Machine Learning and Artificial Intelligence
A.J. Tallón-Ballesteros and C. Chen (Eds.)

doi:10.3233/FAIA200760

Genetically Driven Optimal Selection of Opinion Spreaders in Complex Networks

Alexandru TOPÎRCEANU [a,1]
[a] *Department of Computer and Information Technology, Politehnica University Timişoara, Romania*

Abstract. The problem of influence maximization (IM) represents a major challenge for modern network science, with direct applicability in political science, economy, epidemiology, and rumor spreading. Here, we develop a novel computational intelligence framework (GenOSOS) based on genetic algorithms with emphasis on the optimal layout of spreader nodes in a network. Our algorithm starts with solutions consisting of randomly selected spreader nodes; then, by defining custom original *crossover* and *mutation* operators, we are able to obtain, in a short number of genetic iterations, nearly optimal solutions in terms of the nodes' topological layout. Experiments on both synthetic and real-world networks show that the proposed GenOSOS algorithm is not only a viable alternative to the existing node centrality approach, but that it outperforms state of the art solutions in terms of spreading coverage. Specifically, we benchmark GenOSOS against graph centralities such as node degree, betweenness, PageRank and k-shell using the SIR epidemic model, and find that our solution is, on average, 11.45% more efficient in terms of diffusion coverage.

Keywords. genetic algorithm, influence maximization, computational intelligence, complex networks

1. Introduction

Finding individuals with high social influence is one of the fundamental challenges for network science [1,2,3], and represents a critical issue for better understanding of the market [4], and for predicting political preference [5] as well. In its simplest formulation, IM sets out to select the initial spreader nodes which may influence a maximal number of users in a given network [1]. An important demand faced by IM algorithms is obtaining a balanced trade-off between the accuracy of the solution and the time/memory cost, especially over large-scale networks. Consequently, developing efficient algorithms for IM still represents a challenging research topic.

In terms of the more recent IM state of the art research, we first note the works of Zareie *et al.* [6,7]. Similar to our approach, in [6] the authors suggest that distances between spreaders should be taken into consideration to ensure minimum overlap and maximum coverage of a wider area of the network. Similar in scope, the goal in [7] is to maximize the distance between spreader nodes with the use of gray wolf optimization.

[1] Corresponding Author: Alexandru Topîrceanu, Department of Computer and Information Technology, Politehnica University Timişoara, Timişoara, 300223, Romania; E-mail: alext@cs.upt.ro

Other evolutionary methods used to tackle the IM problem are found in [8,9]. Gong *et al.* [8] propose a local influence criterion for a reliable estimation of the influence propagation in cascade models and use particle swarm optimization (PSO) to optimize local influence criteria. Tang *et al.* [9] use a discrete shuffled frog-leaping algorithm (DSFLA) that combines both deterministic and random walk strategies.

Indeed, compared to our computational intelligence framework, we find also similar genetic methodologies proposed to solve the IM problem in complex networks. Bucur *et al.* [10] define a genetic algorithm approach in which the independent cascade (IC) model is used as a fitness function for nodes. Gong *et al.* [11] make use of a genetic algorithm for community-based influence maximization in social networks. Their idea is to optimize the 2-hop influence spread of nodes to find the most influential nodes. Finally, Cui *et al.* [12] propose degree-descending search evolution (DDSE); this strategy generates a node set whose influence spread is comparable to the degree centrality. The results claimed by the authors are obtained 5x faster than for greedy algorithms.

In this article, we develop a novel computational intelligence framework, called *Genetically driven Optimal Selection of Opinion Spreaders* (GenOSOS), to engage in the IM problem, and provide both qualitative and quantitative means of evaluating the performance of GenOSOS. We first apply state of the art methodology in selecting spreaders based on node centralities (degree, betweenness, PageRank and k-shell), then run the SIR epidemic model [13], and measure the diffusion coverage. The SIR diffusion simulations show that the potential of our solution exceeds expectations by offering superior quantifiable results compared to the state of the art. Compared to the analyzed related work [6,7,8,9,2,10,11,12], this paper brings several important contributions:

- We propose GenOSOS, a genetic algorithm approach for the IM problem, which represents an original attempt for dealing with the trade-off between spreader spacing and diffusion coverage.
- We propose a problem-specific modeling of the population and chromosome representation. Furthermore, we implement the fitness function based on a graph coloring algorithm, which can accelerate the convergence of the spreading process.
- We define an individual (chromosome) as a unique spreader set, bringing along custom implementations of *crossover* and *mutation*.
- We estimate the effectiveness of GenOSOS on synthetic and real-world networks. The experimental results show that our algorithm has competitive performances to similar centrality-based node selection methods.

