International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-11, September 2019

Influence Maximization in Social Networks using Deterministic Crowding Algorithm

Navid Kaveh, Mehdi Bateni

Abstract*: In a social network the individuals connected to one another become influenced by one another, while some are more influential than others and able to direct groups of individuals towards a move, an idea and an entity. These individuals are named influential users. Attempt is made by the social network researchers to identify such individuals because by changing their behaviors and ideologies due to communications and the high influence on one another would change many others' behaviors and ideologies in a given community. In information diffusion models, at all stages, individuals are influenced by their neighboring people. These influences and impressions thereof are constructive in an information diffusion process. In the Influence Maximization problem, the goal is to finding a subset of individuals in a social network such that by activating them, the spread of influence is maximized. In this work a new algorithm is presented to identify most influential users under the linear threshold diffusion model. It uses explicit multimodal evolutionary algorithms. Four different datasets are used to evaluate the proposed method. The results show that the precision of our method in average is improved 4.8% compare to best known previous works.

***Keywords*: Social Network, Influence Spread, Influence Maximization, Explicit Multimodal Evolutionary Algorithms, Deterministic Crowding.**

I. INTRODUCTION

Social network analysis (SNA) book published in 1994 by Stanley Wasserman has led to a serious assessment of social network science as a subset of social and mathematical matter. According to Wasserman, every social structure made through individuals' social relation is named social network. Every social network is based on the two elements of contributive entities participating in connections and the social relations between those entities. Social network is concerned with the relation between individuals, organizations and other social elements [1].

SNA is strategy and approach for analyzing social network structure. This process includes searching and assessing of social network structure as a graph formed through tools and individuals connected with each other by connection lines [2]. In SNA a social structure is considered as a graph, individuals and social relation are termed as node and edge. In SNA analysis, networks are modeled as simple, weighted and directional graphs, $G = (V, E)$ where, nodes V determines the entities and edges E as to their being connection or independent. Nodes can represent any type of entities like individuals and organizations or objects throughout computer networks. The edge among nodes can describe any kind of reactions or dependency like friendship, financial and commercial transactions and network connections.

In social network the majority of individuals, share their own opinion and ideas about news and products with friends and relatives. Thus, individuals in a network are subject to influence of one another and one's opinion can affect others' opinion. Changes throughout one's thoughts, emotions, attitude and behavior are named the social influence. Social influence change one's correlations with others as to persuading and opposing are considered as mentioned changes [3]. The level of social influence depends on different factors, like manner of individuals' connection in network and network's particulars. Social influence leads to information diffusion and influence propagation into a social network.

Information diffusion is a process during which a piece of information is developed virtually or physically from one individual or a group to another [4]. There exist different types of influences spread in a social network consisting of contagious disease outbreak, rumors and belief spreading, advertising, news broadcasting and vote propagation or a virus spreading in a computer network. Due to the individuals influence on each other, a new type of marketing known as viral marketing is formed. This type of marketing adopts the word-of-mouth (WOM) approach to spread product's information in a network [5]. A company, by adopting this strategy for advertising it's product, distributes the product for free among k number of costumers as sample costumers to test the product and, in case of satisfaction, advertise the product among their friends without any expenditure. Some friends of these customers buy the product and suggest the same to others, hence, the product is generalized and advertised among society virally and based on costumers' opinion. In this approach the whole expenses are taken by the company consist of the goods provided to their initial customers. The main issue is the selection of k number of customers who have the maximum influence on others and increase the number of buyers.

In influence maximization (IM) problem, the goal is to find a small subset containing k number of individuals, which can influence the largest number of network members by consuming the minimum amount of time and cost via a diffusion model. This k membered subset is known as the initial nodes or seed set nodes. Through the influence of these members other nodes become activated and lead to information diffusion [6].

Manuscript published on 30 September 2019.
***Correspondence Author(s)**
Navid Kaveh*, School of Engineering, Sheikhbahaee University, Isfahan, Iran. Email: NavidKaveh@shbu.ac.ir
Mehdi Bateni, School of Engineering, Sheikhbahaee University, Isfahan, Iran. Email: Bateni@shbu.ac.ir

The matters related to information diffusion can be classified into two categories: 1) the explanatory models and 2) predictive models. Explanatory models are also classified into two categories of influence models and epidemic models. The three most popular epidemic models are susceptible infected (SI), susceptible infected susceptible (SIS) and susceptible infected removed (SIR). Independent cascading (IC) model and linear threshold (LT) models are the most popular predictive models mostly adopted in studies and SNA subjects like influence spread.

In IC model, for each active node, a probability is considered for the neighbor nodes' activation. The weight of an edge indicates this probability. In this model, each activated node has only one chance to activate its neighbor node. The influence diffusion begins by activation of the initial set at step *t*. Every newly activated node, like *v* node can activate each currently inactive neighbor *w*. If *v* succeeds, then *w* node will become active in step *t+1* and has one chance to activate its own inactive neighbors in next step. This process continues until the whole network is activated or no more activations are possible [7].

