Maximum Degree Based Heuristics for Influence Maximization

Maryam Adineh
Computer Engineering Department
Ferdowsi University Of Mashhad
Mashhad, Iran
Email: maryam.adineh@mail.um.ac.ir

Mostafa Nouri-Baygi
Computer Engineering Department
Ferdowsi University Of Mashhad
Mashhad, Iran
Email: nouribaygi@um.ac.ir

***Abstract*—Influence maximization is the problem of selecting a subset of individuals in a social network that maximizes the influence propagated in the network. With the popularity of social network sites, and the development of viral marketing, the importance of the problem has been increased.**

Finding the most influential vertices, called seeds, in a social network graph is an NP-hard problem, and therefore, time consuming. Many heuristics are proposed to find a nearly good solution in a shorter time. In this paper, we propose two heuristic algorithms to find a good seed set. We evaluate our algorithms on several well-known datasets and show that our heuristics achieve the best results (up to 800 improvements in influence spread) for this problem in a shorter time (up to 10% improvement in runtime).

I. Introduction

Interaction of people in a social network provides a lot of information about their behavior and the structure of the social graph. It has also made the social network a good platform to spread information, believes, innovation, and so on. One of the important applications of the spread of influence in social networks is viral marketing. For example, consider a company that wants to market its product in a social network. A simple and low-cost approach is to select a subset of individuals to offer the product to them, so they will encourage their friends to buy it. This behavior is like spreading a virus in a society. The important portion of this type of marketing is the initial selection of most influential individuals. This problem is known as influence maximization.

Influence maximization problem was first introduced by Domingos and Richardson [1], [2]. Kempe *et al*. [3] formally defined the problem and proved that it is NP-hard. They also introduced two monotone and submodular diffusion models for the spread of influence, namely independent cascade model and linear threshold model, and proved that a greedy hill climbing algorithm approximates the solution within 63% of the optimal solution for these models.

Because the greedy algorithm runs a simulation several thousand times to find the marginal influence of each vertex, and therefore is time-consuming, many heuristics are proposed to improve its performance. Although the heuristics have reduced the running time, they are still time-consuming in large-scale networks, which is the case for most of social networks. On the other hand, degree-centrality based heuristics are very fast even on large-scale networks. Although they don't guarantee the accuracy of the solution, they find solutions as good as the solution of the greedy algorithm.

978-1-5386-9569-2/18/$31.00

In this paper, we propose two maximum-degree based heuristics that are very fast in running time and improve the results of all previous degree-based heuristics. In other words, the results of our algorithms are close to the results produced by the greedy algorithm, while its running time is close to degree-based heuristics.

The rest of this paper is organized as follows. The related works are reviewed in Section II. In Section III the formal definition of the problem is described. We propose our heuristics in Section IV and present the experimental results in Section V. Finally we conclude the paper in Section VI.

II. Related Work

Influence maximization problem was formally defined by Kempe *et al*. [3] and proved to be NP-hard. They proposed a greedy hill climbing algorithm that yields a solution within $(1 - \frac{1}{e} - \varepsilon)$ factor of optimal solution for two models they introduced. In the approximation ratio of the algorithm, e is the base of the natural logarithm and ε is any positive real number which is error of the Monte-Carlo simulations. Choosing a small value for ε increases the running time, while choosing a large value for it reduces the quality of result. In their algorithm, the most influential vertices are selected by their estimated marginal influence. Since estimated marginal influence is computed by a large number of simulations, the algorithm is inefficient.

In order to improve the efficiency of computations, many studies have been made. Leskovec *et al*. [4] proposed Cost-Effective Lazy forward (CELF) optimization that reduces computation cost of influence spread using sub-modularity property of objective function.

Chen *et al*. [5] proposed new greedy algorithms for independent cascade and weighted cascade models. They make the greedy algorithm faster by the combination of their algorithms with CELF. They also proposed *degree discount* heuristic which generates close influence spread to greedy algorithms while is much faster and returns better solution than simple degree and distance centrality heuristics.

