

## Improving Influence Propagation in Social Networks using Fitness Sharing Algorithm with Dynamic Sharing Radius

**Fereshteh Kiani**

Sheikhbahaee University, Isfahan, Iran

[F.Kiani@shbu.ac.ir](mailto:F.Kiani@shbu.ac.ir)

**Navid Kaveh**

Sheikhbahaee University, Isfahan, Iran

[NavidKaveh@shbu.ac.ir](mailto:NavidKaveh@shbu.ac.ir)

**Mehdi Bateni**

Sheikhbahaee University, Isfahan, Iran

[Bateni@shbu.ac.ir](mailto:Bateni@shbu.ac.ir)

preserving population diversity. The test results of the proposed algorithm on different datasets show that this method improves the accuracy of finding the most influential people in the issue of influence maximization compared to other common algorithms.

**Keywords:** Influence Propagation, Influence Maximization, Fitness Sharing Algorithm with Dynamic Sharing Radius, Linear Threshold Model

### Abstract

In social networks, related people are influenced by each other because people share their ideas and views on different issues. If a number of people in a network adopt a particular behavior or belief, this behavior or belief is spread in the network due to the social relationships within the network. This phenomenon is called Influence Propagation. One of the most important issues in influence propagation optimization is the issue of influence maximization. In influence maximization, the goal is to find k subset of members of the social network so that by activating them, under an information diffusion model, the largest number of network members will be affected by information. The purpose of this study is to provide a solution to find the most influential people using the fitness sharing algorithm with dynamic sharing radius and under the Linear Threshold Model. The proposed solution is one of the meta-heuristic solutions in which the genetic algorithm is used. The proposed algorithm prevents premature convergence by modifying the genetic algorithm and turning it into a multimodal mechanism, while

### 1. Introduction

A social network is a structure shaped by related people. Analyzing social networks is about evaluation process of social network structure as a graph of nodes that are connected together by connective lines. These connection lines can be friendship relation, virus transmission, message exchanging, or any relation [1]. In social networks, people share their ideas and viewpoints about different issues with others, and related people are affected by each other. Influential people are able to change thought, behavior and believe of many people that are existed in network. Spreading believes or behavior of considerable amount of network users is called "influence propagation" [2].

Optimization of influence propagation is one of the most important issues in analyzing social networks. Based to different factors such as number of initial activated nodes in the beginning of propagation process, the number of expected activated nodes in the end of propagation process and required time for propagation process, different objective functions are defined for influence propagation optimization problem. The problem of influence maximization (IM) is among considered problems about influence

propagation optimization. The aim of influence maximization is to finding a subset with custom  $k$  nodes, as initial nodes to initiate propagation in a social network, such that if information propagation on the network is under a model of information propagation and to be started through these selected initial nodes, then influence propagation is maximized in network and the maximum number of network members are affected. This subset with  $k$  members is called “seed set” [3].

Information diffusion models define the total propagation process. The Linear Threshold (LT) model is one of the most applicable propagation models in influence maximization problem. In this model, each node is placed in active or inactive state and there is a threshold for each one. Threshold  $\theta(v)$  refers to bias of a node to accept a behavior or idea of its associates. Node  $v$  is connected to its neighbor node  $w$  by an edge with the weight of  $b_{v,w}$ , such that  $\sum b_{v,w} \leq 1$ . Entered edges into each node are representatives of the influence of neighbor nodes on that node. In the beginning of propagation process, an initial set of nodes is activated. In each step all active nodes of last steps are kept active and new activated nodes are added into this set and in the next step, they will flip all of their inactive neighbors. If the total weight of active neighbors of a node exceeds of its threshold, then that node is also to be activated and added to the set. This will continue until all nodes to be activated or no new node activates [2].

