# Targeted Misinformation Blocking on Online Social Networks

Canh V. Pham[1,2], Quat V. Phu[2], and Huan X. Hoang[1]

[1] University of Engineering and Technology
Vietnam National University, Hanoi, Vietnam
Email: canhpv.vnu@gmail.com, huanhx@vnu.edu.vn
[2] Faculty of Information Technology and Security
People's Security Academy, Hanoi, Vietnam
Email: maicanhki@gmail.com, quatphu97mdbg@gmail.com

**Abstract.** In this paper, we investigate a problem of finding smallest set of nodes to remove from a social network so that influence reduction of misinformation sources at least given threshold $\gamma$, called Targeted Misinformation Blocking (TMB) problem. We prove that TBM is #P-hard under LT model. For any parameter $\epsilon \in (0, \gamma)$, we designed Greedy algorithm which return the solution $A$ with the expected influence reduction greater than $\gamma - \epsilon$, and the size of $A$ is within factor $1 + \ln(\gamma/\epsilon)$ of the optimal size. To speed-up Greedy algorithm, we designed an efficient heuristic algorithm, called STBM algorithm. Experiments were conducted on real-world networks which showed the effectiveness of proposed algorithms in term of both effectiveness and efficiency.

**Keywords:** Misinformation, Information diffusion, Social Network, Approximation algorithm

## 1 Introduction

Besides disseminating official information, Online Social Networks (OSNs) are channels in which also allow spreading misinformation and rumors. In order for social networks as a channel of reliable information for users, There should be strategies to again misinformation. Diffusion propagation models are the bases for studying on and identification source of misinformation and restriction the spreading misinformation, in which there are two most common models, *Linear Threshold (LT)* and *Independent Cascade (IC)* models [13]. Base on that, some authors proposed a mathematical approach to detect misinformation or information sources in the case we known the set of nodes were infected by misinformation [1, 2]. Recently, there have been various approaches to decontaminate misinformation by choosing a set nodes to initialize good information and spread it on the same network to convince other users recently [4, 3].

In order to block spreading of misinformation on OSNs, an effective solution is to remove the important nodes from networks [5, 6]. Some authors proposed place monitor or immunization vaccines strategies on some nodes to limit the spread of given misinformation/epidemic sources [2, 7–10]. Placing monitor or vaccination on a node is equivalent to removing this node from the network during propagation process. Zhang et al. formulated the problem of placing monitor at a set nodes so that information spreading from known sources of misinformation to protected central nodes no greater than the protection threshold [2]. Zhang et al. [8] have developed vaccination strategies for nodes that limit the spread of disease on social networks on the IC model. The seminar methods have also been applied for set of edges and nodes to control propagation at groups under LT model [9]. Later on, Song et al. [10, 11] study the problem of limiting misinformation combining time delay on a various of IC model. They also designed heuristic algorithms that outperform the previous algorithms. However, it is difficult to collect data to establish parameters in their models.

Although previous works considered strategies to limit the spreading misinformation, but they do not consider the targeted value for preventing misinformation, i.e, the number of healthy nodes is greater than a given threshold. In reality, to make sure the OSNs are reliable, In reality, to ensure the reliability of information on OSNs, we need to limit the spread of misinformation so that the number of users not infected by misinformation is greater than a given threshold. Motivated by the phenomenon, in this paper, we investigated the *Targeted Misinformation Blocking* (TMB) problem, in which aim to find the smallest set nodes to remove from the network so that the influence reduce from known misinformation sources at least given threshold $\gamma$ under LT model. For the complexity, we proved that TMB problem is #P-hard. We proposed a Greedy algorithm which provided a ratio of $1 + \ln(\gamma/\epsilon)$. We further proposed an efficient heuristic algorithm called STMB which is scalable algorithm for TMB on large-scale networks. Experiments were performed on real-world social traces of NetS, AS and NetHEPT datasets show the performance of our proposed algorithms. In each of the network, we observe that STMB is outperform to the other algorithms in terms of minimizing the size of selected nodes while the runtime is faster.

