

Influence Maximization in Social Networks using Learning Automata

Afshin Mohammadi
Research Scholar

Department of Computer Engineering, Sanandaj
Branch, Islamic Azad University, Sanandaj, Iran

Keyhan Khamforoosh
Assistant Professor

Department of Computer Engineering, Sanandaj
Branch, Islamic Azad University, Sanandaj, Iran

ABSTRACT

Influence maximization problem is one of the challenges in online social networks. This problem refers to finding a small set of members of a social network, by activation of which information propagation can be maximized using one of the propagation models such as independent cascade model. For the maximization problem, the greedy algorithm has been presented which is close to optimal response by 67% in terms of accuracy; but, the problem of this method is its inefficiency in the social networks with a large number of members. The performed works on the improvement of the greedy algorithm have been mostly faced with the problem of scalability, dependence on graph structure, or need for large memory. In this paper, a method was presented using automata learning which could preserve its efficiency in large social networks and obtain results with near-optimal values. For this purpose, space of the problem was reduced by removing low-degree nodes and the effective nodes for starting propagation in social network was found by automata learning which is optimal for achieving global optimization. The obtained results of this paper showed that the proposed method was efficient in large social networks and its results were close to the ones obtained by the greedy algorithm in terms of accuracy.

Keywords

Social networks, Influence maximization, Propagation models, Learning automata

1. INTRODUCTION

Influence maximization problem refers to the maximization of social influence at the level of a network, such as propagation of a culture, acceptance of an innovation, and advertisement of a product among users of a social network. Social influence means change of thoughts, beliefs, and views of a person as a result of his/her interaction with another person or a group [1]. Goal of the influence maximization problem is to select a number of people from among the members of a social network for starting information propagation so that propagation rate in the network could be maximized [2]. To investigate the influence rate of propagation, propagation models are used, among which independent cascade and linear threshold propagation models can be mentioned [3,4]. Social networks can play a very important role in the field of influence propagation [5]; but, propagation in social networks can be rapidly removed and can penetrate into different groups. Starting fast and effective propagation requires proper selection of initial people at the beginning of propagation. One of the applications of the influence maximization problem is viral marketing [6]. Research has shown that people have more trust in the recommendations of their friends than advertisement of the mass media and others [7,8]. For example, a company intends to sell a limited number of its products to the effective people of a network in order to

advertise its product and maximize its sale rate. These people can recommend purchasing that product to their friends via their relations and influence in that network; if these people have been properly selected, their friends can also recommend the product to other friends and thus generate a wave of advertisement and sale for that company. If these people have been selected by mistake in terms of their influence, they will lose the limited budget of that company for its advertisement. Kempe in [3] showed that influence maximization problem is one of the NP-hard problems and presented the greedy algorithm with the approximation of 67% for it. The greedy algorithm does not have the required efficiency in social networks with a large number of nodes, which is related to the use of the Monte Carlo simulation method for obtaining correct approximation of the influence spread of each node. In the Monte Carlo simulation method, the propagation model is executed for at least 10000 times for each algorithm node to obtain correct approximation of node influence. Kempe [3] showed that studying influence spread using independent cascade model and linear threshold methods is one of the #P-hard problems, which is one of the reasons for the inefficiency of the greedy algorithm. In order to improve the greedy algorithm, different methods have been presented; some methods have solved its scalability problem by improving the greedy algorithm, but have not been very successful. Some others are based on the graph structure; but, their accuracy is not high in real social networks.

In this paper, to solve the influence maximization problem using centrality criteria, first, the nodes lacking any influence were excluded from the problem space. Then, using one of the evolutionary algorithms, called learning automata, a method was proposed which could find effective nodes in a shorter period of time than the greedy algorithm. Innovation of this paper can be mentioned as follows:

- Reducing the problem space by excluding the nodes lacking influence using centrality criteria,
- Using learning automata; it is known that this method has been first used for influence maximization.
- Optimizing learning automata using a heuristic method to increase speed in reaching a globally optimal response.

The rest of this paper is organized as follows:

In Section 2, background of the research and previous works in the field of influence maximization are studied. In Section 3, the proposed method, the method presented for reducing problem space, manner of problem mapping into a solution using learning automata, and heuristic method for increasing convergence speed of automata are reviewed. In Section 4, the obtained results are studied and compared. Finally, conclusions

and further works are mentioned in Section 5.

2. REVIEW OF THE RELATED LITERATURE

The influence maximization problem was first presented by Domingos [7] for finding beneficial customers for companies in order to sell their products. A model called customer network value was also presented for it, which could indicate the profit obtained by the sale of products to the customers who have been affected by another buyer.