In order to avoid running repeated influence propagation simulations, Borgs *et al*. [6] generate a random hypergraph according to reverse reachability probability of vertices in the original graph and select k vertices that are most covered by hyperedges in the hypergraph. They guarantee $(1 - \frac{1}{e} - \varepsilon)$ approximation ratio of the solution with probability at least $1-1/n^l$. Tang *et al*. [7], [8] proposed TIM and IMM to cover drawbacks of Borgs *et al*.'s algorithm [6] and improved its running time.

Bucur and Iacca [9] and Krömer and Nowaková [10] used genetic algorithms for influence maximization problem. Weskida and Michalski [11] use GPU acceleration in their genetic algorithm to improve its efficiency.

There are some community-based algorithms for influence maximization problem that partition graph into small subgraphs and select most influential vertices from each subgraph. Chen *et al*. [12] use H-Clustering algorithm and Manaskasemsak *et al*. [13] use markov clustering algorithm for community detection. Song *et al*. [14] divide the graph into communities, then select the most influential vertices by a dynamic programming algorithm.

III. Problem Definition

In this section we formally define influence maximization problem and independent cascade diffusion model.

We consider a social network as an undirected graph $\mathcal{G} = (V, E)$ where V is the set of individuals of size n, and E is the set of relationships of size m.

Let S be the subset of vertices selected to initiate the influence prorogation and $I(S)$ be the influence spread by S.

A. Diffusion Model

There are many diffusion models for influence propagation. In this paper we focus on *Independent Cascade Model* (ICM). In the independent cascade model for each edge (u, v), a newly activated vertex u can activate v with probability $p_{u,v} \in [0, 1]$. The diffusion process is as follows:

Let S_i be the set of activated vertices in timestamp i, In timestamp $i + 1$ each vertex $u \in S_i$ has a chance to activate each of its inactive neighbors. Once u tried to activate neighbor v, either it succeeds or not, u will not try to activate v in later steps. Furthermore, each activated vertex remains active in all subsequent timestamps. This process terminates when there are no more activations possible.

B. Influence Maximization Problem

In the problem of influence maximization, given a graph $\mathcal{G}$, a constant k and a diffusion model $\mathcal{M}$, we are asked for a set S of k vertices with maximum influence propagation, $I(S)$. In this paper, we focus on the independent cascade model as $\mathcal{M}$, and leave extending the algorithms to other models to future work.

IV. Proposed Algorithms

In this section we describe our heuristics for influence maximization problem under the independent cascade (IC) model. Although the greedy algorithm and its variants make a guarantee about the goodness of the solution in terms of approximation ratio, they are time-consuming on large-scale social networks. Since heuristics are much faster and applicable on large networks, we propose two novel heuristics based on degree-centrality that nearly match the solution of the state of the arts while running in a short time.

Degree-centrality heuristics select k vertices with maximum degrees as the most influential individuals. Another clear and more accurate approach called single discount by Chen *et al*. [5] works as follows. When selecting vertex u as a seed, degree of each neighbor v decreases according to the number of common edges they have.

Although these heuristics have large spread of influence, they don't return appropriate solution because when vertex u is selected as a seed, its neighbor v will be influenced by u. Also when probability p is high, the influence of u even on its multi-hop neighbors is significant. Thus Chen *et al*. [5] in degree discount heuristic ignored the indirect influence on multi-hop neighbors and discounted degrees of neighbors according to the expected number of adjacent active vertices. Since the number of activated vertices and amount of discount in degrees is small in their algorithm, still it doesn't work well. Our proposed degree-based algorithms improve the spread of influence considering some features of the social network including closeness of vertices with maximum degree and the number of multiple edges.

In social networks, vertices with the maximum degree are adjacent. In other words, when vertex u with the maximum degree is selected, the probability of propagating influence to its neighbors is high. As the number of hops between u and its neighbors increases, the amount of influence on them decreases. Our heuristics take advantage of this matter.

The goal of our algorithm is selecting k vertices with the maximum degree, avoiding selecting vertices with the chance of being influenced by previously selected seeds. To reach this goal we propose tow methods: ignoring the neighbors and descending decrease in neighbors' degrees.