**Outline of the paper.** The rest of the paper is organized as follows. We first introduce propagation models, problem definition in Sect. 2. We prove the hardness and complexity in Sect. 3. Sect. 4 presents our proposed algorithms. The Experimental results on several datasets are in Sect. 5. Finally, we give some tasks for future work and conclusion in Sect. 6.

## 2 Model and problem definition

First, we introduce Linear Threshold (LT) (see [13]). Based on this, we then formal statement of targeted misinformation blocking problem.

### 2.1 Diffusion Model

Let $G = (V, E, w)$ is a directed graph represents a social network with a node set $V$ and a directed edge $E$, $|V| = n$ and $|E| = m$. Let $N_-(v)$ and $N_+(v)$

are the set of in-neighbors and out-neighbor of node $v$, respectively. Each directed edge $(u, v) \in E$ is associated with an influence weight $w(u, v) \in [0, 1]$ such that $\sum_{u \in N_-(v)} w(u, v) \leq 1$. Given a subset $S \in V, S = \{s_1, s_2, ..., s_k\}$ represents the misinformation sources (as the *seed set* in IM problem [13]). In LT model, each node $v \in V$ has two possible states, *active* and *inactive* and the influence cascades in $G$ as follow. First, every node $v \in V$ uniformly chooses a threshold $\theta_v \in [0, 1]$, which represents the weighted fraction of $u$'s neighbors that must be active to activate $u$. Next the influence propagation happens in round $t = 1, 2, 3...$ At round 1, we activate nodes in set $S$, and set all other nodes inactive. At round $t \geq 1$, an inactive node $v$ is activated if weighted number of its activated neighbors are greater than or equal its threshold, i.e., $\sum_{\texttt{in activated neighbors u}} w(u, v) \geq \theta_v$. Once a node becomes activated, it remains activated in the process of spreading. The influence propagation ends when no more nodes can be activated.

### 2.2 Problem definition

Denote $\sigma_S(G)$ is the influence spread of $S$ in $G$ under LT model, i.e, expected number nodes given activated by $S$. Kempe et al. [13] show that LT model to be equivalent to *live-edge* graph which is constructed by the rules are: (1) for every $v \in V$, select at most one of its incoming edges at random, such that the edge $(u, v)$ is selected with probability $w(u, v)$, (2) and no edge is selected with probability $1 - \sum_{u \in N_(v)} w(u, v)$. The selected edges are called *live* and all other edges are called *blocked*. By claim 2.1 in [13], we have: $\sigma_S(G) = \sum_{g \in \mathcal{G}} \Pr[g] R(g, S)$, where $\mathcal{G}$ is set of sample graphs generated from $G$ according *live-edge* model with a probability denoted by $\Pr[g]$ and $R(g, S)$ denotes the set of nodes reachable from $S$ in $g$. The influence spread from $S$ when remove $A$ is the influence spread of $S$ in induce graph $G[V \setminus A]$, denoted by $\sigma_S(G \setminus A)$. We aim to removing $A$ to maximum the *influence reduction* from $S$ defined as, $h_G(A) = \sigma_S(G) - \sigma_S(G \setminus A)$. For convenience, we simplify the symbol $h_G(.)$ by $h(.)$ due to $G$ is constant. In this paper, we consider *Targeted Misinformation Blocking* (TMB) which is defined as follows:

**Definition 1 (TBM).** *Let $G = (V, E, w)$ is a directed graph represents a social network. Given a set of misinformation source $S = \{s_1, s_2, \ldots, s_k\}, S \in V$ and integer number $\gamma \leq |V|$, find a set $A \subset V \setminus S$ of the smallest size nodes to remove form $G$ such that the expected influence reduction, $h(A)$ at least $\gamma$.*

## 3 Complexity

In this section, we show that TMB problem is #P-hard. Note that a #P problem is at least as hard as the corresponding NP problem.