If $f(s)$ is a submodular and monotonic function, Kempe [3] proved that finding a k -member set such as S which can maximize $f(s)$ is an NP-hard optimization problem. He also proposed a greedy algorithm for this problem whose response was an approximate of $(1 - \frac{1}{e})$. A submodular function is a function which satisfies the properties of diminishing returns; thus, the marginal gain of adding an element to set S is at minimum higher than the marginal gain of adding the same element to a set larger than set S . Monotonicity of a function means that adding an element to set S does not lead to the reduced returns of function $f(s)$.

Submodular Function:

$$\text{for all } S \subseteq T \subseteq V, \text{ all } v \in V \setminus T \\ f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$$

Monotonic Function:

$$\text{for all } S \subseteq T \subseteq V \\ f(S) \leq f(T)$$

The proposed greedy algorithm acted better than all the presented methods in terms of influence spread; but, it had a major weakness: inefficiency in large social networks. The reason for the slowness of this algorithm was use of the Monte Carlo simulation method for obtaining an approximation close to the response for extending the influence of the selected nodes. In algorithm 1, the greedy algorithm presented by Kempe [3] is given.

Algorithm 1: Greedy algorithm for influence maximization

```

1: initialize  $S = \emptyset$ 
2: for  $i = 1$  to  $k$  do
3:   select  $u = \operatorname{argmax}_{w \in V \setminus S} (f(S \cup \{w\}) - f(S))$ 
4:    $S = S \cup \{u\}$ 
5: end for
6: output  $S$ 

```

To solve the scalability problem of the greedy algorithm, some studies have been conducted which can be classified as greedy algorithm optimization, heuristic methods, and evolutionary algorithms.

2.1. Improving greedy algorithm

Leskovec [9] presented cost effective lazy forward (CELFF) algorithm using sub-modular characteristics of maximization problem, which was 700 times faster than the conventional greedy algorithm in terms of speed. Then, Goyal [10] presented CELFF++ algorithm by improving CELFF algorithm, which was 27% faster than CELFF algorithm. In [11], Cheng proposed static greedy algorithm by removing the edges of main graph with the probability of $1 - p_{u,v}$ ($p_{u,v}$ is the probability of activating node v by node u) and producing snapshot graphs which considerably reduced the execution number of the Monte Carlo method. Capone et al. [12] could

increase the speed of the static greedy algorithm by making some improvements.

2.2. Heuristic methods

By presenting degree discount algorithm, Chen et al. [13] proposed a method which was much better than the presented methods based on the greedy algorithm in terms of speed; but, this method was dependent on graph structure and was not accurate enough. PMIA (prefix excluding maximum influence path) method which was presented by Wang et al. [8] had some simple adjustable parameters for determining the balance between execution speed and influence propagation rate at each node. To produce this branch, maximum influence paths are first calculated between each pair of nodes in the network using the shortest path algorithm. Then, the paths with lower probability than the adjustable threshold θ are excluded. Afterwards, the produced MIPs (maximum influence paths) for each node are integrated with each other, a branch structure is produced, and the propagated influence of these local branches is considered. Problem of this method is the use of a large amount of memory for maintaining branches in the memory. Jung et al. [14] introduced IRIE (influence ranking influence estimation) algorithm by combining the advantage of influence rank (IR) and influence evaluation (IE) methods, which was more scalable and resistant than the previous methods. This method was twice as fast as the PMIA method and had less memory.

2.3. Evolutionary algorithms

Evolutionary algorithms include some algorithms for searching among several points in the problem space based on random searching. These methods use a series of initial operations for problem solving and deal with the suitable solution during a series of iterations. Gui-sheng et al. [6] studied maximization problem using intelligent methods such as GA, DE, and PSO. To reduce the space of the problem, they reduced the network size by analyzing the dataset and sampling the data considering the characteristic of power law. The idea of this method for reducing the problem space is that at least half of the social network users are alone and cannot affect anybody. So, time should not be spent on such users. Yang et al. [15] presented a method based on ant colony algorithm to solve the maximization problem. In this method, a complete graph is made for stating the primary social network and a seed set is randomly selected for this graph. Then, propagation rate of their influence is investigated. The problem of this method is the production of a complete graph from all the nodes in social network, which slows down the algorithm and reduces accuracy in large social networks. What distinguishes the results of this paper and those of the previous works is speed of algorithm and the obtained results.