A. Ignoring The Neighbors

In this method in the first step, we select the vertex u with the maximum degree as a seed. Since the reachable neighbor of u will be influenced by it, unlike they will have maximum degrees we remove them from the list of vertices and select next seed with the maximum degree from remained vertices. This process terminates when k seeds are selected. For more description when in each step seed u is selected, its neighbors with h hops are determined with the breadth-first search. Then we remove them from the list of vertices to avoid selecting them in next steps and select next seed from remained vertices. As mentioned before, when the number of hops between u and its reachable neighbor v increase, the spread of influence to v decreases. So we remove reachable

neighbors which there is h hop between them and the seed. h is determined according to the probability of spread of influence considered in the independent cascade model. According to our experiments on different datasets, an appropriate value of h could be the rounded number calculated by equation 1. Therefore, increasing p causes increase in h.

$$h = 12\sqrt{p} \tag{1}$$

B. Descending Degree Decrease

In this method, in the first step, the vertex u with the maximum degree is selected as a seed. In the next step, the degree of its reachable neighbors decreases to reduce their priority. Next seed is selected among vertices with updated degrees. The process continuous until k vertices are selected in the seed set. For each reachable neighbor v, the decrease in its degree is calculated according to the hops that are between v and the selected seed. As the more hops cause fewer activation probabilities, in this method the more hops lead to the less decrease in degree.

In more details for each selected seed u, its degree decrease to constant value α.

$$dec(u) = \alpha$$

where $dec(u)$ is amount of decrease in degree of vertex u, and for each reachable neighbor v its decrease in degree is calculated as equation 2 where c is the number of multiple edges between u and v and $f(p)$ is a function of influence probability p.

$$dec(v) = dec(u) \cdot c \cdot f(p) \tag{2}$$

$f(p)$ is a function of probability p and we consider as follows:

$$f(p) = \beta \cdot p$$

Because the small amount of dec doesn't affect the output significantly. We ignore $dec < 0.1$. So decreasing degrees of reachable neighbors continues until the dec value is more than error value e. According to our experiment, an appropriate value for e is 0.1.

α and β are constant values and our experiments show that values 50 and 10 could be appropriate values for them subsequently.

V. Experiments

In this section, we evaluate experiments of our maximum degree heuristics with some previous works on several real-life datasets. We show that our maximum degree heuristics outperform previous degree based heuristics in terms of the spread of influence in a short time while output the solution close to $(1 - \frac{1}{e})$-approximation algorithms. We also examine the effect of the value of p on the operation of our heuristics.

A. Experiment setup

We evaluate our implementation on tow real-life datasets which are commonly used in related researches including in [5]. First dataset is NetHEPT with $n = 15233$ and $m = 58891$ and second dataset is NetPHY with $n = 37154$ and $m = 231584$.

We compare our algorithms represented by NeighborsRemove and DegreeDecrease with four algorithms named SingleDiscount, DegreeDiscount [5], TIM [7] and IMM [8] that are available by their authors. Our algorithms are implemented in C programming language and compiled with gcc 6.2.1 and are run on a system with 3.60×4 GHz Core i7-3820 Intel and 32G memory.

We use independent cascade model for calculating the spread of influence in our experiments considering probability $p = 0.01$ and $p = 0.1$.

B. Experiment results

Figure 1 and 2 show the runtime of different algorithms under independent cascade model for $p = 0.01$ on NetHEPT and NetPHY subsequently. We see that degree-centrality heuristics are very faster than TIM and IMM. Run-time of DegreeDecrease algorithm is close to DegreeDiscount and SingleDiscount while NeighborsRemove run 10% faster than all. Figure 3 and 4 show running time of all algorithms for $p = 0.1$ and confirm good efficiency of our algorithms for more value of p.

Figure 5 and 6 report the spread of influence of different algorithms under independent cascade model with $p = 0.01$. As it can be seen, although the good performance of our algorithms happens in larger values of p, they work well even for $p = 0.01$ and return close solution to the solution of TIM and IMM.