In LT model, each *v* node in time unit is in one of the following states of zero (inactive) and one (active). A *v* node is influenced by each neighbor *w* according to a weight $b_{v,w}$ such that $\sum_{(w \text{ neighbour of } v)} b_{v,w} \leq 1$. In this model, each *v* node chooses a threshold θ_v uniformly at random from [0,1]. Lack of existing information and knowledge on node activation tendency is the reason of applying random selection. The θ_v indicates the minimum required influence that a *v* node receives from its neighbors to be activated. First, the initial set named *A* is activated. For activation of *v* node, the total weight of activated neighbors should exceed θ_v. In any step all previously activated nodes remain active and are added to the initial set. The diffusion process continues until all nodes are activated or no new node is activated [8].

The existing algorithms of influence maximization in social networks have advantages and disadvantages. A set of algorithms have appropriate influence spread, that is, the nodes selected in these solutions have influence a high percentage of network members. Due to high computational time and not being scalable they are not optimized to be applied in big SNs. As to computational time, another set of algorithms reduce the solution-solving time, while not accuracy in finding the influential nodes, thus, the necessity of having a solution to improve the influence spread.

In the multimodal problems, the populations run the recombination locally and instead of one, several optimized answers are obtained. Thus, the possibility of better and optimized answer obtaining is increases. By converting influence maximization problem to multimodal problem, the diversity of the answers is guaranteed. Each of these obtained answers, have their own advantages and possesses the optimized influence spread. These answers establish coverage or development more than expected spread and influence higher percentage of network members.

This article is organized as follows: the related work is presented in Sec. 2; the multimodal solutions are introduced in Sec. 3; the new algorithm is proposed in Sec.4; the results is evaluated in Sec. 5 and the article is concluded in Sec.6.

II. RELATED WORK

The influence maximization problem was proposed in 2001, by Pedro Domingos and Mathew Richarson as to find a set of beneficial costumers in viral marketing [5]. David Kempe et al., [7] modeled the influence maximization problem as a discrete optimization problem and proved that IM is categorized as a NP-Hard problem in all diffusion models. Thus, the researchers, during years, have been and are seeking manners to develop algorithms that can find nodes of seed set and have appropriate influence spread. These algorithms are categorized as follows [9]:

- Approximant algorithms: In this context, approximant algorithms produce high approximant rate and acceptable influence spread, while most of them are not scalable and by growing the size of the network, running time increases severely. Basic Greedy, CELF, CELF++ are among the approximant algorithms.
- Heuristic solutions: Most of the presented heuristic solutions are more scalable and optimized in running time compared the approximant algorithms. SIMPATH, LDAG, MIA are among the heuristic algorithms.
- Community-Based algorithms: These algorithms adopt community identification in social networks as a solution to improve algorithm's scalability. CGA and ComPath algorithms are among these algorithms.
- Metaheuristic solutions: Most of these algorithms are developed based on Evolutionary Computation approaches including the SA and GA.
- Miscellaneous solutions: These types of algorithms follow no specific modality, indicating the Data based and Query based approaches are this type of algorithms.

A. Approximant Algorithms

Kempe et al., [7] formulate the influence maximization problem as a combinatorial optimization problem for the first time and investigated its computational issues based on LT and IC diffusion models. They suggested a greedy algorithm with provable approximant for selecting seed set nodes, where the KGA becomes an active node and then the diffusion model is adopted for as many times as required. After nodes are tested, the node which has activated most nodes is selected, then the remaining nodes are added to the initial node group and then the simulation begins again to select the best node. The process continues until *k* number of nodes is selected. The approximate results obtained through KGA are of minimum accuracy at (1 - 1/e) that can be considered as an optimization criterion. The KGA is a hill-climbing greedy algorithm and the network's nodes are measured repeatedly in every iteration. To determine each node influence, *R* times simulation is required, which reduces the algorithm's efficiency to (generally *R*=20000), thus a reduction in algorithm efficiency [10]. To improve the scalability problem, Leskovec et al., [11] suggest a slow but computationally efficient method named as CELF which adopts the submodularity properties and local searching to solve the influence maximization problem.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The main idea of this algorithm is in maintaining the influential nodes' graph in a priority queue which can reduce the greedy algorithm evaluation and improve the running time of the algorithm.

B. Heuristic Solutions

These algorithms present no approximate limit to solve the influence maximization while are of better time consumption and scalability. Some of them are independent from diffusion model and heuristics like Random Heuristic and Centrality-Based Heuristics (CBH) are adopted through these algorithms. In Random Heuristic algorithms, *k* number of network's nodes is selected randomly as seed set nodes. This method is applied as a heuristic approach Kempe et al., [7]. The centrality is a known measure in network analysis that indicating the importance of a node in the network. Here, some of the CBH heuristics in influence maximization problem are mentioned: 1) Maximum Degree Heuristic (MDH) where, *k* number of nodes with highest degree as the seed set nodes are selected, 2) High Clustering Coefficient Heuristic (HCH) where, *k* number of nodes with highest clustering coefficient value are selected and 3) High Page Rank Heuristic where, *k* number of nodes with highest page rank value are selected.