2.4. Learning automata

Learning automata is one of the evolutionary algorithms, which has a set of action; at first, selection probability of this set of action is equal and one of the actions is randomly selected and applied to an environment. Considering the response of the environment to the selected action, a rewarded or penalty will be given, which will increase or reduce the selection probability of that action at the next iteration. Learning automata tries to select the optimal action by reducing and increasing the probability of its actions. The environment can be shown with $E \equiv \{\alpha, \beta, c\}$. Where $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ is set of inputs to environment, $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$ is set of outputs, and $c \equiv \{c_1, c_2, \dots, c_r\}$ is set of penalty probabilities. The input is one of the r selected

automata actions. Output (response) of the environment to each action i is specified by β_i [16]. Figure 1 shows the relation of learning automata with the environment.

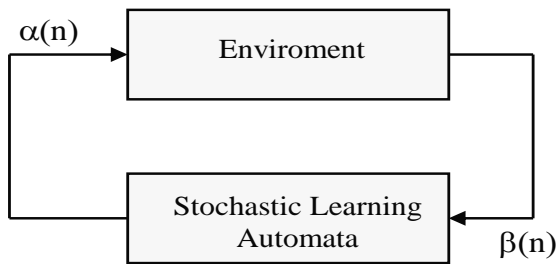


Figure 1: Relationship of the environment with learning automata [17]

2.4 Distributed automata

A learning automata is applied for doing simple problems and, in the case of complex problems, the cooperation of automata with each other should be used. Distributed learning automata is a network of learning automata which cooperate with each other for solving a specific problem. The number of actions of distributed automata is equal to the number of automata connected to it and, by selecting an action from among the actions related to the automata, one automata which is connected to it will be activated and this will continue until reaching a leaf automata (an automata which is in interaction with the environment). Then, the environmental feedback will be sent to the distributed automata and it will update its probability vector based on the environmental feedback. Fig. 2 shows distributed automata [17].

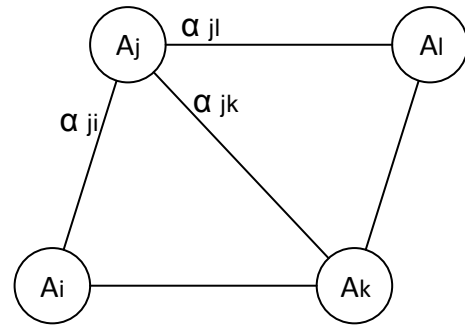


Figure 2: Distributed automata [17]

3. PROPOSED METHOD

Goal of the influence maximization problem is to find a subset of node S with kelement from among the nodes available in a social network so that function $f(S)$ can be maximized under one of the information propagation models. In this paper, elements of set S were selected in each execution of the algorithm and then their propagation was investigated under independent cascade model and optimality or non-optimality of that set was determined. In independent cascade model, nodes can have active or inactive status. At first, at time $t = 0$ ($t = \{0,1,2,\dots,n\}$), only nodes of set S are active and other nodes are inactive. These nodes can change the status of their neighboring nodes by influencing them at time $t = 1$ and make them active. At time $t = 2$, they can activate their neighboring nodes and continue the trend until no other node can be activated. In this model, the activated nodes cannot return to their inactive status. In Figure 3, diagram of the trend of the proposed method is shown and its different parts are also mentioned.