**Theorem 1.** TMB *problem is #P-hard in LT model*

*Proof.* To proved TMB is #P-hard, we reduce from *s*-*t* paths which is known #P-hard defined as follow:

**Definition 2** **($s$-$t$ paths problem [16]).** *Given a directed graph $G = (V, E), |V| = n, |E| = m$, $s$-$t$ paths problem ask to compute the number of (directed) paths from node $s$ to node $t$ that visit every node at most once.*

Consider an instance $\mathcal{I}_1$ of $s$-$t$ paths problem, where $G = (V, E)$, $s, t \in V$ are given. As Fig. 1 shows, from $G$, we construct $G'$ as follow we add a new node $u$ and add two edges $(s, u), (t, u)$ with weights $w(s, u) = w(t, u) = 1/2$. We add more set $Q$ include $2n$ nodes and connect $u$ to them with the same weight is equal to 1. For the others edges, we set the weight is equal to $w = 1/\Delta$, where $\Delta$ be the maximum in-degree of any node in $G$. This assumption still satisfies the LT mode since the total of in-neighbour weight is not greater than 1.

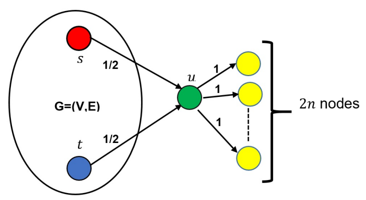


**Fig. 1.** Reduce from $s$-$t$ paths to $\mathsf{TMB}$.

By claim 2.6 of [13], we have: $\sigma_S(G') = \sum_{x \in \mathsf{P}(G',s)} \prod_{e \in x} w(e)$, and $\sigma_S(G' \backslash \{u\}) = \sum_{x \in \mathsf{P}(G' \backslash \{u\},s)} \prod_{e \in x} w(e)$. Eliminate the same elements in the two above equations so the remaining paths containing node $u$. Set these paths divided into two groups: paths have $u$ is the endpoint and the paths have $v \in Q$ is endpoint. Therefore, $h(u) = \sigma_S(G') - \sigma_S(G' \setminus \{u\}) = \sum_{x \in \mathsf{P}(G',s,u)} \prod_{e \in x} w(e) + \sum_{v \in Q} \left( \sum_{x \in \mathsf{P}(G',s,v)} \prod_{e \in x} w(e) \right) = \frac{2n+1}{2} \sum_{i=0}^{n-1} \alpha_i w^i + n$. Where $\alpha_i = |\mathsf{P}_i(G, s, t)|$. Let $f(w) = \sum_{i=0}^{n-1} \alpha_i w^i$, on $G'$ we easy see that $0 \le f(w) \le 1$ $n \le h(u) \le 2n + \frac{1}{2}$, and $h(u) = \max_{v \in G'} h(v)$. We first show that if we can determine $f(u) \ge \beta$ for any integer $\beta \in [0, 1]$ in polynomial time, we can solve $s$-$t$ paths problem in polynomial time. Since the weigh $w = 1/\Delta$, $f(w)$ is a fraction with a numerator of $\Delta^{n-1}$ and the numerator at most $\Delta^{n-1}$. By using binary search from 1 to $\Delta^{n-1}$, we can find value of $f(w)$. This task can be done in $\mathcal{O}(\log(\Delta^{n-1})) = \mathcal{O}((n-1)\log \Delta) = \mathcal{O}(n \log n)$. Hence, we can calculate $f(u)$ in polynomial time. We then the adjust weight $w$ to $n$ distinction values $\frac{1}{\Delta}, \frac{1}{\Delta+1}, \ldots, \frac{1}{\Delta+n-1}$. By using above method, we can find value of $f(w)$ corresponding to each $w$. Hence, we obtain a set of $n$ linear equations $\sum_{i=0}^{n-1} \alpha_i w^i = f(w), w \in \{\frac{1}{\Delta}, \frac{1}{\Delta+1}, \ldots, \frac{1}{\Delta+n-1}\}$ with $\{\alpha_0, \alpha_2, \ldots, \alpha_{n-1}\}$ as variables. The matrix of this equation is $M_{n \times n} = \{m_{ij}\}$ and $m_{ij} = w^i, i, j = 0, .., n$ so this is Vandermonde matrix and we can easily to compute the unique solution $\{\alpha_0, \alpha_2, \ldots, \alpha_{n-1}\}$ for the linear system of equations. The total of $s$-$t$ paths in $G$ is $\sum_{i=0}^{n-1} \alpha_i$. Therefore, we can solve $s$-$t$ paths problem in polynomial time.
We now consider an instance $\mathcal{I}_2$ of $\mathsf{TMB}$ where $S = \{s\}$, $\gamma = \beta \frac{2n+1}{2} + n, \beta \in [0, 1]$.