2. Methods

2.1. Problem Definition

We consider a complex network modeled as an undirected graph $G = \{N, E\}$, where $N = \{n_i\}$ is the node set and $E = \{e_{ij} \,|\, n_i, n_j \in N\}$ is the edge set in the network. Nodes represent individual entities and edges represent social relationships between any two nodes. A node can be marked as a spreader if it has already adopted an opinion, or inactive otherwise.

Thus, the problem of IM is defined as follows: given network G and a number p, determine subset $N^* \subset N$ consisting of p spreaders (i.e., nodes) such that these nodes can spread their influence to other nodes $N \setminus N^*$ in G by maximizing the influence coverage and minimizing the time taken by these spreaders.

In terms of centrality based spreader assignment, we select several of the most popular and robust node centralities to serve as comparison for our proposed selection method, namely, node degree, betweenness, PageRank, and k-shell centrality [14,15]. As a baseline serving for comparison, we also use random spreader assignment.

The degree Deg of a node n_i is defined as the sum of all incident edge weights to that node's vicinity N_i as $Deg(n_i) = \sum_{n_j \in N_i} w_{ij}$, with $w_{ij} = 1$ in an unweighted context. Betweenness centrality Btw is defined as the fraction of shortest paths between all node pairs that pass through a specific node n_i [16].The PageRank algorithm, which is used at the core of Google's search engine [17], interprets an edge e_{ij} as a vote by node n_i to node n_j. Finally, with k-shell centrality, for every node a k-shell index gets assigned based on its topological location; to this end, nodes that are closer to the network core have higher k-shells. Nodes with greater k-shells are considered as more influential nodes. [15].

2.2. Network Datasets

In order to run and test GenOSOS we fist create models for the four fundamental complex network topologies [14]: a random Erdös-Rényi network *ER*, a regular mesh network *Me*, a Watts-Strogatz small-world network *SW*, and a Barabási-Albert scale-free network *SF*. Then, we generate four complex synthetic models: Holme-Kim *HK* [18], covert cellular networks *Cell* [19], Watts-Strogatz networks with degree distribution *WD* [20], and genetically optimized Genosian social networks *Gen* [21]. Finally, we include in our study the following four real-world datasets: a co-authorship network *CoA* [22], an online social network *OSN* [23], a scientific collaboration network *Geo* [24], and an email communication network *Em* [25].

We measure a standard set of network properties, for each dataset, which are given in Table 1. Here we include the network size (number of nodes N), number of edges E, average degree *avgD*, maximum degree *maxD*, average path length *APL*, average clustering coefficient *ACC*, network modularity *Mod*, and diameter *Dmt* [14].

Table 1. Network measures for the validation datasets divided into fundamental synthetic topologies, complex synthetic topologies, and real-world networks.

Dataset	*N*	*E*	*avgD*	*maxD*	*APL*	*ACC*	*Mod*	*Dmt*
ER	5000	25061	5.012	26	3.994	0.002	0.362	7
Me	5000	26948	5.390	44	11.515	0.148	0.821	30
SW	5000	19999	4.000	13	6.738	0.298	0.739	12
SF	5000	15762	3.152	294	5.378	0.007	0.64	13
HK	1000	3330	3.330	85	3.553	0.506	0.488	7
Cell	1041	6012	5.775	95	4.428	0.258	0.885	10
WD	1178	9048	7.681	58	15.419	0.659	0.93	32
Gen	1063	6915	6.505	25	4.765	0.498	0.882	9
CoA	1589	2742	3.451	34	5.823	0.878	0.955	17
OSN	1899	20296	10.688	339	3.055	0.138	0.338	8
Geo	3621	9461	5.226	102	5.316	0.679	0.743	14
Em	12625	20362	3.226	576	3.811	0.577	0.684	9

3. The GenOSOS Framework

3.1. Chromosome Representation

Each genetic generation consists of a solution population $S^j = \{s_1^j, s_2^j, ..., s_i^j, ..., s_n^j\}$ of n individual solutions s_i^j (chromosomes). The algorithm loop further consists of $1 \leq j \leq k$ iterative generations. As such, we represent a chromosome i from generation j as a solution network $s_i^j = \{N, N^*, E\}$. Here, N is the same set of nodes from the original network N, N^* is a subset of marked spreader nodes of size $|N^*| = p$, and E is the same set of edges which remain unchanged throughout the algorithm. Thus, each chromosome s_i^j differs through its custom selection of spreaders N^*.