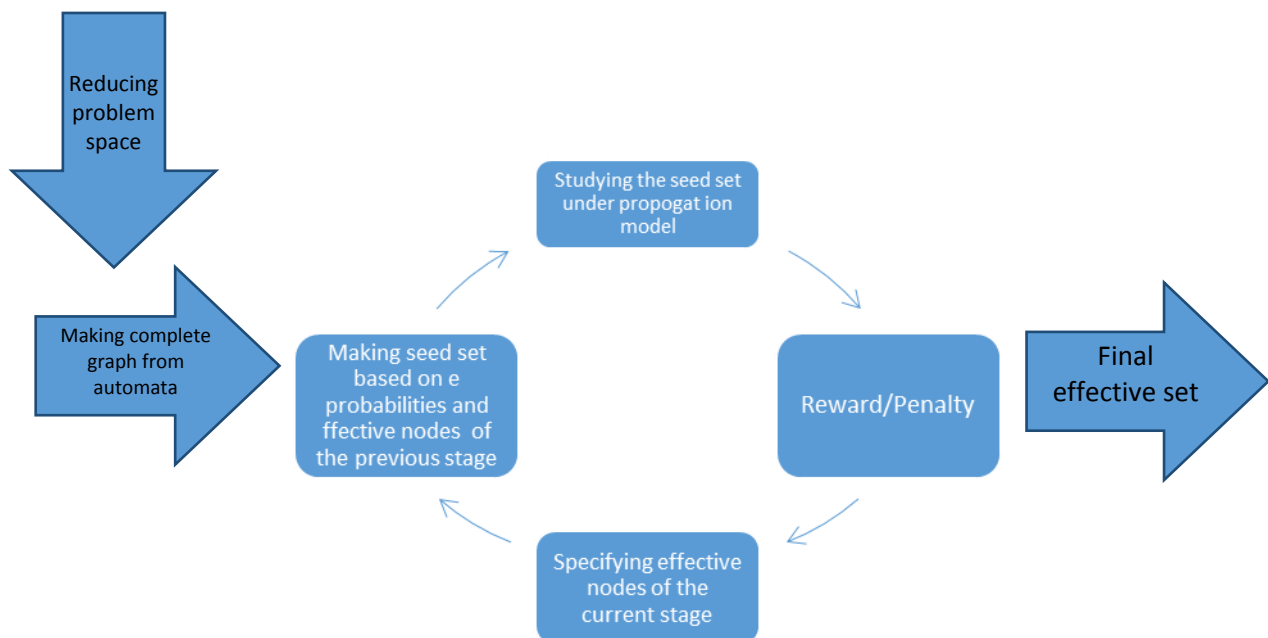


Figure 3: Trend of the proposed method

3.1.Reducing the problem space

In the influence maximization problem, if all nodes of the network are considered as effective nodes, the algorithm will slow down and accuracy will be reduced. Assume that we select a node with degree 1 as an effective node. Thus, the

evaluated function is executed, which naturally hasan ineffectiveresult,because a node with a low degree cannot start an effective propagation in the entire network. Calculating the evaluation function for all of these nodes which form a large population of network nodes slows the algorithm down. Gui-

sheng et al. [6] showed that most of the network nodes are only connected to a few other nodes, while a few number of large nodes (with power in terms of centrality criterion such as degree) are connected to many other nodes. Considering this characteristic which exists in social networks, the problem space can be limited to a smaller interval of powerful nodes and effective people can be selected, which increases the convergence speed of the algorithm and its accuracy. In the proposed method, first, mean degree of the total nodes of the network was calculated to reduce the problem space and the nodes with lower than mean degree of the graph were ignored in the problem space. Figure 4 shows the frequency of the nodes available in the graph used in this paper, which is related to co-authorship network. As shown in this figure, the number of nodes with degrees 1 to 3 in the used graph was several times of the total number of nodes with higher degrees. Mean degree of the used graph in this paper was 4.12 and the nodes with the degree below 4 were not involved in the problem space.

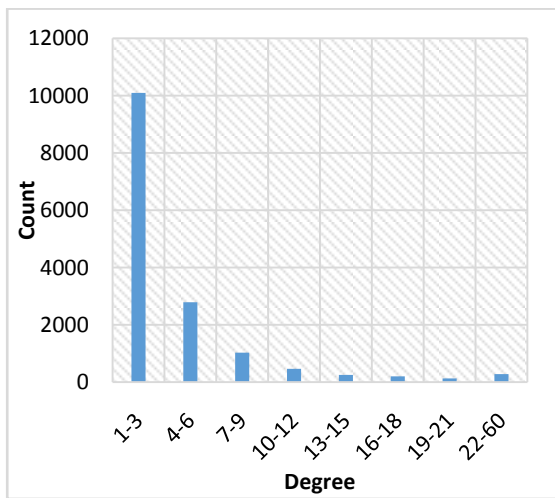


Figure 4: Frequency nodes of NetHEPT dataset in different degrees

3.2. Mapping the problem into learning automata

In this problem, distributed learning automata is used instead of automata due to the largeness of the search space so that each of the nodes in the problem space is assumed as a learning automata and a complete graph of these automata is formed [18]. Figure 5 shows the distributed automata, in which the nodes in the problem space are assumed as a graph of learning automata. Given that the number of the candidate nodes in the problem space is r -node, the probability of selecting each node at time $t = 0$ will be obtained by Relation $\frac{1}{(r-t)}$. This step will continue until the seed set is completed. After completion of the seed set, its elements are sent to the evaluation function which is propagation model (environment); in case the result is optimal, the selected nodes will be rewarded by Relation 1.