Assume that an $\mathcal{A}$ is a polynomial-time algorithm solving TMB problem. Consider two cases: (1) If $\mathcal{A}$ returns the solution set $A$ whose size is equal to 1, we only need to select $A = \{u\}$, infer $f(w) \geq \beta$. (2) If $\mathcal{A}$ returns the solution set $A$ whose size larger than 1. At that besides $u$, some nodes are chosen into $A$. We infer $f(w) < \beta$. Therefore, $\mathcal{A}$ can be used to decide $f(w)$ is greater than $\beta$, that can also solve the $s$-$t$ paths problem. This implying that our TMB problem is at least as hard as $s$-$t$ paths problem. □

# 4 Proposed algorithms

## 4.1 Greedy algorithm

We introduce an approximation algorithm that provide a ratio of $1 + \ln(\gamma/\epsilon)$ base on $h(.)$ is proved *submodular* and *monotone* function, i.e, for $A \subset T, v \notin T$ $h(A + \{v\}) - h(A) \geq h(T + \{v\}) - h(T)$

**Algorithm 1:** Greedy Algorithm (GA)

**Data:** Graph $G = (V, E, w)$, $\mathcal{S} = \{S_1, S_2, .., S_q\}$, threshold $< \gamma < |V|$, parameter $\epsilon \in (0, \gamma)$

**Result:** set of nodes $A$

```
A ← ∅;
while h(A) > γ − ε do
    u = arg max_{v∈V\A} δ(A, v); A ← A ∪ {u};
end
return A;
```

**Theorem 2.** *The function $h(.)$ is submodular and monotone function.*

*Proof.* Denote $N_E(A)$ is set of edges adjacent with all nodes in $A$. By theorem 5 in [12], for $A \subseteq T$ we have $h(T)-h(A) = \sigma_S(N_E(A))-\sigma_S(N_E(T)) \geq 0$. Therefore $h(.)$ is a monotonically increasing. We then show that the function $\sigma_S(G_i \setminus A)$ is a supermodular function of the set $A$ is the variable, i.e, $\forall A \subseteq T \subset V$, $\forall v \in T \setminus A$, we have $\sigma_S(G \setminus (A \cup \{v\})) - \sigma_S(G \setminus A) \leq \sigma_S(G \setminus (T \cup \{v\})) - \sigma_S(G \setminus T)$ Let $E_{T,v} = N_E(T+\{v\}) \setminus N_E(T), E_{A,v} = N_E(A+\{v\}) \setminus N_E(A)$ we have $E_{T,v} \subseteq E_{A,v}$ and due to $A \subseteq T$. We obtain $N_E(A) \cup E_{T,v} \subseteq N_E(A + \{v\})$. Let $\sigma_S(G \setminus X)$ is the influence of $S$ for graph $G$ after remove the set edges $X \subset E$, we obtain $\sigma_S(G \setminus A) = \sigma_S(G \setminus N_E(A)$. By theorem theorem 6 in [12], $\forall X \subseteq Y, e \in Y \setminus X$, we have:

$$\sigma_S(G_i \setminus (X \cup \{e\}) - \sigma_S(G_i \setminus X) \leq \sigma_S(G \setminus (Y \cup \{e\})) - \sigma_S(G_i \setminus Y) \quad (1)$$

