Particle Swarm Optimization (DPSO) [23], Ant Colony Optimization for Influence Maximization (ACOIM) [10], and the proposed HybridACO, was evaluated on four benchmark networks: **NetScience, NetHept, Gnutella30,** and **GRQC**. Based on the results shown in **Fig. 8 [(a) - (d)]** and **Fig. 9 [(a) - (d)]**, which reflect the performance under both the Linear Threshold (LT) and Independent Cascade (IC) models, it is evident that HybridACO consistently outperforms all other methods across different networks and seed set sizes.

The Linear Threshold (LT) model is a popular method for modelling the spread of influence in social networks. In this model, each node is activated if the combined influence of its neighbors crosses a certain threshold. In **Fig. 8 [(a) - (d)]**, HybridACO consistently achieves the highest influence spread across all seed set sizes and networks. CELF++ also showed strong results, closely following those of HybridACO. Other approaches such as ACO-IM, DPSO, and PSO deliver intermediate spreads, whereas DD still performs the lowest. These results suggest that HybridACO is the most effective at maximizing the influence under the LT model, especially as the seed set size increases, confirming its ability to handle complex real-world networks and outperform traditional heuristics.

The Independent Cascade (IC) model is another popular approach for demonstrating how influence spreads on social networks. It begins with an initial set of activated nodes (seed nodes), which are then given one opportunity to activate each of their neighbors according to fixed probabilities assigned to the edges. Similar patterns were observed (**Fig. 9 [(a) - (d)]**) across multiple real-world social networks, with HybridACO achieving the highest performance, par-

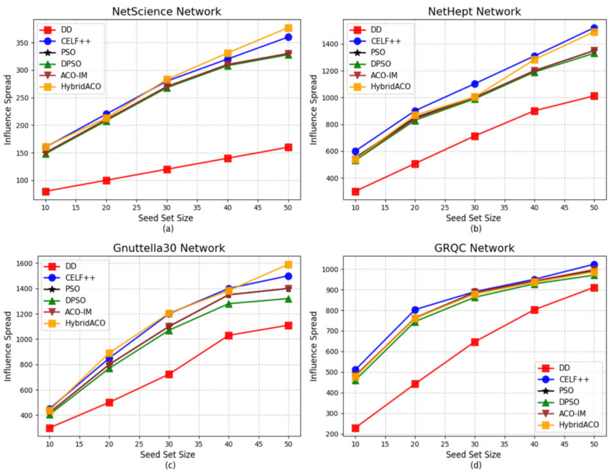

Fig.8: *Comparison of Influence Spread across Benchmark Networks Using Six Algorithms under LT Model.*

ticularly in more extensive and inter-connected networks, such as Gnutella30 and NetHept. The consistent dominance across both models highlights the effectiveness of the hybrid and metaheuristic strategies.

This illustrates that HybridACO, by integrating attribute-based scouting with ant colony optimization, robustly maximizes the influence spread more efficiently than traditional heuristics or metaheuristics. The increasing trend in influence spread with larger seed sets confirms the scalability of the model. Its strong performance across varying network structures and diffusion models demonstrates its potential for real-world applications where maximizing social influence is critical.

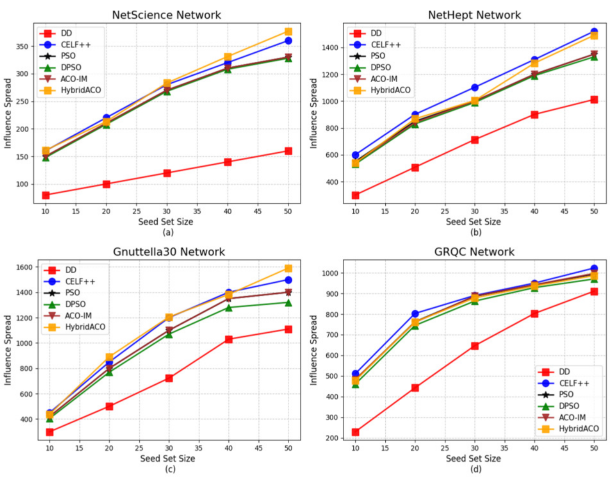

Fig.9: *Comparison of Influence Spread across Benchmark Networks Using Six Algorithms under IC Model.*

(c) Running Time Analysis:

Fig. 10 illustrates the running times (**Table 5**) of the six influence maximization algorithm variants CELF++ [2], DPSO [23], PSO [11], ACOIM [10], HybridACO, and DD [4] across four real-world social networks with a seed set size of $k = 50$. The x-axis represents the algorithms applied to the NetScience, NetHept, GRQC, and Gnutella30 networks, whereas the y-axis shows their running times on a logarithmic scale. CELF++ consistently demonstrated the highest running times across all the networks, indicating poor scalability and computational inefficiency. In contrast, DD consistently achieved the lowest running times, demonstrating its superior efficiency and adaptability for large-scale networks, however it produced the lowest influence spread. Other algorithms, including DPSO, PSO, ACOIM, and HybridACO, offer a trade-off between the computational cost and performance. This balance of efficiency and computational cost positions HybridACO as a promising approach for influence maximization, particularly in scenarios demanding scalability without compromising effectiveness.