Relation 1 (Rewards of desirable responses received from the environment)

$$p_i(n+1) = p_i(n) + a(1 - p_i(n)) \quad \alpha(n) = \alpha_i$$

$$p_i(n+1) = p_j(n) + ap_i(n) \quad \alpha(n) = \alpha_i, \quad \forall j \neq i$$

Relation 2 (Penalties of undesirable responses received from the environment)

$$p_i(n+1) = (1-b).p_i(n) \quad \alpha(n) = \alpha_i$$

$$p_i(n+1) = \frac{b}{r-1} + (1-b).p_j(n) \quad \alpha(n) = \alpha_i, \quad \forall j \neq i$$

In case the result of the evaluation function is not optimal, the selected nodes will be fined by Relation 2. The goal of rewarding and fining the nodes is to increase the probability of selecting optimal actions (effective nodes) in the seed set of the next steps and reducing the probability of ineffective nodes.

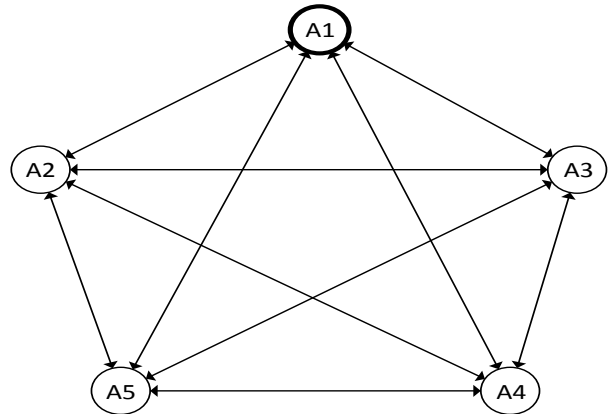


Figure 5: Mapping learning automata into the influence maximization problem

3.3. Increasing convergence speed

In this paper, a heuristic method was used to rapidly achieve the optimal solution by producing low generations in learning automata so that the effective nodes were specified among the selected nodes in the seed set of step n and were used in step $n+1$, irrespective of their probability. The idea of this method originates from the fact that it is not possible for all the nodes to be effective in the primary generations for selecting the members of the seed set. Therefore, effective nodes can be separated from the ineffective ones by calculating the influence of each node in current generation. In this method, the number of nodes which can be transferred from the previous generation to the next one is limited to at most 75% of the number of elements of seed set in order to prevent the inclusion of response in local optimum and a larger number of this value cannot be transferred to the next generation. In each step, it is possible to exclude effective nodes by finding more effective nodes.

Effective nodes for being placed in the set of the next step are selected as follows: If we assume that the number of elements of seed set S is k , $\sigma(S)$ is the influence of set S , and value θ is calculated using Relation $\theta = \frac{\sigma(S)}{K}$, the nodes which can be placed in the seed set of the next generation are $\partial(s_n) \geq \theta$ so that $s = \{s_1, s_2, \dots, s_k\}$. Algorithm 2 shows the steps of the proposed method along with the required inputs of the algorithm.

Algorithm 2: Optimized learning automata algorithm for solving influence maximization problem

Input: Graph $G = (V, E)$, Size k , α, β , GenerationsCount;
Output: set of top k influential nodes S ;
1: $\alpha \rightarrow 0.003, \beta \rightarrow 0.003, |S| = k, bestNodes = \emptyset, N = \emptyset, gens = 0; inf = 0, bestInf = 0$
2: while $(i < GenerationsCount)$
3: $gens = \frac{3k}{4}$

```

4:if (bestNodes.count > gens) { put bestnodes from
5:N ← bestNodes(gens)           previous generation
6:else N ← bestNodes             to current seedset}
7: while (N.count < k)
8:select N nodes N ⊂
V randomly based on their probabilities;
9:inf = calculate σ(N)
10:if (inf > bestInf)
11:  reward N
12:  get bestnods of N
13:  S ← N
14: else
15:  penalty N
16:return S

```

4. OBTAINED RESULTS

To prove the convergence speed of the improved automata method, its results were compared with those of the conventional distributed automata method. The used evaluation function in the proposed method was independent cascade model. Finally, the obtained results in this paper were compared with those of the high degree methods (selecting nodes with the highest degree of other nodes), random method (random selection of seed set)[3], degree discount method [13] which is one of the heuristic algorithms, and also the greedy algorithm [3].

4.1. Used dataset

The dataset used in this method was extracted from High-Energy Physics-Theory part of the website arxiv.org, which includes information related to 1991 to 2002. In this dataset, each node indicates an author and each edge shows cooperation of two writers in writing a paper. The graph used in this paper was weighted undirected graph containing 15233 nodes and 58871 edges. Weight of edge (u, v) indicates the influence of node u on node v whose value is obtained from $\frac{1}{deg_v}$, where deg_v means degree of v .