Therefore, $\sigma_S(G\setminus A)-\sigma_S(G\setminus(A\cup\{u\})) = \sigma_S(G\setminus N_E(A))-\sigma_S(G\setminus N_E(A+\{v\})) \geq \sigma_S(G \setminus N_E(A)) - \sigma_S(G \setminus (N_E(A) \cup E_{T,v})) \geq \sigma_S(G \setminus N_E(T)) - \sigma_S(G \setminus (N_E(T) \cup$

$E_{T,v})) = \sigma_S(G \setminus T) - \sigma_S(G \setminus (T \cup \{u\}))$(Apply the inequality (1)). Combine with $h(A) = \sigma_S(G) - \sigma_S(G \setminus A)$ we easy see that $h(.)$ is a supermodular function □

.

**Theorem 3.** *Algorithm 1 return solution A satisfies* $h(A) \geq \gamma - \epsilon$*, and the size of A is within factor* $1 + \ln(\gamma/\epsilon)$ *of the optimal size*

The proof of theorem 3 straightforward based on [15]. Base on theorem 2, the greedy algorithm given in Algorithm 1 achieve $1 + \ln(\gamma/\epsilon)$ approximation ratio. The algorithm simply choses the node that provides maximum largest *incremental influence reduction* in each step, defined as $\delta(A, u) = \min\{\gamma, h(A + \{u\})\} - h(A)$. The main challenge of this algorithm comes from calculate $\sigma_S(.)$ is #P-hard (see [14]). Therefore, we introduce an efficiency algorithm in next subsection.

### 4.2 Scalable TMB Algorithm

We try to tackle this problem with a speed-up approach proposed by Zhang [9]. This approach use characteristics of the LT model, in which the set nodes that reach from a seed node $v$ in live-edge is a tree root at $v$. In our proposed algorithm, we first simplify the instance of TMB problem by merging set source $S = \{s_1, s_2, .., s_k\}$ into a *supper source node I*. For each node $v \in N_+(S)$, we assign weight $w(I, v) = \sum_{s \in N_-(v) \cup S} w(s_i, v)$ and remove $S$ after update the new weight set, the result's called *merged graph* $G'$. Based on the characteristic of LT model, the instances before and after of TMB are equivalence (see more details in [8, 9]). Next, we'll generate $\eta$ sample graphs $g$ from the $G'$. For each $g$, we construct an induced tree root at $I$ by removing the edges $(v, I), \forall v \in g$. We obtained set $\mathcal{L}$ which contains $\eta$ tree (line 3). The influence reduction of a node $v$ on each tree is calculated by using DFS algorithm. We then approximate the marginal influence reduction of node $u$ on $G$ is equal to average influence reduction of node $u$ on all tree $T_I \in \mathcal{L}$ (line 4).

After that, we apply the lazy forward method in [17] to select the solution based on $h(.)$ is submodular function (line 10-23). The node is selected in each step also removed from each tree $T_I \in \mathcal{L}$ and $h(u, T_I)u \in T_I$ will be updated (line 18) in the way as follows: (1) For children of $u$, we can remove them because it is not reachable from $I$, (2) for any ancestor $v$ of $u$, $h(v, T_I \setminus u) = h(v, T_I) - h(u, T_I)$, which can be done in constant time. The details of algorithm are presented in algorithm 2

**Complexity**. Merge algorithm takes $\mathcal{O}(k + |N_+(S)|)$ (line 3). Generating $\eta$ sample takes $\mathcal{O}(\eta(m + n))$. Calculating $h(T_I, u), \forall u \in T_I$ can be done in $\mathcal{O}(\eta n)$. For lazy forward phase, the total time needed takes at most $\mathcal{O}(q\eta n)$ where $q$ is the number of iterations of while loop. Therefore, algorithm 2 runs in $\mathcal{O}(\eta(m + qn))$.