Each initial chromosome is initialized by randomly marking $p << N$ nodes as spreaders. For a deterministic approach, we would have C_N^p possible combinations, which can be approximated by N^p solutions. While it is mathematically possible to obtain two initial equivalent chromosomes (with equivalent N^*), given the usual size of real-world networks $N > 1000$, and the limited number of selected spreaders $n < 100$, this probability is extremely small and is not a concern for our study.

3.2. Fitness Calculation

For every chromosome we need to be able to quantify its spreading efficiency. This efficiency is calculated by adopting a classic graph coloring algorithm starting from the marked nodes. Figure 1 exemplifies the fitness calculation on a network of $N = 20$ nodes. In step 1, we consider the $N^*(t = 1)$ spreaders from the chromosome representation as sources for coloring. Next, we repeat the graph coloring and mark all neighbors of the spreaders, obtaining a larger spreader set $N^*(t+1)$. We keep track of the growing set of marked nodes until $N^*(t) \geq 95\%N$ of the network is covered. Once this stop condition is met, the fitness of chromosome s_i^j is expressed as the number of nodes successfully colored divided by the number of steps required $f(s_i^j) = N^*(t)/t$.

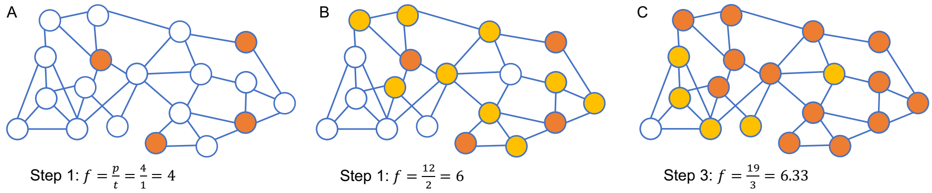
A
B
C
Step 1: $f = \frac{p}{t} = \frac{4}{1} = 4$
Step 1: $f = \frac{12}{2} = 6$
Step 3: $f = \frac{19}{3} = 6.33$

Figure 1. Example of computing fitness f on a small network with $N = 20$ nodes. (**A**) A number of $p = 4$ spreaders (dark orange) originate from the chromosome representation, leading to a fitness $f = 4$. (**B**) All adjacent nodes to the original spreaders are also marked as spreaders (yellow orange); at this stage $f = 6$ and only 60% of the network is colored. (C) The process continues until at least 95% of the network is colored; at this stage $f = 6.33$ and the graph coloring algorithm stops.

3.3. Defining Custom Genetic Operators

Elitism implies that a proportion r_e of the highest fitness (best) solutions from the n chromosomes are copied over to the next generation. This approach ensures that the fitness scores of the top $r_e n$ of the next generation will be at least as good as the current generation.

Crossover takes a pair of two randomly selected chromosomes from the pool of elite solutions, merges them together, and returns two new chromosomes. A crossover index in each of the chromosomes is randomly selected, and all the spreader nodes of the chromosomes after that selection index are exchanged between the two chromosomes. We symbolize r_c as the crossover rate.

Mutation on a chromosome is implemented by randomly selecting a spreader node n_i from the marked spreaders N^*, and swapping it with a random unmarked node from the remaining graph $N \setminus N^*$. The mutation operator is repeated given the mutation rate r_m.