Figure 7 and 8 show the influence spread of algorithms under ICM model with $p = 0.1$. It is clear that our maximum degree based heuristics outperform DegreeDiscount significantly almost 800 improvements in influence spread on NetPHY. The reason for this remarkable improvement is that increase in the value of p causes more efficiency in our algorithms. When p has a higher value, spreading the influence increases and our strategy is to avoid selecting vertices with the high probability of being influenced. So the selected seed set will be an appropriate solution. Generally, the efficiency of our algorithms is more explicit when considering high influence spread probability in independent cascade model. Also, without any complicated computation, the influence spread of NeighboersRemove and DegreeDecrease is near to TIM and IMM. Overall good performance of our algorithms is represented clearly.

Figure 9 show influence spread of our algorithms under independent cascade model for different values of p. As it is seen, influence spread of our algorithms significantly increases proportionally with the increase of probability value p in the independent cascade model.

VI. Conclusion

In this paper, we propose maximum degree based heuristics considering the closeness of maximum-degree vertices for influence maximization problem under independent cascade model. Experiments show that our heuristics outperform

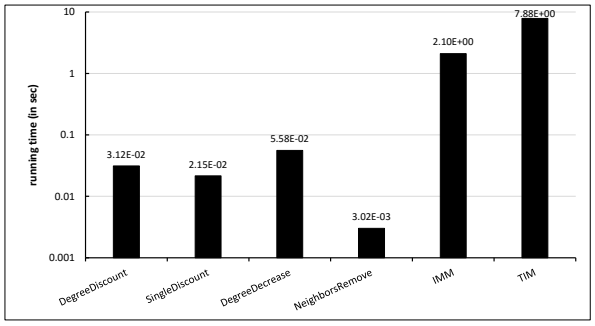

Fig. 1. running time of algorithms on NetHEPT under independent cascade model ($p = 0.01$, $k = 50$).

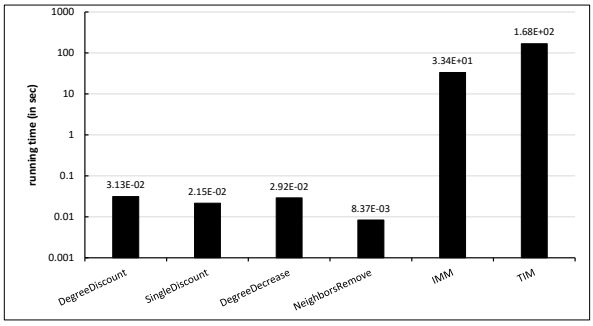

Fig. 3. running time of algorithms on NetHEPT under independent cascade model ($p = 0.1$, $k = 50$).

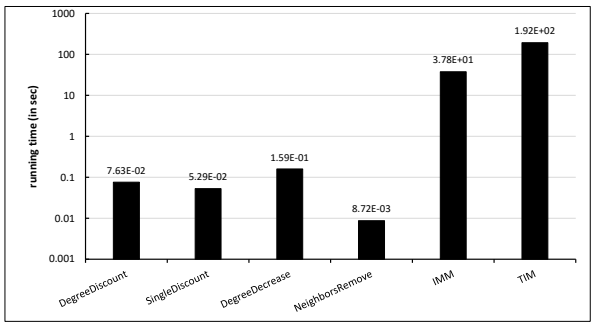

Fig. 2. running time of algorithms on NetPHY under independent cascade model for ($p = 0.01$, $k = 50$).

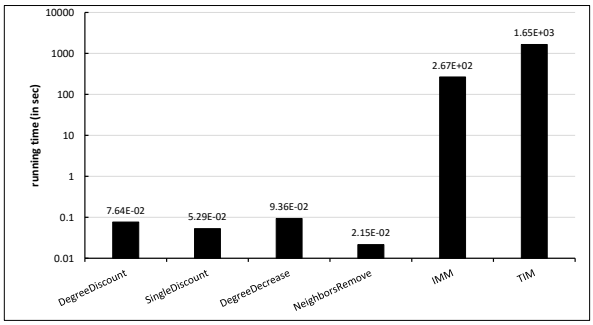

Fig. 4. running time of algorithms on NetPHY under independent cascade model for ($p = 0.1$, $k = 50$).

previous degree-centrality heuristics in terms of the spread of influence in the network which are close to outputs of algorithms that guarantee the solution approximately. Also, they run in a short time. Since algorithms that guarantee the accuracy of the outputs are very time-consuming in large-scale networks, proposing heuristics which have fast speed could be an efficient solution. In future works, we will examine maximum-degree based heuristics for other cascade models. Also, we will study more accurate strategies to improve the spread of influence in fast speed.