4.2. Adjusting parameters

In the proposed method, there were not many parameters for adjustment. The only adjustable parameters in this method were parameters α, β which indicate reward and penalty for the set of optimal and non-optimal seeds in order to reduce or increase the probabilities of selecting nodes. In this method, 0.003 which was obtained using trial and error was selected for both of them. In all the results, the execution number of the Monte Carlo method was set to 10000. To study the influence rate or evaluation function, the independent cascade model was used. All the simulations were made on a household system with Intel core i7 CPU specifications and frequency of 1.7GHz for each core and main memory of 4 GB.

4.3. Results

Convergence of the proposed method: In Figure 6, the conventional distributed automata algorithm and the proposed optimized automata were compared with each other in terms of influence spread in an equal environment to obtain the number of 10-node seed set. The results showed better performance of the proposed method and that the conventional distributed automata would not finally converge to globally optimal results when the number of the produced generations was low. Also, convergence speed of the conventional distributed automata toward the global optimum was very slow. Results in Figure 6 demonstrate the influence spread when the produced generations were 50, 100, and 150, indicating

that convergence speed in the conventional automata method was very slow. Also, results of Figure 7 represented that the production of more generations in the conventional automata was inefficient in terms of spent time. The obtained results in Figure 7 showed that, with increasing the number of produced generations, the spent time on the production of optimal response of the proposed method for optimizing automata led to reduced speed of the algorithm so that the required time for the proposed method linearly increased compared with the conventional automata. But, the problem which led to the priority of the proposed method for the automata optimization was fast achievement of optimal response and lack of need for the production of more generations compared to the conventional automata. As shown in the results, the conventional automata reached the response of 44.14 by producing 50 generations within 16 min, while the proposed method reached the response of 62.31 to optimize automata by producing 50 generations within 20 min; this result showed that there was no need to produce more generations. Therefore, by comparing the obtained results for both of the introduced methods, it can be concluded that using optimization method for automata acts better in fast achievement of globally optimal response within a suitable period of time.

Comparing the proposed method and heuristic methods:

Figure 8 shows the comparison of the proposed method and the high degree, degree discount, and random methods for producing 5, 10, 20, 30, and 50-element seed sets. Independent cascade model was used to study the influence of each method. Results showed higher accuracy of the automata-based method compared with other methods. In this diagram, random method acted very inefficiently in terms of accuracy, which can prove the convergence and usefulness of learning automata, because learning automata first starts working with a seed set which is selected randomly. Degree discount method was not very different from high degree method in terms of accuracy, which can be attributed to the dependence of both methods on the degree of the selected nodes.

Comparing speed of the proposed method and greedy algorithm:

To prove the optimality of the results obtained by the automata-based method, it was compared with the greedy algorithm. As shown in Figure 9, the proposed method was very close to the greedy algorithm in terms of results and had better performance than the greedy method in terms of execution time. These results showed that the proposed method was scalable, in which a balance in terms of time and accuracy can be obtained by adjusting its input parameters. In other words, if accuracy of the algorithm is important, better results can be achieved by increasing the number of generations which could increase execution time; thus, a suitable result can be achieved within a short time by decreasing the number of produced generations.

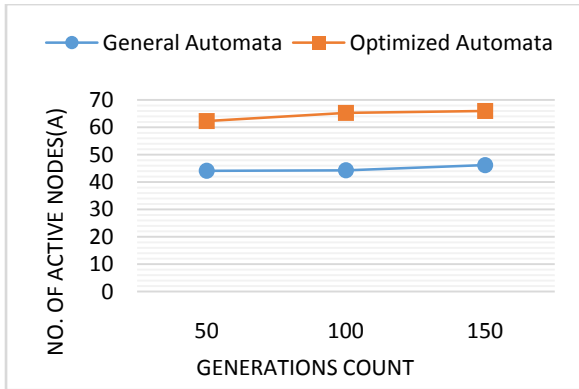


Figure 6: Studying influence spread (accuracy) of distributed and optimized automata methods