3.4. Algorithm Implementation

The genetic algorithm of GenOSOS, shown in Figure 2, relies on three genetic operators – elitism, crossover, and mutation – and runs according to the following steps:

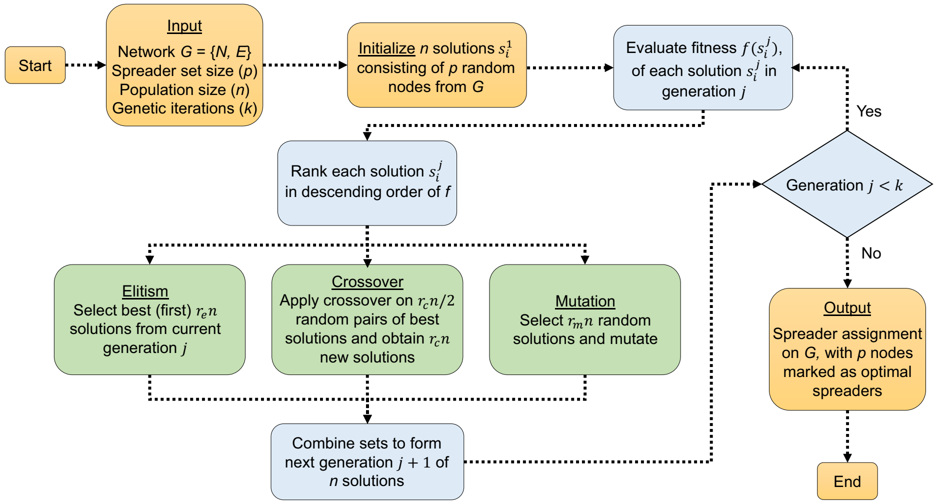
Start
Input
Network $G = \{N, E\}$
Spreader set size (p)
Population size (n)
Genetic iterations (k)
Initialize n solutions s_i^1 consisting of p random nodes from G
Evaluate fitness $f(s_i^j)$, of each solution s_i^j in generation j
Yes
Rank each solution s_i^j in descending order of f
Generation $j < k$
No
Elitism
Select best (first) $r_e n$ solutions from current generation j
Crossover
Apply crossover on $r_c n/2$ random pairs of best solutions and obtain $r_c n$ new solutions
Mutation
Select $r_m n$ random solutions and mutate
Output
Spreader assignment on G, with p nodes marked as optimal spreaders
Combine sets to form next generation $j + 1$ of n solutions
End

Figure 2. Flowchart of GenOSOS emphasizing the main algorithmic steps: input/output (orange), generation control (blue), and genetic operators (green). According to the flowchart, the algorithm finds an optimal solution s_i^j for placing p spreader nodes in a graph G, and runs k genetic iterations consisting of three operators that are used to generate n new solutions, from generation j, for the next generation $j+1$. The output consists of a set of p nodes marked as spreaders in graph G.

1. Initialize n solutions (chromosomes), each with p randomly marked spreaders.
2. Compute fitness $f(s_i^j)$ of each chromosome s_i^j in generation j.
3. Sort chromosomes in descending order of fitness f.
4. Copy the first (best) $r_e n$ of the chromosomes to the next generation $j+1$.
5. Pick $r_c/2 \cdot n$ randomly selected pairs of chromosomes from the best chromosomes and apply crossover, resulting in $r_c n$ new chromosomes.
6. Pick $r_m n$ randomly selected chromosomes and apply mutation on them.
7. Combine sets $r_e n$, $r_c n$, and $r_m n$ to form the next generation $j+1$ of size n.

8. Repeat steps (2-7) for $1 \leq j \leq k$ generations.

When solving NP-hard problems with heuristic methods (e.g., genetic algorithms), multiple combinations of model parameters can be feasible. As a trade-off between algorithmic speed and result precision, we simulate with a population size of $n = 1000$ chromosomes, a number of $k = 10$ generations, an elitism rate of $r_e = 0.5$, a crossover rate of $r_c = 0.3$, and a mutation rate of $r_m = 0.2$.

4. Results

4.1. Diffusion Coverage

We start by analyzing the diffusion coverage obtained by varying $p = 1 - 100$ spreaders on the random *ER*, mesh *Me*, small-world *SW*, and scale-free *SF* topologies, based on the SIR epidemic model. Spreaders are selected according to each of the six discussed selection methods: random *Rand*, degree *Deg*, betweenness *Btw*, PageRank *PR*, k-shell *KS*, and the proposed GenOSOS (*GOS*) method. Figure 3 displays the results for increasing p, and given in Table 2, are the values of diffusion coverage for $p = 10$ and $p = 50$. Each represented measurement is obtained after 10 repeated simulations.