In order to improving the influence propagation in social networks, various algorithms are presented. A group of algorithms use greedy methods. Greedy methods have good precision and are very optimum in terms of influence propagation. But these methods are not scalable due to their high runtime. Because of ever increasing size of online social networks, finding most influential network members depended on scalable methods. So, different heuristic methods are introduced to increase scalability of algorithms. Heuristic algorithms are optimum in terms of run time and therefore are scalable but precision is not quarantined. The innovation of this research is in using fitness sharing algorithm with dynamic sharing radius

(FSDSR) under linear threshold model. Recommended solution belongs to meta-heuristic solutions and use genetic algorithm. One of the genetic algorithm disadvantages is that after a number of iterations, due to loosing population diversity, is tend to be tapped into local optimum. Recommended algorithm prevents early convergence also keeping population diversity by changing evolutionary genetic algorithm.

Paper organizes as follow: Section 2 reviews the related works on influence maximization problem. Section 3 introduces the Niching technique. In section 4 the recommended method is introduced and in section 5 the results of evaluating recommended method on different data sets are presented. Section 6 consists of paper conclusion and in the last are references.

## 2. Related works

Kempe et al. [2], for the first time model the influence maximization problem as an optimization problem and prove that this problem belongs to NP-Hard problems under linear threshold and independent cascade propagation models. They introduce a greedy algorithm with provable approximation named KGA by considering submodularity and monotone feature of influence maximization function. Initially, in KGA algorithm amount of active initial nodes are specified. Then, a node activated and diffusion model are applied in custom frequency. After examining all nodes, a node is selected that activates a greater number of nodes. Each of survivor nodes are added to this node and simulation is running to select the best node to add into initial node set. This process iterates to select  $k$  node. Approximate results of KGA greedy algorithm has minimum precision  $\left(1 - \frac{1}{e} - \epsilon\right)$ . Main issue about greedy algorithm is Monte Carlo simulation and its low-level productivity. Each evaluation of influence propagation by Monte Carlo simulation can slow algorithm. This simulation needs  $R$  simulation iteration independent of propagation process. In order to reach more functionality for greedy algorithm,  $R$  is considered 20000.

Although the greedy algorithm has been able to introduce the best solution but it is not practicable to use it in big networks. Therefore, different heuristics are studied to introduce a scalable solution with greedy approach. The aim of all of these solutions is to improving runtime and increasing algorithm scalability. Leskovec et al. [4], introduced CELF method to improve algorithm scalability. In this method, only influence propagation of some node is calculated by using a priority queue and submodularity feature of influence propagation function. This results to decreasing the number of evaluations for greedy algorithm and consequently, considerably improves algorithm run time. Goyal et al. [5], proposed an efficient algorithm for influence maximization problem under linear threshold model called "SIMPATh". In this algorithm path counting techniques are used. The possibility of the path between each two nodes is the crossing of the probability of each edge. The spread of the influence from a node can be computed by summing the weights (e.g. probabilities) of all simple paths originating from it. Chen et al. [6], introduced a community-based algorithm called "CGA". This algorithm has two parts. At first, it reaches a new graph by pruning the graph and identifying communities. Then, greedy algorithm searches the set of seed nodes in new graph. Because the influence of an active node just is computed in the same community, the algorithm runtime is improved relative to greedy algorithm. Also, minimizing the search space makes the algorithm scalable. Meta-heuristic algorithms are a group of approximate algorithms that try to find a way to efficiently and effectively search response space through the combination of initial basics of heuristics. Genetic algorithm (GA) is one of them. Zhang et al. [7], proposed a genetic algorithm for influence maximization problem. GA initially generates multiple k-node seed sets, called "populations". Influence propagation of the seed sets in assumed diffusion model is considered as fitness function. In each iteration, the optimum seed set is selected by using fitness function. Although genetic algorithm is a multipoint local search method but after combining population together, gradually algorithm

reaches convergence and response diversity decreases. Kaveh et al. [8], proposed deterministic crowding algorithm to eliminate early convergence issue of genetic algorithm and also increase diversity of solutions. This method concentrates on replacement process and makes some changes in survivor selection step in genetic algorithm. In this algorithm, offspring compete with their parents and if they have higher fitness will replace with their nearest parent. In this case, population diversity is kept and also prevents early convergence to local optimum points.