Table 5: *Running Time (in seconds).*

Algorithm	NetScience	NetHept	GRQC	Gnutell30
CELF++	180	623	913	1231
DPSO	73	219	387	683
PSO	67	195	314	616
ACOIM	69	203	354	623
DD	43	163	293	577
HybridACO	28	63	108	151

(d) Statistical Analysis:

Statistical analysis conducted using the Friedman test [27] and Nemenyi [28] post-hoc test provides a detailed evaluation of the performance of six state-of-the-art (SOTA) algorithms: DD [4], CELF++ [2], PSO [11], DPSO [23], ACOIM [10], and HybridACO for maximum influence spread.

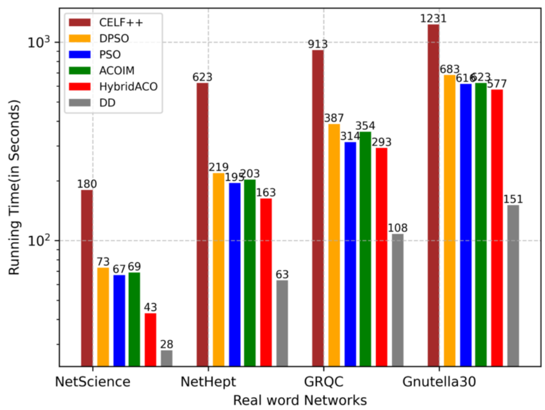

Fig.10: *Running times of various algorithms on four large real-world networks.*

The Friedman test is a non-parametric statistical test used to detect the performance differences across various algorithms. The Friedman test is particularly suitable for comparing multiple algorithms and methods across different datasets. The Friedman test results (F-Statistic = 24.0000, $p = 0.0001$) confirmed that there was a statistically significant difference in performance among the algorithms based on the alternate hypothesis (H_1), rejecting the null hypothesis (H_0). The ranking results show that HybridACO achieves the best performance with an average rank of 1.80, followed by CELF++ (2.00) (**Table 6**).

Table 6: *Average ranking of the algorithm calculated using the Friedman test.*

SN	Algorithm	Average Rank
1	HybridACO	1.80
2	CELF++	2.00
3	ACOIM	2.50
4	DPSO	4.10
5	PSO	4.60
6	DD	6.00

The Nemenyi post-hoc test was applied to perform pairwise comparisons among all algorithms following the primary statistical test. The p-value matrix from the heatmap (**Fig. 11**) shows that HybridACO is a highly competitive algorithm that significantly outperforms DD ($p = 0.0052$), while showing no statistically significant differences from CELF++ ($p = 0.9999$), ACOIM ($p = 0.99$), PSO ($p = 0.17$), or DPSO ($p = 0.38$). It means that HybridACO, CELF++, ACOIM, PSO, and DPSO all have similar performance levels, whereas only DD is clearly weaker and consistently outperforms the others. HybridACO ranks as the best algorithm in Table 6 because it achieves the highest influence spread, and the post-hoc Nemenyi test shows that it performs significantly better than DD, the weakest method.

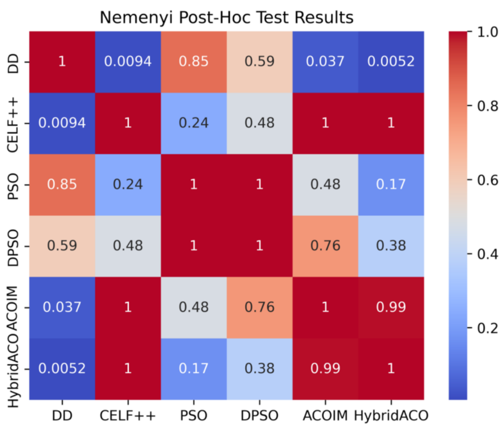

Fig.11: *Heatmap of Pairwise Comparison of Algorithms Using Nemenyi Post-Hoc Test.*

CELF++ and ACOIM are similar, with results that are statistically similar to those of HybridACO. DPSO and PSO performed moderately well, while DD ranked lowest in both statistical ranking and pairwise comparisons. This analysis confirms HybridACO's statistically superior performance to DD while maintaining competitive performance with other state-of-the-art (SOTA) algorithms. All p-values from the heatmap support these conclusions.