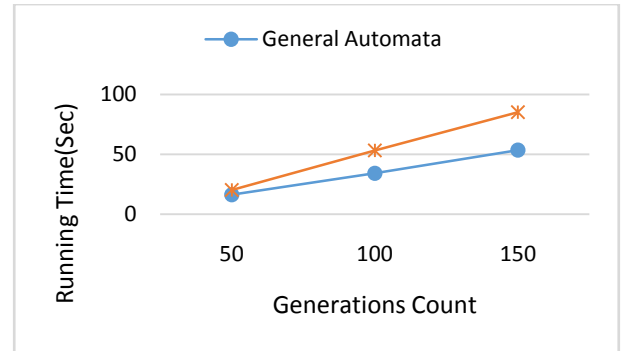


Figure 7: Studying time spent on reaching optimal response

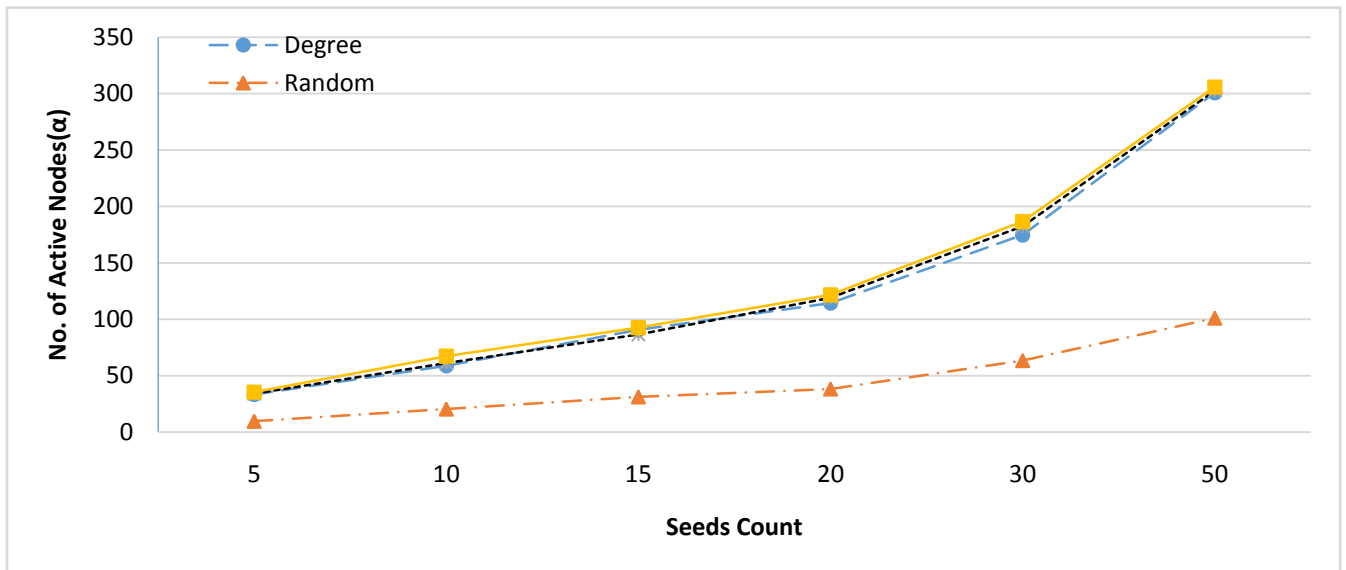


Figure 8: Comparing the proposed method and heuristic methods

In this research, the greedy algorithm was executed only in three states of 5, 10, and 20-element seed sets using independent cascade model due to the lack of access to a computer with a suitable server. Figure 10 shows the comparison of learning automata with the greedy algorithm in terms of accuracy. The results showed higher speed of this method than the greedy algorithm so that execution time vector of the learning automata method compared with the greedy algorithm was a direct line. For example, the learning automata algorithm needs 410 min to produce a 5-element seed set, while this time is reduced to 10 min (approximately 40 times reduction) in the learning automata method; this difference will increase with increasing the number of elements of the seed set. Thus, production time of a 20-element set is 1623 min in the greedy algorithm, while this value is 20 min (approximately 80 times reduction) in the learning automata.

5. CONCLUSION AND FURTHER WORKS

In this research, the learning automata method was used to solve the influence maximization problem. Also, using a heuristic method and the effective nodes of the previous step for producing the seed set of the next generation, convergence

speed of the automata was increased. The obtained results showed that this method could act better than the degree discount and high degree methods in terms of accuracy and the spent time of these two methods can be neglected by presenting better results. Moreover, accuracy of the proposed method was close to that of the greedy algorithm and its execution time was much shorter than that of the greedy algorithm. These results demonstrated that the presented method was scalable. To improve the presented method in future studies, execution speed of the algorithm can be considerably increased by creating a semi-local method for investigating influence spread. Hierarchical learning automata or game of automata can be used to find a method with higher speed and accuracy in terms of convergence. In this research, results were only executed on one of the propagation models (independent cascade). As a new case study, a method can be proposed to execute on all propagation models and achieve optimal results.