**Algorithm 2:** Scalable TMB (STMB) Algorithm

**Data:** Graph $G = (V, E, w)$, $\mathcal{S} = \{s_1, s_2, .., s_q\}$, threshold $\gamma > 0$
**Result:** set of nodes $A$
$A \leftarrow \emptyset$; $(G', I) \leftarrow \texttt{Merge}(G, \mathcal{S})$.
Remove all node, $I$ can't reach in $G$.
Generate $\eta$ live-edge graphs and set $\eta$ tree $\mathcal{L} = \{T_I^1, T_I^2, \ldots, T_I^{|\eta|}\}$
For each $T_I \in \mathcal{L}$, calculate $r(u, T_I)$ for all $u \in T_I$ (by using DFS algorithm).
**for** $u \in V$ **do**
  $u.\delta(u) \leftarrow \frac{1}{\eta} \sum_{T_I \in \mathcal{L}} r(u, T_I)$; $u.cur \leftarrow 1$
  Insert element $u$ into $Q$ with $u.\delta(u)$ as the key
**end**
$h_{max} \leftarrow 0$; $iteration \leftarrow 1$
**while** $h_{max} < \gamma - \epsilon$ **do**
  $u_{max} \leftarrow dequence\ Q$
  **if** $u_{max}.cur = iteration$ **then**
    $A \leftarrow A \cup \{u_{max}\}$
    $iteration \leftarrow iteration + 1$
    **foreach** $T_I \in \mathcal{L}_c$ **do**
      If $u_{max} \in T_I$, remove node $u_{max}$ and update $r(v, T_I)$, $\forall v \in T_I$.
    **end**
    $h_{max} \leftarrow h_{max} + u_{max}.\delta(u_{max})$
  **else**
    $u_{max}.\delta(u_{max}) \leftarrow \frac{1}{\eta}\big(\sum_{T_I \in \mathcal{L}} r(I, T_I) - \sum_{T_I \in \mathcal{L}} r(I, T_I \setminus u_{max})\big)$
    $u_{max}.cur = iteration$; re-insert $u_{max}$ into $Q$
  **end**
**end**
**return** $A$;

## 5 Experiments

In this section, we show experimental results of proposed algorithms on three real-world datasets to evaluate the performance and compare them with several other baselines algorithm.

### 5.1 Experiment setup

*Dataset.* The three real-world networks we use and their basic statistics are summarized in Table 1. We assign the weights of edges in LT model according to previous studies [12–14]. The weight of the edge $(u, v)$ is $w(u, v) = \frac{1}{d_-(v)}$, where $d_-(v) = |N_-(v)|$. For the misinformation source, we randomly choose $S$ in 4-6% of the set nodes. The code is written in Python 2.7 using the NetworkX library and all experiments are run on a Linux Server machine with 2.30 GHz Intel® Xeon® CPU E5-2697 and 128G of RAM DDR4.
*Algorithms Compared.* In our experiments, we compare STMB algorithm with other algorithms listed below:

**Table 1.** Datasets

| **Dataset** | NetS [18] | AS [19] | NetHEPT [13, 14] |
|---|---|---|---|
| **Num. of nodes** | 1.5K | 6.4K | 15.2K |
| **Num. of edges** | 5.4K | 12.5K | 32.2K |
| **Avg. degree** | 3.8 | 7.5 | 4.2 |
| **Num. of source nodes** | 100 | 300 | 1000 |

-PageRank: Compute a ranking of the nodes in the graph $G$ based on the structure of the incoming links. It was originally designed as an algorithm to rank web pages. We setup damping parameter for PageRank is 0.85. Because $h(.)$ is monotonic function, we used binary search algorithm to find A set with $|A|$ nodes having highest-ranked.
-High − Degree: A heuristic based on the notion of degree centrality. We sort all nodes base on degree of each node then making the same to PageRank, we use binary search algorithm to find $A$.
-Greedy: The Greedy algorithm (algorithm 1) with the lazy evaluation optimization in [17].
We run $10,000$ simulations to accurately estimate $h(A)$ for every $A$ set obtained for each algorithm.