Some other heuristic methods are specifically designed for a diffusion model with appropriate efficiency, including 1) Maximum Influence Arborescence (MIA) for IC model [12], 2) Local Directed Acyclic Graph (LDAG) method [13] and, 3) SIMPATH method for LT model [14]. In MIA algorithm for simplification of influence spread the tree structure is emphasized. The SIMPATH heuristic algorithm functions based on CELF principals. This algorithm uses path enumeration techniques for determination influence spread instead of Mont-Carlo simulation. This algorithm contains a controlling parameter by the user that maintains the balance between accuracy and algorithm running time. The LDAG algorithm is developed based on LT model. The influence spread in Directed Acyclic Graph (DAG) in easy calculation, while usually the real social networks are not of DAG type, thus, the LDAG model is introduced to solve the influence maximization problem.

C. Community-Based Solutions

In social networks, community can be considered as the minimized samples of the real world. Community is a subset interrelated nodes considered as a united group. The CGA is a community based greedy algorithm presented by Chen et al., [15], when by pruning the graphs improves the KGA. The CGA by identifying the communities seeks to obtain a new graph and find the seed set nodes therein. By considering the fact when calculating the active node influence in a community, the influence is limited only to the same community, the running time of algorithms improves and followed by a reduction in the search space which makes the algorithm scalability possible.

D. Metaheuristic Solutions

Jiang et al., [16] introduced a Simulated Annealing (SA) based algorithm subject to the IC model, which approximates the seed set nodes influence spread through local search in a faster manner compared to greedy algorithm because there is no need to add nodes to the seed set at every iteration. The Zhang et al., [10] proposed an algorithm based on Genetic algorithm for this purpose under the LT model and by applying a multi-population competition and population evolution obtained optimized solutions on influence. This GA is much faster compared to greedy algorithm, but because it is of multipoint local search type, after combination the populations, it gradually approaches premature convergence, leading to a decrease in solution diversity.

E. Miscellaneous Solutions

The methods in this context differ from the discussed algorithms. Every presented solution is different from another. The algorithm presented by Ma et al., [17] operates based on the heat diffusion process. Goyal et al., [18] developed a data-based approach. They introduced Credit Distribution model (CD) which when tracking diffusion and learning the influence process pattern presents appropriate approximant approach. Lee et al., [19] presented a query-based method subject to IC model to solve the effective nodes finding problem that is considered to maximize the influence of certain group of users in viral marketing. The specifications of the discussed algorithms are observed in Table- I, in brief.

Table I: Literature Content Review

Year	Name of the Algorithm	Solution Approaches	Diffusion Model
2016	GA Algorithm	Metaheuristic solutions	LT Model
2011	SA Algorithm	Metaheuristic solutions	IC Model
2011	SIMPATH Algorithm	Heuristic solutions	LT Model
2010	CGA Greedy Algorithm	Community-Based solutions	IC Model
2010	LDAG Algorithm	Heuristic solutions	LT Model
2010	MIA Algorithm	Heuristic solutions	IC Model
2007	CELF Greedy Algorithm	Approximant Algorithms	Triggering Model
2003	KGA Greedy Algorithm	Approximant Algorithms	LT Model and IC Model

III. INTRODUCING MULTIMODAL SOLUTIONS

The evolution in real-world is influenced by factors like physical space. It leads to a condition of locality in evolution, that is, the individuals of one species that evolve in different physical spaces, leading to have different forms and this phenomenon leads to diversity in species. The multimodal solutions are based on this phenomenon. Most of the real-world problems have search space with several local optima where usual solution can find only one of them, which have various optimum points, if all points contribute in problem solving, and then the problem is multimodal. Each one of these points is a local optimum and their best would be the global optimum. Each one of these optimums is named Niche, with limited physical source that should be shared among the Niche population individuals. Niching allows the evolution algorithms like the GA to investigate many peaks in a parallel manner.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In multimodal problems, for each objective a set of solution would be provided, where, each solution contains a good aspect and each can be the final optimal. It is possible that the fitness function may not represent the exact concept and lack enough accuracy. Thus, if such optimal is obtained, all of this optimal will be introduced and the possibility of judgment for optimal answer selection will be provided. It means that there is a possibility to choose a solution from the set of solutions as the optimum. The mechanism of solving multimodal problem consist of the two: Implicit and explicit approaches: in the first, some changes are made in evolution algorithm to distribute the population around several optimal points instead of converging all around one optimal point, thus, the required diversity is provided. Of course, these approaches do not guarantee the diversity of algorithm solutions; in the second, the existing operators in the evolutionary algorithm is changed in a sense that it would be able to maintain the diversity of population explicitly. Usually the maintenance of population's diversity in explicit approach is done in two ways: the fitness sharing and, the crowding.