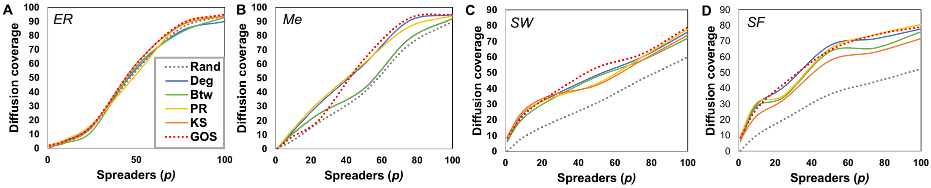

Figure 3. Diffusion coverage with increasing spreader set size $p = 1 - 100$ on the four fundamental complex topologies: (**A**) random *ER*, (**B**) mesh *Me*, (**C**) small-world *SW*, and (**D**) scale-free *SF*. The coverage obtained by GenOSOS is highlighted with the red dotted line (GOS).

Table 2. Diffusion coverage on the fundamental synthetic topologies, expressed in percentage (%), for the scenarios with $p = 10$ spreaders and $p = 50$ spreaders.

Topology	*p*	*Rand*	*Deg*	*Btw*	*PR*	*KS*	*GOS*
ER	10	4.03	5.22	3.78	6.12	4.56	5.16
Me	10	5.98	12.64	10.90	14.02	10.87	8.83
SW	10	9.05	24.50	21.42	25.21	25.94	23.15
SF	10	10.10	29.64	28.43	30.85	22.47	27.03
ER	50	55.03	56.32	57.88	52.46	60.02	59.54
Me	50	39.80	59.72	42.55	60.50	42.59	63.24
SW	50	30.72	48.45	47.43	42.98	42.04	53.22
SF	50	36.10	67.23	63.27	65.03	57.54	64.11

The simulation results over the synthetic topologies show that spreaders placed according to GenOSOS are capable of achieving similar, and superior diffusion performance compared to the state of the art centrality approach. Specifically, with GenOSOS we obtain the highest spreading coverage on the mesh and small-world (for $p = 50$). Our genetic algorithm approach outperforms state of the art graph centralities in 2 out 4 cases on the fundamental topologies. On average, *Rand* is 32.67% lower, *Deg* is 3.49% lower, *Btw* is 12.06% lower, *PR* is 7.97% lower, and *KS* is 15.79% lower in terms of diffusion coverage.

Next, we analyze the diffusion coverage on the Holme-Kim HK, cellular $Cell$, Watts-Strogatz with degree distribution WD, and Genosian Gen synthetic topologies. The same amount of spreaders is increased from $p = 1$ to $p = 100$ in the network, according to each of the six selection centralities. Table 3 presents the best results after 10 independent repetitions for each simulation scenario, obtained when $p = 10$, and $p = 50$. Figure 4 displays the increasing diffusion coverage for all values of p.

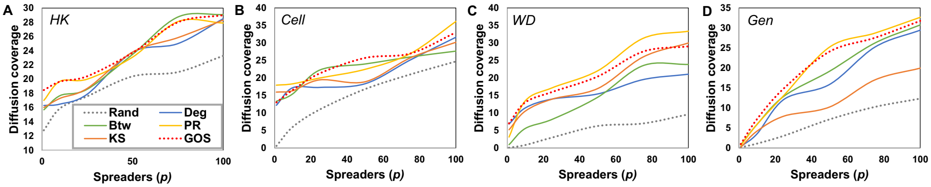

Figure 4. Diffusion coverage with increasing spreader set size $p = 1 - 100$ on the four complex synthetic networks: (**A**) Holme-Kim HK, (**B**) cellular $Cell$, (**C**) Watts-Strogatz with degree distribution WD, and (**D**) Genosian Gen. The coverage obtained by GenOSOS is highlighted with the red dotted line (GOS).

Table 3. Diffusion coverage on the complex synthetic topologies, expressed in percentage (%), for the scenarios with $p = 10$ spreaders and $p = 50$ spreaders.

Topology	***p***	***Rand***	***Deg***	***Btw***	***PR***	***KS***	***GOS***
HK	10	15.93	16.51	17.75	19.78	17.33	19.65
Cell	10	6.03	17.19	15.46	18.34	16.33	16.56
WD	10	0.84	11.46	5.62	13.49	10.55	13.04
Gen	10	1.12	3.57	5.92	5.92	4.25	7.24
HK	50	20.40	23.86	23.54	23.19	23.85	23.71
Cell	50	16.81	18.35	24.09	22.48	18.91	25.73
WD	50	6.28	15.44	14.51	22.07	16.97	20.43
Gen	50	7.15	15.89	19.19	25.21	10.24	24.10

The simulation results on the complex synthetic topologies show that spreaders placed according to GenOSOS achieve, again, a diffusion performance comparable to the centrality approach. Namely, GenOSOS scores the highest spreading coverage on the HK and Gen networks for $p = 10$, respectively HK and $Cell$ for $p = 50$, being roughly on par with the other centralities on the other networks.