REFERENCES

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 57–66.

[2] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 61–70.

[3] D. Kempe, 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*. ACM, 2003, pp. 137–146.

[4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 420–429.

[5] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 199–208.

[6] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2014, pp. 946–957.

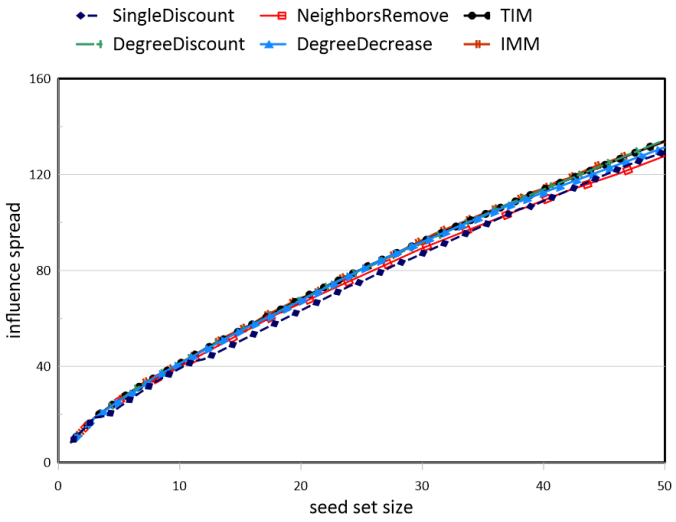

Fig. 5. Influence spreads on NetHEPT under independent cascade model with $p = 0.01$.

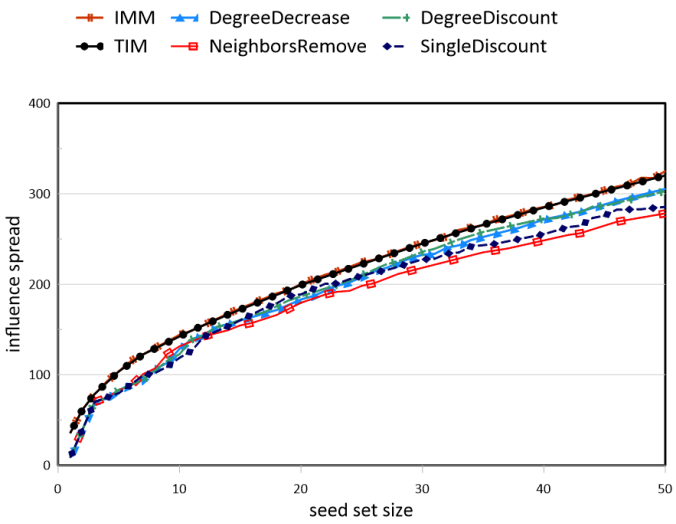

Fig. 6. Influence spreads on NetPHY under independent cascade model with $p = 0.01$.

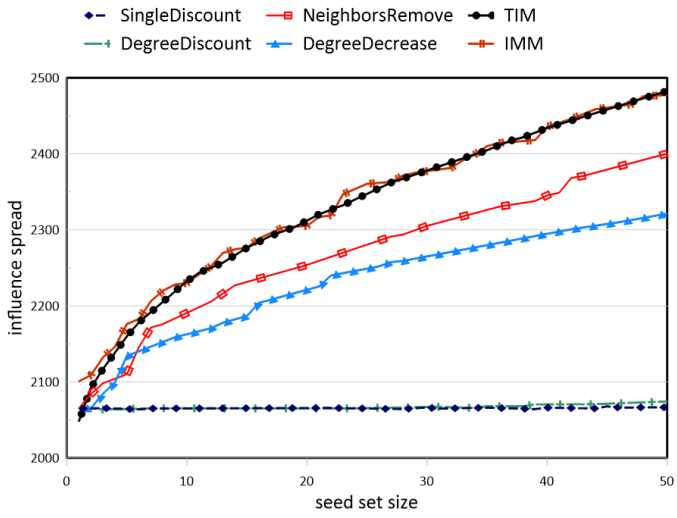

Fig. 7. Influence spreads on NetHEPT under independent cascade model with $p = 0.1$.