IV. SUGGESTED ALGORITHM

This solution is categorized as the metaheuristic algorithms, where the GA is applied while, by making changes in GA and converting it into a multimodal one, it seeks to provide diversity and prevent the premature convergence. The method adopted in this study is Deterministic Crowding (DC). The important symbols applied in describing this proposed algorithm are tabulated in Table II.

Table II: The Important Applied Symbols

Symbols	Explanations
$G= (V, E)$	A network including V nodes and E edges set
V	Nodes set
E	Edges set
A	The seed set node $A= \{v_1, v_2, \ldots, v_k\}$
k	The size of seed set nodes named A
$\sigma (v)$	The nodes set where node v can influence
R	The number of simulation rounds in greedy algorithms

The information diffusion model applied through this study is LT. In network G, V nodes can be active or inactive. The influence process from one active node or a set of active nodes is named influence spread. As influence spreads under a diffusion model, the network nodes can be converted from inactive to active. The IM problem seeks to find a set with k number of nodes named A where, $\sigma(A)$ is maximized based on the diffusion model. In KGA, to prove that the influence spread of LT model is submodular, node threshold is selected randomly and LT model are considered as a random model. In this method, to evaluate and calculate the average influence of each node, in every simulation, the threshold of each node is selected randomly, that is, in KGA, the $\sigma(A)$ variable is an average that maximized in R simulation round. Selecting different thresholds randomly in each round of simulation can have different influence spread. The greedy algorithm may hold only one solution and discuss diversity of solutions rarely. The suggested heuristic solutions for IM problem have different mechanism from greedy algorithm. For example, the SA is designed for IC diffusion model and finding more than one optimal solution for fixed threshold in LT model in this algorithm is difficult [10].

In this study, for solving IM problem based on the LT model with fixed thresholds, metaheuristic genetic algorithm with explicit multimodal mechanism is suggested. Applying a multi-population stochastic algorithm, populations competition and evolution can lead to diversity in solutions. The GA contains advantages in problem solving and search for diversity of solutions. In the GA a global optimum is introduced as the best solution. It is possible that some of the stronger nodes lead to premature convergence in these points. After a few iterations, this algorithm, due to losing the population diversity becomes trapped in local optima. The explicit multimode approaches have expanded as to maintain population diversity in standard GA. In this study, by applying this proposed algorithm in addition to finding the best optimum as the global optimum and it is sought to find few qualified solutions as the local optimum. Next to maintaining GA solution diversity premature convergence is prevented.

A. The Main Algorithm Framework

This proposed framework for the purpose here is provided in algorithm 1:

Input: Graph $G = (V, E)$, Threshold θ, Seed set scale k, Population number m, Mutation Probability p_{m1}, Mutation Probability p_{m2};
Output: Set of seed set A;
1: Initialization population $p_i = \{v_1, \ldots, v_k\}$ $(i = 1, \ldots, m)$;
2: Calculate influence of each p_i: $\sigma (p_i)$;
3: **while** Termination Condition **do**
4: Select two parents p_i, p_j randomly with no replacement.
5: Crossover operator: $p_i \leftrightarrow p_j$;
6: **for** each offspring **do**
7: **if** random $\xi_m < p_{m1}$ **then**
8: Mutation operator: $o'_i \leftarrow o_i$;
9: **end if**
10: **if** random $\xi_n < p_{m2}$ **then**
11: Mutation operator: $o'_i \leftarrow o_i$;
12: **end if**
13: **end for**
14: Calculate influence of each offspring $\sigma (o'_i)$;
15: Crowding: Offspring o'_i, o'_j replace the nearest parent; if they have greater fitness;
16: Calculate influence of each p_i: $\sigma (p_i)$ and select $max|\sigma(p_i)|$, $A = argmax (|\sigma(p_i)|)$;
17: **end while**
18: **return** A;

Algorithm 1: The Pseudo code of Main Algorithm

In GA, first, a few seed sets are formed each with k nodes, named the population. At the initialization phase the required population for GA is constructed.

The population $p_i = \{v_1, v_2, \ldots, v_k\}$ $(i = 1, 2, \ldots, m)$ is the m count of seed set, with size k (line 1). Through the genetic operators, the chromosomes compete and combine to generate new chromosome. To make the contribution of randomness in GA outstanding, selection of nodes at initialization phase is considered random. The main GA operators (lines 5-13) consist of crossover, (line 5) and two mutation operators (lines 6-13). To generate more diversity in the population, here, the crossover rate is 1, indicating that the recombination process would certainly run. To get new offspring and make the difference with parents two mutations occur. In deterministic crowding (line 15), the offspring next to competing with their parents, if highly fitness, replace their closest parent. The count of nodes activated by each seed set nodes is used as the fitness function. The termination condition of the algorithm is the active nodes by a seed set node remains monotonous with no advance.