Based on the presented measurements, our genetic approach outperforms the state of the art on 2 out of 4 networks. In terms of diffusion coverage, the spreaders selected according to GenOSOS achieve higher coverage rates, 46.08% more than *Rand*, 21.77% more than *Deg*, 13.55% more than *Btw*, 1.13% more than *PR*, and 25.52% more than *KS*.

Finally, we measure the diffusion coverage on the real-world co-authorship network CoA, online social network OSN, Geometry scientific collaboration Geo, and Email Em networks. The same amount of $p = 1 - 100$ spreaders are selected according to each of the six selection methods. In Figure 5 we display the measured diffusion coverage for all values of p. Each entry in Table 4 represents the best measurement obtained after 10 independent simulation repetitions, for $p = 10$ and $p = 50$ spreaders.

Overall, we notice that spreaders placed according to GenOSOS achieve high diffusion performance compared to the state of the art. Specifically, GenOSOS scores the

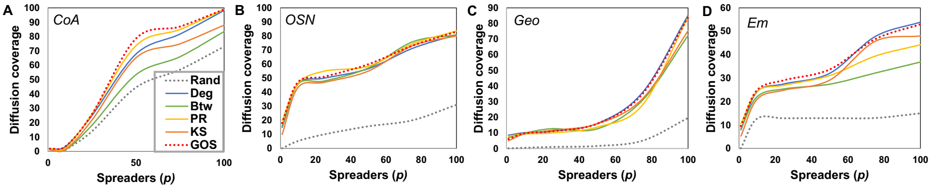

Figure 5. Diffusion coverage with increasing spreader set size $p = 1 - 100$ on the four real-world networks: (**A**) co-authorship network *CoA*, (**B**) online social network *OSN*, (**C**) Geometry scientific collaboration *Geo*, and (**D**) Emails *Em*. The coverage obtained by GenOSOS is highlighted with the red dotted line (GOS).

Table 4. Diffusion coverage on the real-world networks, expressed in percentage (%), for the scenarios with $p = 10$ spreaders and $p = 50$ spreaders.

Topology	*p*	*Rand*	*Deg*	*Btw*	*PR*	*KS*	*GOS*
CoA	10	1.06	3.52	1.25	2.39	1.46	3.69
OSN	10	5.22	45.86	45.81	46.18	43.33	47.10
Geo	10	0.46	10.08	9.91	9.36	9.38	9.75
Em	10	12.76	24.25	22.12	24.47	21.04	24.93
CoA	50	44.60	67.91	53.50	73.77	65.32	78.67
OSN	50	15.73	56.73	57.26	58.51	55.02	59.85
Geo	50	1.61	15.96	12.62	13.71	15.78	15.90
Em	50	12.95	32.06	27.43	31.11	29.15	33.72

highest spreading coverage on the co-authorship *CoA*, *OSN*, and email *Em* networks for $p = 10$. When $p = 50$, the results remain consistent, with GenOSOS scoring the highest coverage on the same networks. Based on the analyzed simulation results, we conclude that our genetic approach outperforms the state of the art on 3 out of 4 cases networks in terms of diffusion coverage. Specifically, the spreaders selected according to GenOSOS achieve higher coverage rates, namely 60.19% more than *Rand*, 8.22% more than *Deg*, 19.84% more than *Btw*, 5.86% more than *PR*, and 15.15% more than *Ks*.

5. Conclusion

In this paper we present a novel computational intelligence approach of selecting spreaders in complex networks based on genetic algorithms. We introduce the GenOSOS framework and compare it against state of the art methodology in selecting spreaders based on node centralities. SIR simulation results are quantified through diffusion coverage achieved on both synthetic and real-world datasets. The detailed analysis on three categories of network datasets show that the potential of our proposed solution is not only viable, but offers superior results compared to the state of the art centrality approach. Specifically, GenOSOS obtains a 11.45% higher coverage, averaged over all 12 datasets. In essence, our solution is superior to the state of the art on 7 out of 12 datasets (58.3%) in terms of diffusion coverage.