### 5.2 Experiment Results

**Solution quality.** As demonstrated in the Fig. 2, the number of selected nodes gave by STMB algorithm is the smallest. STMB is up to 39% better than Greedy method, 60%-95% and 57%-87% better than that PageRank and High − Degree respectively. To check $A$ set got from STMB algorithm, we run 10000 times Monte-Carlo simulations to calculate function $h(A)$ and result is shown in Fig.3. In most cases $h(A)$ is greater than $\gamma$.
**Running Time.** The running time of different algorithms on the three networks are given in Fig.2 and Table 2. On the NetS and NetHEPT dataset, our STMB algorithm is roughly two times faster than the PageRank, High − Degree and 800-3500 times faster than the Greedy. On the AS dataset, STMB algorithm is slower than the PageRank and High Degree but still 300 times faster than Greedy. From the result, we see that STMB algorithm is very competitive in its time efficiency.

## 6 Conclusions

In this paper, we studied the TBM problem, in which aim to finding smallest set nodes to remove from a social network so that the number of influence reduction no less than a given threshold $\gamma$. Besides proving the problem is #NP-Hard. We proposed two algorithm: Greedy and STMB algorithms. In the future, we will tackle the TBM problem in other diffusion model, especially IC model.

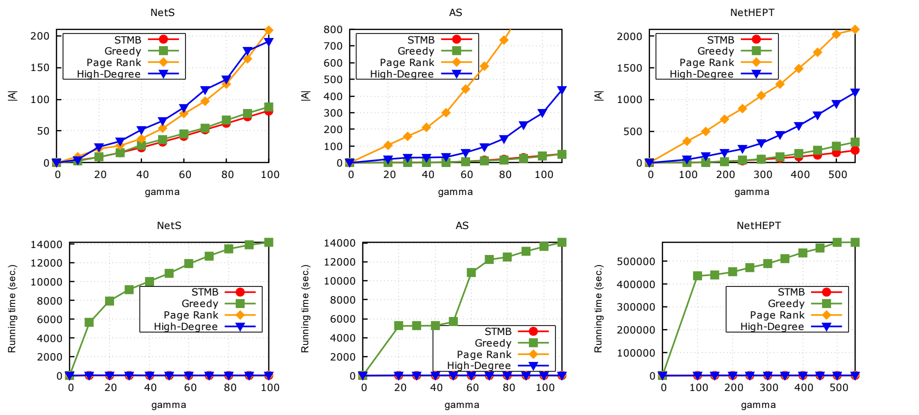


**Fig. 2.** Comparison of Solution quality of algorithms on NetS, AS, NetHEPT networks

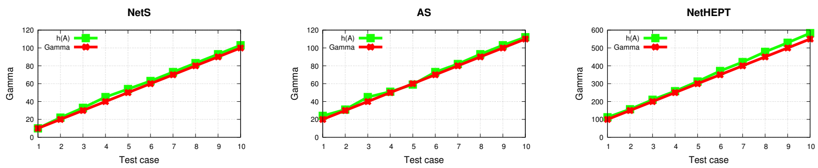


**Fig. 3.** Check result of STBM on TBM problem.

## References


1. D. T. Nguyen, N. P. Nguyen, and M. T. Thai.: Sources of Misinformation in Online Social Networks: Who to Suspect?. In: Proceedings of the IEEE Military Communications Conference (MILCOM), pp.1–6, IEEE(2012)
2. H. Zhang, A. Kuhnle, H. Zhang, and M. T. Thai.: Detecting misinformation in online social networks before it is too late. In: Proceedings of 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 541–548, IEEE(2016)
3. Nam P. Nguyen, Guanhua Yan, My T. Thai.: Analysis of misinformation containment in online social networks. Computer Networks 57, 2133–2146 (2013)
4. Ceren Budak, Divyakant Agrawal, and Amr El Abbadi.: Limiting the spread of misinformation in social networks. In: Proceedings of the 20th international conference on World wide web. pp.665–674, ACM(2011)
5. H. Tong, B. A. Prakash, C. Tsourakakis, T. Eliassi–Rad, C. Faloutsos, and D. H. Chau.: On the vulnerability of large graphs. In: Proceedings of IEEE International Conference on Data Mining (ICDM), pp. 1091–1096, IEEE (2010)
6. Prakash BA, Tong H, Valler N, Faloutsos M, Faloutsos C (2010).: Virus propagation on time–varying networks: Theory and immunization algorithms. In: Proceedings of 2010 Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 99–114, Springer (2010)