B. The Crossover Operator

The most essential operator in GA is the crossover. Crossover is a change process among the population members' nodes with the objective to generate new members and cause diversity in the population. This operator runs based on the probability and crossover rate, which here is 1. To occur the crossover, this proposed method runs as follows: Two p_i and p_j members are selected as the parents from the population in a random manner. To generate offspring, first, the common nodes in both the parents are determined (lines 2-4), next, they are eliminated from both the parents. Here, the parents' length is reduced per the common nodes count (lines 5-7). Now the crossover point of the parents are determined, where the crossover point is located in the middle of both the parents (line 10). By selecting the condition C_p, a single point crossover is run and the nodes in both the parents are replaced. At this point both the parents break up and their tails become replaced (lines 11-14). The omitted common nodes set are added to the yield crossover and the new offspring are generated (lines 15-16).

Input: Crossover two parents p_i and p_j;
Output: New offspring o_i and o_j;
1: Empty population t, is for saving similar vertexes (v) in two parents;
2: **for** $n = 1: k$ **do**
3: $v_n \in p_i$;
4: **if** $v_n \in p_j$ **then**
5: $p'_i = p_i - v_n$;
6: $p'_j = p_j - v_n$;
7: $t = t + v_n$;
8: **end if**
9: **end for**
10: Calculate *flag* C_p, $C_p = \lceil |p'_i| / 2 \rceil$
11: **for** $n = 1$ *to* C_p **do**
12: $v_n \in p'_i$, $u_n \in p'_j$;
13: $v_n \leftrightarrow u_n$;
14: **end for**
15: $o_i = p'_i + t$;
16: $o_j = p'_j + t$;
17: **return** o_i and o_j;

Algorithm 2: The Crossover Operator Pseudo code

C. The Mutation Operator

Mutation is a random change of genes in a parent to generate a child. Mutation is known as an exploration in space search. This operator is run on the offspring with p_m probability. Here, two mutations are applied, in the first, a random number is generated between [0,1], (line 2) and if it is smaller than the first mutation rate, then the mutation takes place (line 4). In this mutation, first, a node is selected in a chromosome in a random manner (line 3) and another node is selected in a network in the same manner (line 5). If this node is not repetitive with the node selected in the chromosome it replaces the latter (line 6 and 7). In the second mutation, a new random number is generated between [0,1] and if it is smaller than the second mutation rate, then the mutation takes place (line 10). In this mutation a number is selected in a random manner (line 3) which determines the number of the node in the best solution (line 11). If this node is not repetitive with the corresponding node in the chromosome, it replaces the latter (line 13).

Input: Crossover two offspring o_i and o_j;
Output: New offspring o'_i and o'_j;
1: **for** each offspring **do**
2: Create random ξ_m and ξ_n;
3: Create random g for choose vertex in offspring and best solution $| 1 \le g \le k$
4: **if** random $\xi_m < p_{m1}$ **then**
5: Create random h for chooce vertex in network $| 1 \le h \le N$;
6: **if** $v_h \notin o_i$ **then**
7: Replace $v_g \leftarrow v_h$ for create o'_i;
8: **end if**
9: **end if**
10: **if** random $\xi_n < p_{m2}$ **then**
11: Go to best solution and select v_g, Called $v_g.best$;
12: **if** $v_g.best \notin o_i$ **then**
13: Replace $v_g \leftarrow v_g.best$ for create o'_i;
14: **end if**
15: **end if**
16: **end for**
17: **return** o'_i and o'_j;

Algorithm 3: The Mutation Operator Pseudo code

D. Deterministic Crowding

Crowding is a technique applied in preserving diversity in population and prevent premature convergence in the local optimum. This technique concentrates on replacement process and at survivor selection stage it changes. In crowding algorithm new individuals replaced similar members of the population. Determining the similarity of all individuals is a difficult task with high calculation load. In this case an improved algorithm named deterministic crowding (DC) is applied [20], where the selection is removed and to reproduce the population it is divided into pairs in a random manner. Here, to begin with, the two p_i and p_j parents are selected in a random manner, (line 1) and by applying the crossover operator the two o_i and o_j offspring are produced (line 2).

At this stage, each offspring becomes subject to mutation and a new offspring is produced (line 3). Now the new offspring is evaluated (line 4). Whether to survive or be eliminated each offspring should compete in a tournament with one of its parents, (line 5). To determine this operation four pairwise distance between parents and offspring are calculated to find which offspring similar its parents the most, Eq. (1):

$$d(pi, oi) + d(pj, oj) \leq d(pi, oj) + d(pj, oi) \quad (1)$$

where, if true, that is the right term is bigger, then p_i parent competes with o_i offspring and p_j competes with o_j and if smaller, p_i competes with o_j and p_j competes with o_i. At the end, as to fitness competition the winner survives for the next generation and the loser is eliminated (lines 6-10).