Singh et al. [9], introduce an algorithm based on automatic learning and particle swarm optimization algorithm, called "LAPSO-IM". Particle swarm optimization algorithm is a collective search algorithm that is modeled based on social behavior in bird flock. In this algorithm, the function of particle speed is defined based on automatic learning to prevent algorithm early convergence. Zareie et al. [10], proposed an algorithm based on gray wolf optimization algorithm to find more effective people of network. They are modeled the influence maximization problem as an optimization problem with cost functions like node influences and their distance.

### 3. Introducing Niching technique

In the real world, physical space results in evolution of a species and its more diversity. In other words, people of the same species that are evolution in different physical space, have different shape. In many optimization problems, search space has unequal peaks where normal solutions just find one of them as global optimum. A problem with more than one optimum called "multimodal problem" [11].

In multimodal problems, the area of discontinuity between the optimizations is called "niche". Niching term refers to natural ecosystems where different species try to survive in order to evolve and fill different niches. Niches are considered as stable subpopulations that each covers a zone of search space. Using Niching technique in evolutionary algorithms makes it possible to finds and maintains multiple peaks simultaneously. Main objective of Niching technique is to let an algorithm to find and

keep many solutions. In population-based algorithms, this definition normally means to keep population diversity. This method potentially prevents the algorithm from rapid convergence to local optimizations. Therefore, Niching methods are applied even when the problem only has one optimum [8].

#### 4. Proposed method

FSDSR algorithm is the proposed method for solving influence maximization problem. The

main structure of proposed algorithm is provided in algorithm 1.

**Input:** Graph  $G = (V, E)$ , Seed set size  $k$ , Population size  $m$ , Crossover Probability  $p_c$ , Mutation Probability  $p_m$ ;  
**Output:** Seed set  $A$ ;

**Input:** Graph  $G = (V, E)$ , Seed set size  $k$ , Population size  $m$ , Crossover Probability  $p_c$ , Mutation Probability  $p_m$ ;

**Output:** Seed set  $A$ ;

- 1: Randomly initialize the population:  $p_i = \{v_1, \dots, v_k\}$  ( $i = 1, 2, \dots, m$ );
- 2: Evaluate fitness of each individual,  $\sigma(p_i)$ ;
- 3: **while** Termination Condition **do**
- 4: Fitness Sharing: Divide population into niches and shared fitness of individuals;
- 5: Selection of parents according to  $f'(p_i)$  with normalized roulette-wheel selection;
- 6: **if** random  $rnd_c < p_c$  **then**
- 7: Crossover parents  $p_i$  and  $p_j$ ; Create new offspring  $o_i$  and  $o_j$ ;
- 8: **end if**
- 9: **for** each offspring,  $o_i$  and  $o_j$  **do**
- 10: **if** random  $rnd_m < p_m$  **then**
- 11: Mutation and create new offspring  $o'_i$  and  $o'_j$  from  $o_i$  and  $o_j$ ;
- 12: **end if**
- 13: **end for**
- 14: Evaluate fitness of each offspring  $o_i$  and  $o_j$ ;
- 15: Selection of survivor with age-based selection;
- 16: Select  $\max|\sigma(p_i)|$ ,  $A = \operatorname{argmax}(|\sigma(p_i)|)$ ;
- 17: **end while**
- 18: **return**  $A$ ;

#### Algorithm 1: Structure of proposed algorithm

Algorithm inputs include a network, size of the seed sets, the population size, crossover rate and mutation rate. Algorithm output is a seed set that have maximum influence number in network. Proposed algorithm starts with random generating an initial population. Population includes  $m$  chromosomes with  $k$  length. Genes represent the node's number and a chromosome is made by putting randomly  $k$  genes together. Each chromosome is a

probable solution set for problem (Line 1). Fitness of each chromosome (seed set) is obtained by fitness function. Because research subject is influence maximization under linear threshold model, linear threshold model is considered as fitness function of proposed algorithm. In other words, fitness amount of each chromosome is the number of nodes that are influenced and activated by that seed set under linear threshold model (Line 2). In

proposed algorithm the sharing radius is obtained dynamically and by using chromosomes length and distance. Using this radius, population is divided into different subgroups and fitness of people is varied based on their similarity, thereby the diversity is kept (Line 4). The procedure of this algorithm is described in section 4.1. Parent selection operator, selects two parents by roulette wheel selection and based on sharing radius and changed fitness of chromosomes (Line 5). Two selected parents by using two-points crossover operator, create two offspring (Lines 6 to 8). In order to making differentiation between offspring and parents also producing new values in offspring, mutation operator applied on each offspring (Lines 9 to 13). Again, fitness of all chromosomes is computed (Line 14) and survivor selection operator replace two new chromosomes in population based on age (Line 15). At the end of each iteration, the chromosome with maximum fitness is stored as the best solution (Line 16). If the influence number of an optimal seed set remains at a steady state, the algorithm ends (Line 3).