**Table 2.** Compare running time between algorithms

| **Dataset** | Heuristic | Greedy | Page Rank | High-Degree |
|---|---|---|---|---|
| NetS | **17.57** | 14206.80 | 35.73 | 30.24 |
| AS | 45.70 | 14074.87 | **14.39** | 17.85 |
| NetHEPT | **165.12** | 582566.74 | 392.34 | 374.66 |


7. H. Zhang, M. Alim, X. Li, M. T. Thai and H. Nguyen.: Misinformation in Online Social Networks: Catch Them All with Limited Budget, ACM Trans. Inf. Syst. (TOIS), 34 (3), 24 pages (2016)
8. Y. Zhang and B. Prakash. Dava.: Data-Aware Vaccine Allocation over Large Networks. ACM Trans. Knowl. Discov, 10(2), 32 pages (2015)
9. Y. Zhang, A. Adiga, S. Saha, A. Vullikanti, and B. A. Prakash.: Nearoptimal lgorithms for controlling propagation at group scale on networks. IEEE Trans. Knowl. Data Eng, 28(12), 3339–3352 (2016)
10. Chonggang Song, Wynne Hsu, and Mong Li Lee.: Node immunization over infectious period. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 831-840, ACM(2015)
11. Chonggang Song, Wynne Hsu, and Mong Li Lee.: Temporal influence blocking.: Minimizing the effect of misinformation in social networks. In proceedings of IEEE 33rd International Conference on Data Engineering , pp. 847-858, IEEE(2017)
12. Elias Boutros Khalil, Bistra Dilkina, Le Song.: Scalable Diffusion–Aware Optimization of Network Topology. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1226–1235, ACM (2014).
13. D. Kempe, J. Kleinberg, and E. 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, pp. 137-146, ACM(2003)
14. W. Chen, C. Wang, and Y. Wang.: Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In: Proceedings of the 2010 IEEE International Conference on Data Mining, pp. 88-97, IEEE(2010)
15. A. Goyal, F. Bonchi, L. V. S. Lakshmanan, and S. Venkatasubramanian.: On minimizing budget and time in influence propagation over social networks. Social Network Analysis and Mining, 2(1), 179–192 (2012)
16. L. G. Valiant.: The complexity of enumeration and reliability problems. SIAM Journal on Computing 8(3), 410–421 (1979)
17. J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. Van–Briesen, and N. S. Glance.: Cost–effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 420-429, ACM(2007)
18. DT Nguyen, H Zhang, S Das, MT Thai, TN Dinh.: Least cost influence in multiplex social networks: Model representation and analysis. In: Proceedings of IEEE 13th International Conference on Data mining, pp. 567–576, IEEE(2013)
19. J. Leskovec, J. Kleinberg and C. Faloutsos. Graphs over Time.: Densification Laws, Shrinking Diameters and Possible Explanations. In: Proceedings of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 177-187, ACM (2005)