1: Select two parents p_i, p_j randomly with no replacement.
2: Perform a crossover between them yielding offspring o_i, o_j.
3: Apply mutation operator to generate o'_i, o'_j.
4: Calculate influence of each offspring $f(o'_i) = \sigma(o'_i)$. (Step 1 to 4, in Main Algorithm done)
5: **if** $[d(p_i, o'_i) + d(p_j, o'_j) \leq d(p_i, o'_j) + d(p_j, o'_i)]$ **then**
6: **if** $f(o'_i) \geq f(p_i)$ **then** replace p_i with o'_i;
7: **if** $f(o'_j) \geq f(p_j)$ **then** replace p_j with o'_j;
8: **else**
9: **if** $f(o'_j) \geq f(p_i)$ **then** replace p_i with o'_j;
10: **if** $f(o'_i) \geq f(p_j)$ **then** replace p_j with o'_i;
11: **end if**

Algorithm 4: The Deterministic Crowding Pseudo code

E. Fitness Function

To evaluate each population influence spread, the fitness function is applied. The count of influenced nodes, by each seed set node is considered as its fitness. In its contextual sense, the fitness value each node equals the node count which by activation of the node of concern, become activated. Consequently, for each genetic chromosome or seed set the fitness value is equals the total number of activated nodes by the same seed set.

Input: Graph $G = (V, E)$, set of initially influenced nodes $\sigma(t_0)$;
Output: Size of final set of influenced nodes $\sigma(I)$;
1: $i = 0$;
2: Uniformly assign random thresholds θ_v from the interval [0,1];
3: **while** $i = 0$ or $\sigma(t_{i-1}) \# \sigma(t_i)$ **do**
4: $\sigma(t_{i+1}) = \sigma(t_i)$;
5: $uninfluenced = V \setminus \sigma(t_i)$;
6: **for all** $v \in uninfluenced$ **do**
7: if $\sum_{(w \text{ influenced neighbour of } v)} b_{v,w} \geq \theta_v$ then
8: influence v;
9: $\sigma(t_{i+1}) = \sigma(t_{i+1}) \cup \{v\}$;
10: **end if**
11: **end for**
12: $i = i + 1$;
13: **end while**
14: $\sigma(I) = \sigma(i)$;
15: **return** $|\sigma(I)|$;

Algorithm 5: Fitness Function Pseudo code

V. EVALUATION AND RESULTS

By applying the deterministic crowding algorithm, the most influential nodes are found in the datasets. By comparing the obtained results through this proposed algorithm, the extent of its capabilities against its counterparts is revealed in this context.

A. The Dataset

This algorithm is run in four networks for evaluation, Table III. In all tests the threshold of the nodes subject based on LT model is adjusted within [0,1] range. The NETSCIENCE network is a network of researchers who work on network theory. EMAIL is a dynamic internal email network with the edges weight equaling its emails among a company's staff in a certain period. INFECTIOUS is a network of science gallery visitors where, edges reveal the face to face communications among the visitors. This network is dynamic and its edges' weight indicates the audience count. UCSOCIAL is an online messenger network specific to students of the University of California where its edge weights indicate the number of messages [10].

Table III: Specifications of the Experimental Networks

Network	Nodes	Edges	Average Degree	Density
NetScience	1589	2742	3.451	0.002
Email	167	5784	34.635	0.209
Infectious	410	5530	13.488	0.033
Ucsocial	1899	20296	10.688	0.006

B. Experiments

To solve the IM problem each dataset is tested through deterministic crowding approach and the results are compared with KGA and GA. The lab environ consist of a PC with Intel® Pentium® G4400, 3.30 GHz processor and a 4 GB RAM. All algorithms are tested by applying different seed set size and then compared. In all tests the influence spread is subject to fixed threshold θ_v, within [0,1] range, indicating that KGA does not require R rounds of simulations.

C. Influence Spread Test Results

In NETSCIENCE network, Fig. 1, the outperformance of this algorithm in relation to GA but not KGA is evident. In NETSCIENCE, the network structure is Sparse, thus algorithm not efficient in searching neighbor set. In EMAIL network, Fig. 2, when the seed set size > 10 a significant increase in the influenced node count become evident and for seed set size > 30 this algorithm performs well. In INFECTIOUS network, Fig. 3, where the results almost resemble that of the KGA and GA, at seed set size < 30 this algorithm performs well. In UCSOCIAL network, Fig. 4, where seed set size > 15 this algorithm outperforms its counterparts.