#### 4-1. Sharing fitness with dynamic radius

$$\begin{aligned} \hat{f}(i) &= \frac{f(i)}{\sum_{j=1}^m sh(d(i, j))} \\ sh(d(i, j)) &= \begin{cases} 1 - \left( \frac{d(i, j)}{\sigma_{share}} \right)^\alpha & \text{if } d(i, j) < \sigma_{share} \\ 0 & \text{Otherwise} \end{cases} \end{aligned} \quad (1) \quad (2)$$

Important parameter in fitness sharing algorithm is sharing radius. Sharing radius obtains based on population size and search space, because the size of population determines maximum number of peaks. Distance between two individuals in population, shows their similarity. In this research the distance between two chromosomes is difference of chromosome length and the number of similar genes in two chromosomes.

Fitness sharing algorithm making some changes before the parent selection step in genetic algorithm. In this method, the fitness of people with higher fitness that has higher chance to bias total population to themselves is decreased and contrary. Therefore, the weaker individuals that probably are eliminated during the time, are kept [12]. Formulas 1 and 2 show their procedure. In Formula 1, raw fitness of individual  $i$  is  $f(i)$ , before selection step, this fitness is changed to  $\hat{f}(i)$  by using sharing function. In this formula,  $m$  is the size of populations. In Formula 2, the distance of individual  $i$  from others is  $d(i, j)$  and  $sh(d(i, j))$  is sharing function. Sharing function describes the similarity of various individuals. Niche size is determined by parameter  $\sigma_{share}$  that is called “sharing radius”. If the distance of two individuals is smaller than sharing radius, it means that these two individuals are placed in the same niche. The people in the same niche are considered similar and their fitness should be changed.  $sh(d(i, j)) = 0$  means that two persons  $i$  and  $j$  belongs to different niches. The constant  $\alpha$ , controls the magnitude of sharing and normally set to 1.

Assuming that each niche is encompassed in  $p$ -dimension search space and  $q$  be the number of current peaks in problem space or the number of niches, then it is possible to obtain sharing radius by using formulas 3 and 4. To dynamically compute sharing radius, initially distance between chromosomes is obtained and then according to number of niches and problem dimension, niche radius is computed. In formula 3,  $x_{k,max}$  is upper bound and  $x_{k,min}$  is lower bound of search space [13].

$$r = \frac{1}{2} \sqrt{\sum_{k=1}^n (x_{k,max} - x_{k,min})^2} \quad (3)$$

$$\sigma_{share} = \frac{r}{\sqrt[p]{q}} \quad (4)$$

$$= \frac{\sqrt{\sum_{k=1}^p (x_{k,max} - x_{k,min})^2}}{2 \sqrt[p]{q}}$$

#### 4-2. Algorithm operators

After calculating the fitness of chromosomes, genetic algorithm operators are used to exert changes on them. Proposed algorithm uses Roulette wheel operator to select parents. In this operator, raw fitness of individuals is replaced by shared fitness of them. In this case, initially the sum of shared fitness of all individuals is obtained. The probability of selecting each chromosome is computed by fitness summation and then probability collectives of each individual is obtained. In order to select parent, a random number between 0 and 1 is picked up. From the beginning of population this number is compared with probability collectives of each individual. The first one with greater probability collectives than random number is selected as one of the parents. This process iterates to select the second parent.