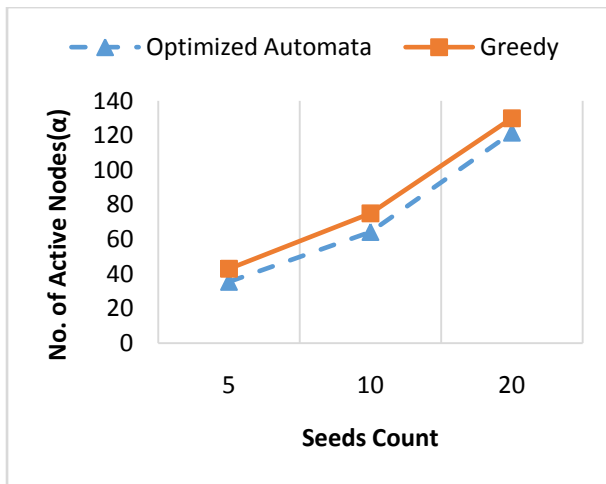


Figure 9: Comparing optimized learning automata and the greedy algorithm (in terms of accuracy)

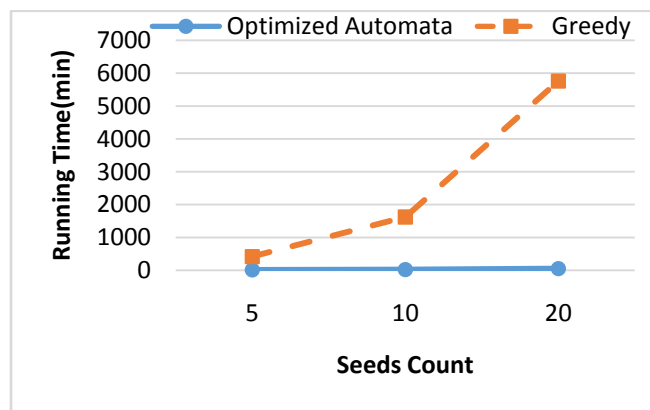


Figure 10: Comparing improved learning automata and the greedy algorithm (in terms of time)

6. REFERENCES

- [1] Rashotte, L., Social influence. The blackwell encyclopedia of social psychology, 2007. 9: p. 562-563.
- [2] Chen, W., Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. in Data Mining (ICDM), 2010 IEEE 10th International Conference on. 2010. IEEE.
- [3] Kempe, D., J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. 2003. ACM.
- [4] Saha, B. and S. Navlakha, Survey on Information Diffusion. 2008.
- [5] Guille, A., et al., Information diffusion in online social networks: A survey. ACM SIGMOD Record, 2013. 42(1): p. 17-28.
- [6] Gui-sheng, Y., W. Ji-jie, and L. Jia. Intelligent Viral Marketing algorithm over online social network. in Networking and Distributed Computing (ICNDC), 2011 Second International Conference on. 2011. IEEE.
- [7] Domingos, P. and M. Richardson. Mining the network value of customers. in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. 2001. ACM.
- [8] Wang, C., W. Chen, and Y. Wang, Scalable influence maximization for independent cascade model in large-scale social networks. Data Mining and Knowledge Discovery, 2012. 25(3): p. 545-576.
- [9] Leskovec, J., et al. Cost-effective outbreak detection in networks. in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007. ACM.
- [10] Goyal, A., W. Lu, and L.V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. in Proceedings of the 20th international conference companion on World wide web. 2011. ACM.
- [11] Cheng, S., et al., StaticGreedy: solving the scalability-accuracy dilemma in influence maximization. proceedings of CIKM2013, 2012.
- [12] Capone, L., N. Di Mauro, and F. Esposito, Optimizing a static greedy algorithm for influence maximization. 2013.
- [13] Chen, W., Y. Wang, and S. Yang. Efficient influence maximization in social networks. in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009. ACM.
- [14] Jung, K., W. Chen, and W. Heo, IRIE: A scalable influence maximization algorithm for independent cascade model and its extensions. 2011.
- [15] Yang, W.-S. and S.-X. Weng, Application of the Ant Colony Optimization Algorithm to the Influence-Maximization Problem. 2012.
- [16] Narendra, K.S. and M. Thathachar, Learning automata-a survey. Systems, Man and Cybernetics, IEEE Transactions on, 1974(4): p. 323-334.
- [17] Mousavian, A., A. Rezvanian, and M.R. Meybodi, Solving Minimum Vertex Cover Problem Using Learning Automata. arXiv preprint arXiv:1311.7215, 2013.
- [18] Soleimani-Pouri, M., A. Rezvanian, and M.R. Meybodi. Solving maximum clique problem in stochastic graphs using learning automata. in CASoN. 2012.