Overall, we have achieved to goal of this study, namely to: (i) investigate the alternative of optimal spreader selection using genetic algorithms, and (ii) also show that the genetic alternative can be, often, equal or superior in diffusion performance in comparison to the state of the art. Consequently, we have developed an important alternative spreader selection method without the need to estimate nodes centrality.

Acknowledgments

Author A.T. was supported by the Romanian National Authority for Scientific Research and Innovation (UEFISCDI), project number PN-III-P1-1.1-PD-2019-0379.

References

[1] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks. In: Proc. of 15th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining. ACM; 2009. p. 199–208.

[2] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proc. ACM SIGKDD Int. Conf. on Knowledge discovery and data mining. ACM; 2003. p. 137–146.

[3] Topîrceanu A. Competition-Based Benchmarking of Influence Ranking Methods in Social Networks. Complexity. 2018;2018.

[4] Easley D, Kleinberg J. Networks, crowds, and markets: Reasoning about a highly connected world. Cambridge University Press; 2010.

[5] Golbeck J. Analyzing the Social Web. Access Online via Elsevier; 2013.

[6] Zareie A, Sheikhahmadi A, Khamforoosh K. Influence maximization in social networks based on TOPSIS. Expert Systems with Applications. 2018;108:96–107.

[7] Zareie A, Sheikhahmadi A, Jalili M. Identification of influential users in social network using gray wolf optimization algorithm. Expert Systems with Applications. 2020;142:112971.

[8] Gong M, Yan J, Shen B, Ma L, Cai Q. Influence maximization in social networks based on discrete particle swarm optimization. Information Sciences. 2016;367:600–614.

[9] Tang J, Zhang R, Wang Pea. A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks. Knowledge-Based Systems. 2020;187:104833.

[10] Bucur D, Iacca G. Influence maximization in social networks with genetic algorithms. In: European conference on the applications of evolutionary computation. Springer; 2016. p. 379–392.

[11] Gong M, Song C, Duan C, Ma L, Shen B. An efficient memetic algorithm for influence maximization in social networks. IEEE Computational Intelligence. 2016;11(3):22–33.

[12] Cui L, Hu H, Yu Sea. DDSE: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. J of Net and Comp App. 2018;103:119–130.

[13] Pastor-Satorras R, Castellano C, Van Mieghem P, Vespignani A. Epidemic processes in complex networks. Reviews of modern physics. 2015;87(3):925.

[14] Wang XF, Chen G. Complex networks: small-world, scale-free and beyond. IEEE circuits and systems. 2003;3:6–20.

[15] Kitsak M, Gallos LK, Havlin S, Liljeros F, Muchnik L, Stanley HE, et al. Identification of influential spreaders in complex networks. Nature physics. 2010;6(11):888.

[16] Topirceanu A, Udrescu M, Marculescu R. Weighted betweenness preferential attachment: A new mechanism explaining social network formation and evolution. Scientific Reports. 2018;8(1):1–14.

[17] Brin S, Page L. Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks. 2012;56(18):3825–3833.

[18] Holme P, Kim BJ. Growing scale-free networks with tunable clustering. Phys Rev E. 2002;65(2).

[19] Tsvetovat M, Carley KM. Generation of realistic social network datasets for testing of analysis and simulation tools. DTIC Document; 2005.

[20] Chen Y, Zhang L, Huang J. The Watts–Strogatz network model developed by including degree distribution: theory and computer simulation. J of Physics A: Math and Theo. 2007;40(29):8237.

[21] Topirceanu A, Udrescu M, Vladutiu M. Genetically Optimized Realistic Social Network Topology Inspired by Facebook. In: Onl Soc Media Anal and Vis. Springer; 2014. p. 163–179.

[22] Newman MEJ. Finding community structure in networks using the eigenvectors of matrices. Phys rev E. 2006;74(3):036104.

[23] Opsahl T, Panzarasa P. Clustering in weighted networks. Social networks. 2009;31(2):155–163.

[24] Batagelj V, Mrvar A. Pajek-program for large network analysis. Connections. 1998;21(2):47–57.

[25] Leskovec J, Lang KJ, Dasgupta A, Mahoney MW. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Math. 2009;6(1):29–123.