Crossover operator, combines two sub-string of population to obtain strings with better chromosomes than their parents. In proposed algorithm, crossover rate is considered 0.8 and using two-point crossover. Two-point crossover in compare to single-point crossover has more exploration ability. First, to specify the replacement place of chromosomes, two points  $f_1$  and  $f_2$  are selected randomly. Point  $f_1$  is a number between zero and chromosome length and point  $f_2$  should be a number between  $f_1$  and chromosome length. In two selected parents the genes between these two points are swapped if are not existed in another chromosome. In this case, two new offspring are created due to genes replacement between two breaking points in parents.

The mutation operator, by randomly changing the number of genes, causes diversity in the population of chromosomes

and converges to a better optimal answer in the search space of the problem. In the proposed algorithm, the mutation rate is assumed to be equal to 0.2. The input of the mutation operator is two offspring created as a result of the crossover operator. If a mutation is performed on the incoming chromosome, two new offspring are returned as output. First, one node in the chromosome and one node in the network are randomly selected. The node selected from the network is compared to all nodes on the chromosome and, if not duplicated, replaces the node selected on the chromosome. In this way, a new offspring is produced.

Survivor selection determines that who should be eliminated and who should be kept in next generation. In age-based survivor selection method, the fitness of individual in current population is not important and each individual is permitted to attend in population based on limited number of generations that can reproduce. In this operator, initially the oldest members of the population are kicked out of the population and the ages of the rest of the members are incremented by one. Finally eliminated individuals are replaced by two new offspring. In this way, an iteration of algorithm is done and finally, the best solution stores. Above mentioned procedure iterates again to uniform the number of activated nodes by the seed sets.

#### 5. Evaluation results

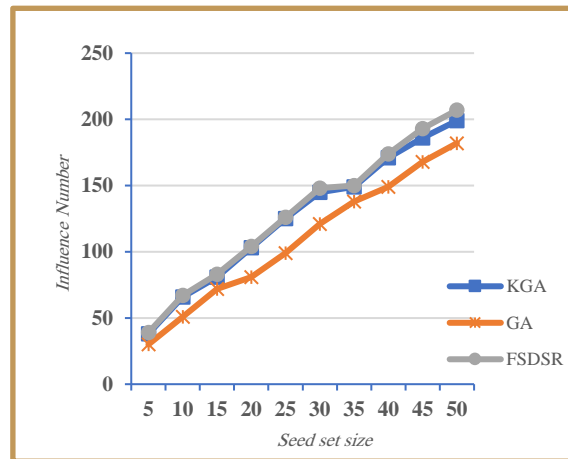
FSDSR algorithm is run in four networks for evaluation. To solve the influence maximization problem each dataset is tested through proposed algorithm and the results are compared with greedy algorithm (KGA) and

genetic algorithm (GA). Netscience network [14] is a co-authorship network of scientists. This dataset contains all components of the network, for a total of 1589 scientists that study on network theory. Email network [15] is an internal email network of a mid-sized manufacturing company with 167 nodes. This dataset includes sender, receiver and sending time. Period includes information of 9 month completely from the beginning of 2010. Infectious network [16] is the network of science gallery visitors. In this network, face to face contacts of 410 people of visitors are showed by edges. This dataset contains the daily dynamic contact networks collected during the Infectious SocioPatterns event that took place at the Science Gallery in Dublin, Ireland, during the artscience exhibition INFECTIOUS: STAY AWAY. Ucsocial network [17] is local messaging network of California University. This network is directive and includes 1899 users of an online community of students.

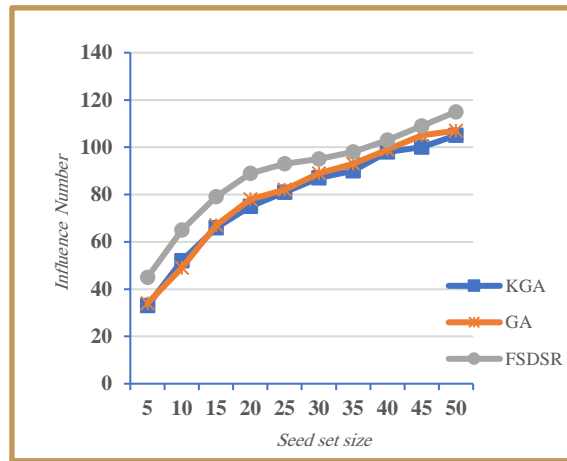
The experimental environment is a personal computer with Intel® Pentium® G4400, 3.30GHz CPU and 6GB RAM. The size of seed set is a multiple of 5 and maximum 50 nodes are considered. The threshold of nodes

is selected randomly and between 0 and 1. In all tests, amount of influence propagation is examined under a fixed threshold. Edges are weighting uniformly and influence propagation is obtained under linear threshold model.