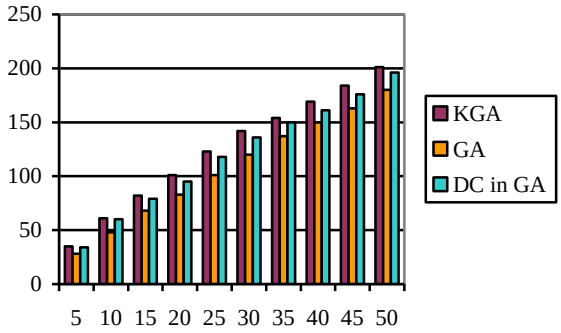

Fig. 1.Test Results of Influence Number on NETSCIENCE

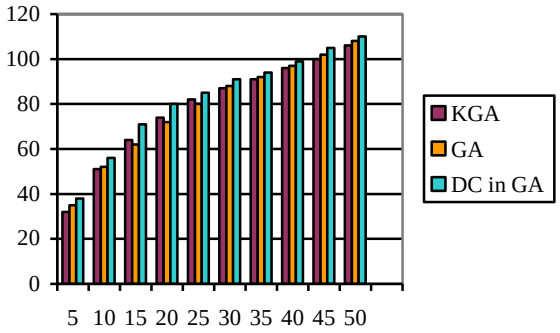

Fig. 2.Test Results of Influence Number on EMAIL

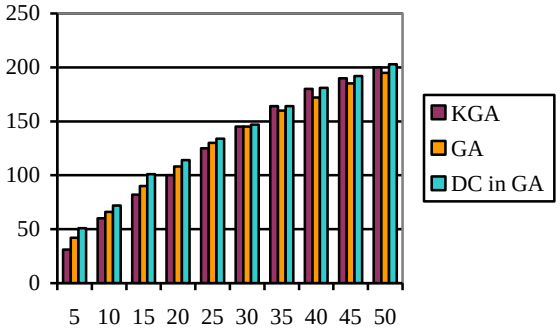

Fig. 3.Test Results of Influence Number on INFECTIOUS

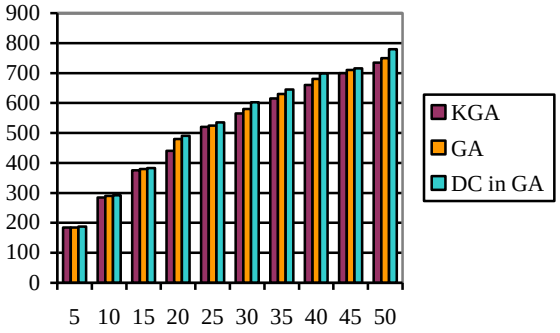

Fig. 4.Test Results of Influence Number on UCSOCIAL

D. The Runtime Test Results

As to efficiency, we compare running time of the three algorithms when seed set size is 10, 30 and 50 on four networks. Here, because all thresholds are similar the simulation round in KGA is 1 (*R=1*). As observed in Figs. 5-8, the runtime of proposed algorithm is slightly more than that of GA, while, much faster than that of KGA.