(e) Summary:

The following table (**Table 7**) summarizes the performance of each algorithm across various metrics and highlights the unique characteristics of HybridACO as a balanced and scalable solution.

Table 7: *Comparison of Influence Maximization Algorithms across Key Metrics.*

Algorithm	Influence Spread	Running Time	Robust-ness	Scalabil-ity
CELF++ [2]	H	VH	M	L
DD [4]	M	L	M	H
PSO [11]	M - H	M	H	M
DPSO [23]	H	M	H	M
ACOIM [10]	H	M	H	M - H
HybridACO (Proposed)	H - VH	M	H	M - H

H: High, VH: Very High, M: Medium, L: Low

HybridACO consistently achieves high to very high **influence spread**, as demonstrated through extensive experiments on synthetic and real-world networks, supported by robust statistical tests that rank it highest with significant differences from weaker algorithms such as DD. CELF++ exhibits the longest running time because of the computationally expensive Monte Carlo simulations, whereas DD has the shortest running time, which is attributed to its simple heuristic approach. The HybridACO **running time** is rated medium, reflecting its heuristic/metaheuristic balance, which is more efficient than CELF++ but comparable to PSO, DPSO, and ACOIM, as supported by the empirical data (**Table 5**). It's **robustness** is rated high, showing stable performance across diverse network structures and diffusion models, comparable to PSO and DPSO. The **Scalability** of HybridACO ranges from medium to high, effectively handling large-scale networks and outperforming the poor scalability of CELF++ while maintaining a balanced approach compared to the fast but simplistic DD method. These evaluations collectively validate the qualitative ratings in **Table 7**, and HybridACO offers an excellent balance of high influence spread, robustness, and scalability, making it a strong choice compared with other algorithms.

7. CONCLUSIONS

HybridACO provides an excellent balance between efficiency and performance for influence maximization. Unlike CELF++, which runs slowly on large networks, HybridACO significantly reduces the computational costs while maintaining strong influence spread results. It outperforms traditional heuristic-based methods by integrating adaptive strategies, thereby rendering it more suitable for large-scale networks. Compared with DD, which has the lowest running time but produces a minimal influence spread, HybridACO strikes an optimal trade-off between scalability and effectiveness. HybridACO is a promising algorithm for real-world influence maximization applications, where both efficiency and effectiveness are critical.

In the future, further optimization of HybridACO can be explored using machine-learning models to learn the network structure adaptively. Additionally, its application can be extended to diverse real-world domains, including marketing, healthcare, and disaster management, where influence maximization plays a critical role. Future research could focus on enhancing its performance in dynamic networks to ensure sustained reliability as network conditions evolve.

AUTHOR CONTRIBUTIONS

Conceptualization, M.R. and I.P.; methodology, M.R.; validation, M.R. and I.P.; formal analysis, M.R.; investi-gation, M.R.; data curation, M.R.; writing-original draft preparation, M.R. and I.P.; writing-review and editing, M.R. and I.P.; visualization, I.P.; supervision, I.P. All authors have read and agreed to the published version of the manuscript.

References

[1] Y. C. Chen, W. Y. Zhu, W. C. Peng, W. C. Lee and S. Y. Lee, "CIM: Community-based influence maximi-zation in social networks," *ACM Transactions on In-telligent Systems and Technology (TIST)*, vol. 5, no. 2, pp. 1–31, 2014.

[2] A. Goyal, W. Lu and L. V. Lakshmanan, "CELF++: Optimizing the greedy algorithm for influence max-imization in social networks," in *Proceedings of the 20th International Conference Companion on World Wide Web*, pp. 47–48, 2011.

[3] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 420–429, 2007.

[4] W. Chen, Y. Wang and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 199–208, 2009.

[5] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the Sev-enth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 57–66, 2001.

[6] . Kempe, J. Kleinberg and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146, 2003.

[7] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 259–271, 2006.

[8] W. Chen, C. Wang and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1029–1038, 2010.

[9] S. S. Singh, A. Kumar, K. Singh and B. Biswas, "LAPSO-IM: A learning-based influence maximization approach for social networks," *Applied Soft Computing*, vol. 82, p. 105554, 2019.

[10] S. S. Singh, K. Singh, A. Kumar and B. Biswas, "ACO-IM: Maximizing influence in social networks using ant colony optimization," *Soft Computing*, vol. 24, no. 13, pp. 10181–10203, 2020.

[11] M. Gong, J. Yan, B. Shen, L. Ma and Q. Cai, "Influence maximization in social networks based on discrete particle swarm optimization," *Information Sci-ences*, vol. 367, pp. 600–614, 2016.

[12] C. Salavati and A. Abdollahpouri, "Identifying influential nodes based on ant colony optimization to maximize profit in social networks," *Swarm and Evo-lutionary Computation*, vol. 51, p. 100614, 2019.