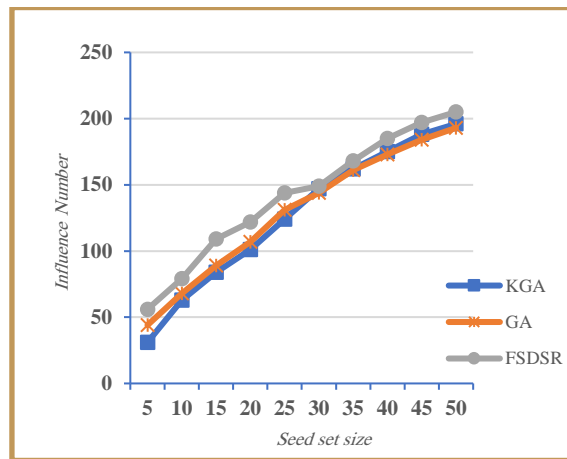
In figure.1 the results of the test of influence number on Netscience network are presented. This network includes 1,589 nodes and 2,742 edges. According to the number of nodes and edges and network density, it is obvious that Netscience is a sparse network. Therefore, in this network the genetic algorithm becomes trapped in local optima and is not able to search all the space, so, results of this algorithm are worse than greedy algorithm. FSDSR algorithm divide population into multiple subpopulations by using Niching technique and sharing radius and as a result, algorithm can search in all spaces. Comparing the results shows that this algorithm has better performance than both greedy and genetic algorithms. In the small seed sets, difference between the results of proposed algorithm and greedy algorithm is not specified and results are so closed. But by increasing the size of the seed sets, the differences between responses grows.



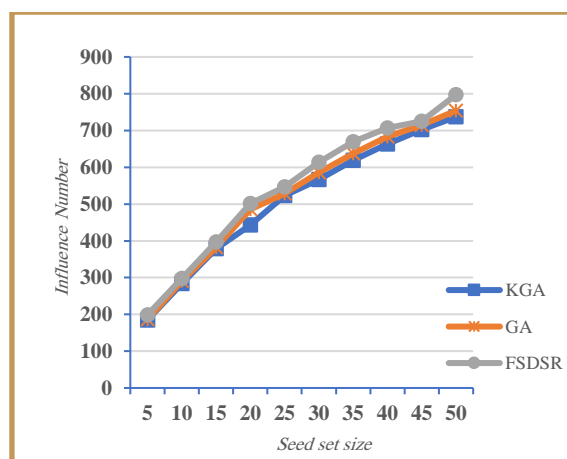
**Figure.1: The results of the tests for influence number on Netscience network**



**Figure.2: The results of the tests for influence number on Email network.**



**Figure.3: The results of the tests for influence number on Infectious network.**



**Figure.4: The results of the tests for influence number on Ucsocial network.**



In Email network in figure.2, results of genetic algorithm are so closed to greedy algorithm. In this network, FSDSR algorithm improves results better than genetic algorithm. Especially, when the size of the seed set is smaller than or equal to 25 then the number of influenced nodes grows considerably. Also, for the seed set greater than 25, FSDSR algorithm has better results than both greedy and genetic algorithms.

In Infectious network in figure.3, in the small seed sets, the results of genetic algorithm are better than greedy algorithm and, in the seed sets greater than 25, the results are similar and sometime worse than greedy algorithm. In this network although the results of genetic algorithm are somewhat like to greedy algorithm but FSDSR algorithm has gained better results than both greedy and genetic algorithms. If the size of the seed set is smaller than 30, then result improvement in FSDSR algorithm is considerable. While in the seed sets greater than or equal to 30, the percent of results improvement in proposed method is decreased, but still FSDSR algorithm has better performance than the other two algorithms.