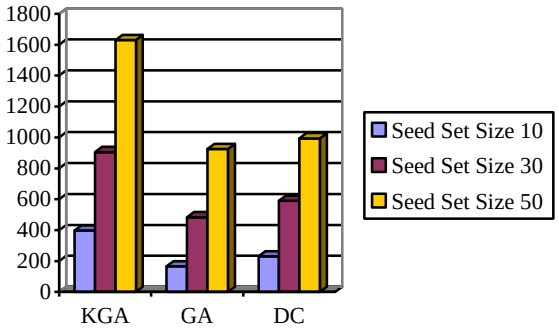

Fig. 5.The Results of Running Time on NETSCIENCE

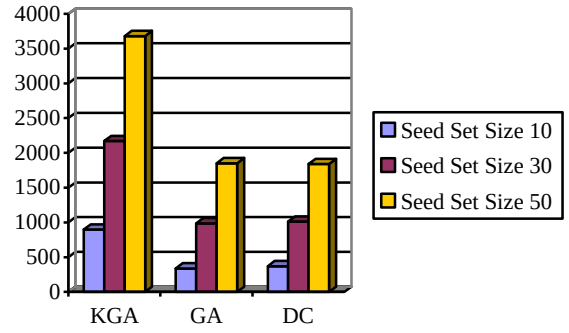

Fig. 6.The Results of Running Time on EMAIL

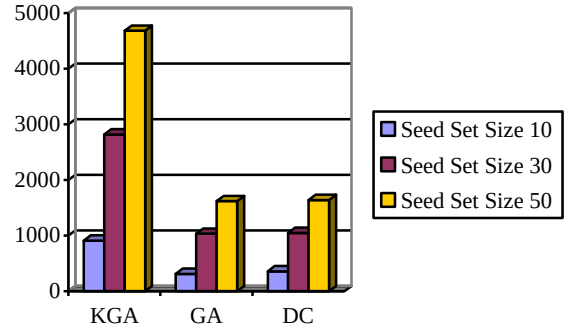

Fig. 7.The Results of Running Time on INFECTIOUS

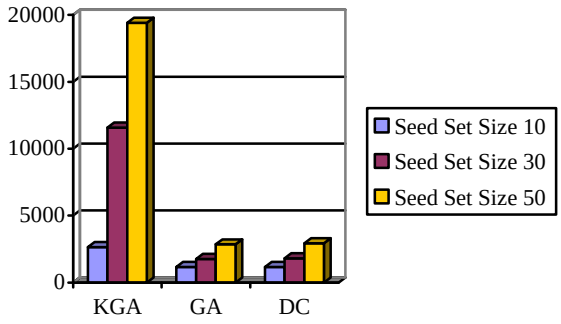

Fig. 8.The Results of Running Time on UCSOCIAL

VI. CONCLUSION

In social networks, the behavior and ideas exchange takes place through person to person interaction. In influence maximization problem, the goal is to find a small subset of network nodes that would influence the most node count in the network. In this context the available algorithms have both the advantages and disadvantages. A group of algorithms can provide the greatest influence spread value and their accuracy is relatively high, but are not scalable, due to their high running time. As to computational time some are reduce the solution running time, while not accuracy in finding the influential nodes. Attempt is made here to provide an appropriate solution for influence spread in social networks. The novelty here is the application of explicit multimodal evolutionary algorithms to serve this purpose. By applying the deterministic crowding algorithm, the search space is divided into different regions all searched in parallel. This algorithm is tested on four actual networks. In these networks, this proposed algorithm has led to a better improvement by 3.2% and 4.8% in relation to KGA and GA, respectively. In all subject networks, as to efficiency, there is more runtime than GA but in relation to KGA a 54.8% reduction in running time is evident.

As to the future works, this algorithm can be tested on other newly developed diffusion models. The online social networks consist of individuals. If an individual is selected as a seed node, it is also important to consider what benefit will be obtained by activating that. At the same time, for influence spread of time dependent events (like political pre-election campaigns) consideration of diffusion time is also important.

REFERENCES

1. S. Wasserman and K. L. M. Faust, Social network analysis: Methods and applications, vol. 8. Cambridge: Cambridge University Press, 1994.
2. E. Otte and R. Rousseau, “Social network analysis: a powerful strategy, also for the information sciences,” Journal of Information Science, vol. 28, no. 6, pp. 441–453, 2002.
3. L. Rashotte, “Social Influence,” The Blackwell Encyclopedia of Sociology, vol. 9, pp.4426–4429, 2007.
4. M. Li, X. Wang, K. Gao, and S. Zhang, “A Survey on Information Diffusion in Online Social Networks: Models and Methods,” Information, vol. 8, no. 4, p. 118, 2017.
5. P. Domingos and M. Richardson, “Mining the network value of customers,” Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 01, 2001.
6. 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.
7. 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.
8. D. Peleg, Local majority voting, small coalitions and controlling monopolies in graphs: a review. Rehovot: Weizmann Institute of Science. Department of Applied Mathematics and Computer Science, 1996, pp. 152-169.
9. S. Banerjee, M. Jenamani, and D. K. Pratihar, “A Survey on Influence Maximization in a Social Network.,” CoRR, vol. abs/1808.05502, 2018.
10. 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.
11. 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.
12. W. Chen, C. Wang, and Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks,” Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 10, pp. 1029–1038, 2010.
13. W. Chen, Y. Yuan, and L. Zhang, “Scalable Influence Maximization in Social Networks under the Linear Threshold Model,” 2010 IEEE International Conference on Data Mining, pp. 88–97, 2010.
14. A. Goyal, W. Lu, and L. V. Lakshmanan, “SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model,” 2011 IEEE 11th International Conference on Data Mining, pp. 211–220, 2011.
15. 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, p. 199, 2009.
16. Q. Jiang, G. Song, G. Cong, and Y. Wang, “Simulated Annealing Based Influence Maximization in Social Networks,” Twenty-Fifth AAAI Conference on Artificial Intelligence Simulated, pp. 127–132, 2011.
17. H. Ma, H. Yang, M. R. Lyu, and I. King, “Mining social networks using heat diffusion processes for marketing candidates selection,” Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM 08, pp. 233–242, 2008.
18. A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, “A data-based approach to social influence maximization,” Proceedings of the VLDB Endowment, vol. 5, no. 1, pp. 73–84, 2011.
19. J.-R. Lee and C.-W. Chung, “A Query Approach for Influence Maximization on Specific Users in Social Networks,” IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 2, pp. 340–353, 2015.
20. S. W. Mahfoud, “Crowding and preselection revisited,” Parallel Problem Solving from Nature, pp. 27–36, 1992.

AUTHORS PROFILE

Navid Kaveh, received the B.Sc. degree in Computer Engineering in 2014 from University of UAST, Isfahan, Iran, and currently M.Sc. student at the Faculty of Engineering of the Sheikhbahaee University (SHBU), Isfahan, Iran.

Mehdi Bateni, is an assistant professor of computer engineering at the Faculty of Engineering of the Sheikhbahaee University (SHBU). He received his B.Sc. in Computer Engineering in 1997 from University of Isfahan, Isfahan, Iran and his M. Sc. in Computer Engineering from Ferdowsi University of Mashhad, Mashhad, Iran in 2000. He received his Ph.D. in Computer Engineering in 2012 from University of Isfahan, Isfahan, Iran.