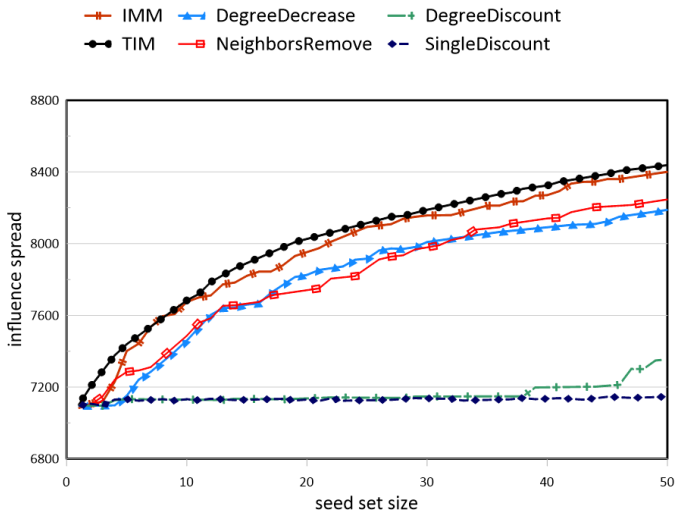

Fig. 8. Influence spreads on NetPHY under independent cascade model with $p = 0.1$.

[7] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 75–86.

[8] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1539–1554.

[9] D. Bucur and G. Iacca, "Influence maximization in social networks with genetic algorithms," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 379–392.

[10] P. Krömer and J. Nowaková, "Guided genetic algorithm for the influence maximization problem," in *International Computing and Combinatorics Conference*. Springer, 2017, pp. 630–641.

[11] M. Weskida and R. Michalski, "Evolutionary algorithm for seed selection in social influence process," in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 1189–1196.

[12] Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, and S.-Y. Lee, "Cim: Community-based influence maximization in social networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 2, p. 25, 2014.

[13] B. Manaskasemsak, N. Dejkajonwuth, and A. Rungsawang, "Community centrality-based greedy approach for identifying top-k influencers in social networks," in *International Conference on Context-Aware Systems and Applications*. Springer, 2015, pp. 141–150.

[14] G. Song, X. Zhou, Y. Wang, and K. Xie, "Influence maximization on large-scale mobile social network: a divide-and-conquer method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1379–1392, 2015.

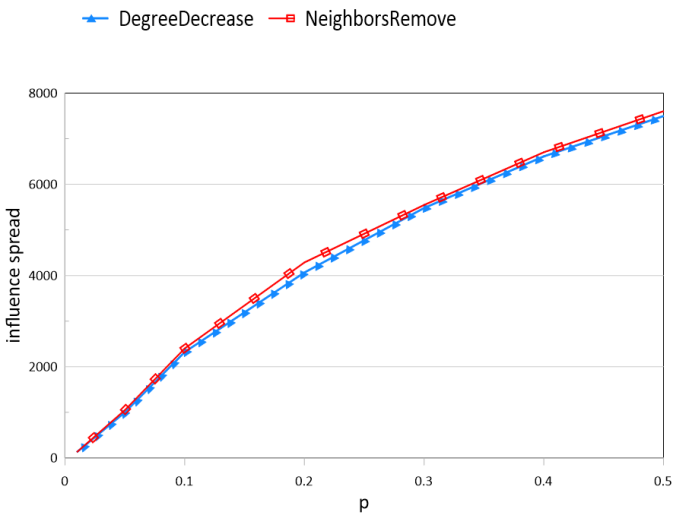

Fig. 9. Influence spreads on NetHEPT under independent cascade model for different values of p