[13] Z. Aghaee, M. M. Ghasemi, H. A. Beni, A. Bouyer and A. Fatemi, "A survey on metaheuristic algo-rithms for the influence maximization problem in so-cial networks," *Computing*, vol. 103, pp. 2437–2477, 2021.

[14] M. Sumathi, N. Vijayaraj, S. P. Raja and M. Raj-kamal, "HHO-ACO hybridized load balancing tech-nique in cloud computing," *International Journal of Information Technology*, vol. 15, no. 3, pp. 1357–1365, 2023.

[15] J. K. C. Revanna and N. Y. B. Al-Nakash, "Metaheuristic link prediction (MLP) using an AI-based ACO-GA optimization model for solving vehicle routing problem," *International Journal of Infor-mation Technology*, vol. 15, no. 7, pp. 3425–3439, 2023.

[16] Y. Y. Raghav and V. Vyas, "ACBSO: A hybrid solution for load balancing using ant colony and bird swarm optimization algorithms," *International Jour-nal of Information Technology*, vol. 15, no. 5, pp. 2847–2857, 2023.

[17] Q. Jiang, Z. Peng, X. Xie, K. Du, G. Hu and Y. Liu, "Preparation of high active Pt/C cathode electrocatalyst for direct methanol fuel cell by citrate-stabilized method," *Trans. Nonferrous Metals Soc. China*, vol. 20, no. 1, pp. 127–132, 2010.

[18] J. C. Liang, Y. J. Gong, X. K. Wu and Y.

Li, "Customized influence maximization in attributed social networks: Heuristic and metaheuristic algorithms," *Complex and Intelligent Systems*, vol. 10, no. 1, pp. 1409–1424, 2024.

[19] I. Khatri, A. Choudhry, A. Rao, A. Tyagi, D. K. Vishwakarma and M. Prasad, "Influence maximization in social networks using discretized Harris Hawks Optimization algorithm," *Applied Soft Computing*, vol. 149, p. 111037, 2023.

[20] M. Roy and I. Pan, "Most Influential Node Selection in Social Network using Genetic Algorithm," *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, Kolkata, India, pp. 214-220, 2018.

[21] H. Huang, H. Shen, and Z. Meng, "Community-based influence maximization in attributed networks," *Appl. Intell.*, vol. 50, no. 2, pp. 354–364, 2020

[22] D. Bucur and G. Iacca, "Influence maximization in social networks with genetic algorithms," in *Applications of Evolutionary Computation: 19th European Conference Proceedings, Part I, Springer Interna-tional Publishing*, pp. 379–392, 2016.

[23] J. Tang, R. Zhang, Y. Yao, F. Yang, Z. Zhao, R. Hu and Y. Yuan, "Identification of top-k influential nodes based on enhanced discrete particle swarm optimization for influence maximization," *Physica A: Statistical Mechanics and its Applications*, vol. 513, pp. 477–496, 2019.

[24] M. Roy and I. Pan, "Lazy forward differential evolution for influence maximization in large data networks," *SN Computer Science*, vol. 1, no. 2, p. 107, 2020.

[25] H. Zhou, Y. Li and Z. Wu, "A hybrid ACO-SA ap-proach for influence maximization in social net-works," *Journal of Computational Science*, vol. 54, p. 101368, 2021.

[26] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," in *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, Nov. 2006

[27] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of var-iance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[28] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton Univ., Princeton, NJ, 1963.

Mithun Roy serves as an Assistant Professor in the Department of Computer Science and Engineering at Siliguri Institute of Technology, Siliguri, West Bengal, India. He earned his B.Sc. in Mathematics (Honours) from the University of Calcutta in 2000, followed by an M.Sc. in Computer Science from Annamalai University, Tamil Nadu, in 2009. He later completed his M.Tech. in Computer Science and Engineering from Maulana Abul Kalam Azad University of Technology, West Bengal, in 2013. Mithun is a Member of the Association for Computing Machinery (ACM), USA, and a Life Member of The Institution of Engineers (India). His research interests primarily focus on Machine Learning and Social Network Analysis.

Indrajit Pan is a Professor in the Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) at RCC Institute of Information Technology, Kolkata, West Bengal, India. He obtained his B.E. in Computer Science and Engineering with Honors from the University of Burdwan in 2005, followed by M.Tech. in Information Technology from Bengal Engineering and Science University, Shibpur, where he received the University Medal. He was awarded his Ph.D. in Engineering from the Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, in 2015. Dr. Pan is a Senior Member of both IEEE (USA) and ACM (USA), a Fellow of IETE, and a Life Member of CSI and The Institution of Engineers (India). His research interests include Digital Microfluidic Biochips, Information Science, and Cloud Computing. He has co-authored numerous journal and conference papers, Indian patents, copyrights, and edited book volumes, including international conference proceedings.