In figure.4 Ucsocial network, genetic algorithm has good performance and gains better results than greedy algorithm. The results of FSDSR algorithm are better than two other algorithms, especially when the size of the seed sets is greater than 30, then the proposed algorithm has much better performance.

## 6. Conclusion

The influence maximization problem includes finding a small subset of nodes where activating them results in maximum influence propagation. Many algorithms are introduced to solve the influence maximization problem. A group of algorithms are optimum in terms of amount of influence propagation but due to high runtime, are not scalable. Another group of algorithms are optimum in terms of runtime but are not optimum in terms of maximum influence. The innovation of this study is in using fitness sharing algorithm with dynamic sharing radius for influence maximization problem. Proposed algorithm (FSDSR), greedy algorithm (KGA) and genetic

algorithm (GA) are tested on four networks and results are compared in terms of the influence number of seed sets. Results indicate that the proposed algorithm on average improves, respectively, 6.3% and 9.2% the results of influence maximization problem than greedy and genetic algorithms.

It is suggested that for future works, test the proposed algorithm on other diffusion models except for linear threshold (LT) model. Also, in order to propagate influence of time-dependent events, it is possible to convert the problem into multi-objective problem and then, solve it.

## References

- [1] S. Wasserman and K. L. M. Faust, *Social network analysis: Methods and applications*, vol. 8. Cambridge: Cambridge University Press, 1994.
- [2] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 03*, pp. 137–146, 2003.
- [3] R. Narayanam and Y. Narahari, "A Shapley Value-Based Approach to Discover Influential Nodes in Social Networks," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 130–147, 2011.
- [4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Vanbriesen, and N. Glance, "Cost-effective outbreak detection in networks," *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 07*, pp. 420–429, 2007.
- [5] A. Goyal, W. Lu, and L. V. Lakshmanan, "SIMPAT: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model," *2011 IEEE 11th International Conference on Data Mining*, pp. 211–220, 2011.
- [6] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 09*, vol. 67, no. 1, pp. 199–208, 2009.
- [7] K. Zhang, H. Du, and M. W. Feldman, "Maximizing influence in a social network:

Improved results using a genetic algorithm,” *Physica A: Statistical Mechanics and its Applications*, vol. 478, pp. 20–30, 2017.

[8] N. Kaveh and M. Bateni, “Influence Maximization in Social Networks using Deterministic Crowding Algorithm,” *International Journal of Innovative Technology and Exploring Engineering Regular Issue*, vol. 8, no. 11, pp. 4258–4265, 2019.

[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] A. Zareie, A. Sheikahmadi, and M. Jalili, “Identification of influential users in social network using gray wolf optimization algorithm,” *Expert Systems with Applications*, vol. 142, p. 112971, 2020.

[11] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Berlin: Springer Berlin, 2016.

[12] D. E. Goldberg and J. Richardson, “Genetic algorithms with sharing for multimodal function optimization,” *Proceedings of the Second International Conference on Genetic algorithms and their Application*, Hillsdale, NJ, USA, pp. 41–49, 1987.

[13] T. Iwase, R. Takano, F. Uwano, H. Sato, and K. Takadama, “The Bat Algorithm with Dynamic Niche Radius for Multimodal Optimization,” *Proceedings of the 2019 3rd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence - ISMSI 2019*, 2019.

[14] M. E. J. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical Review E*, vol. 74, no. 3, 2006.

[15] R. Michalski, S. Palus, and P. Kazienko, “Matching Organizational Structure and Social Network Extracted from Email Communication,” *Business Information Systems Lecture Notes in Business Information Processing*, pp. 197–206, 2011.

[16] “DATASET: Infectious SocioPatterns dynamic contact networks,” *SocioPatterns.org RSS*. [Online]. Available: <http://www.sociopatterns.org/datasets/infectio>

us-sociopatterns-dynamic-contact-networks. [Accessed: 01-Aug-2020].

[17] T. Opsahl and P. Panzarasa, “Clustering in weighted networks,” *Social Networks*, vol. 31, no. 2, pp. 155–163, 2009.