A Heuristic for Influence Maximization Under Deterministic Linear Threshold Model

Eszter Csókás, Tamás Vinkó
University of Szeged, Institute of Informatics, Hungary
E-mail: csokas@inf.u-szeged.hu, tvinko@inf.u-szeged.hu

Keywords: influence maximization, centrality metrics, deterministic linear threshold model

Received: April 16, 2023

Influence maximization (IM) is perhaps one of the most actively studied problems in network science. It is a combinatorial optimization problem in which, given a directed social network with influence weights, a spreading model, and a positive integer k, it is required to identify the set of seed nodes of size k which can make the largest influence in the network. We proposed an exact ILP model in our recent work and iterative solution approach to solve the IM problem under the so-called deterministic linear threshold spreading model. Since the solution describes how the diffusion happens for different time constraints, it is of interest to investigate how the various characteristics of the underlying graph relates to the result. In this paper, we present two new centrality metrics, computed from the input network structure, and use them to minimize the number of possible seed nodes. The solver chooses among the potential seed nodes and solves the problem, reducing the solution time. Benchmarking results are shown and discussed to demonstrate the efficiency of the proposed method.

Povzetek: Predlagana sta nova centralnostna kazalnika za optimizacijo izbire začetnih vozlišč v determinističnem linearnem modelu praga, kar izboljša učinkovitost in čas rešitve problema maksimizacije vpliva v omrežjih.

1 Introduction

Influence maximization Influence maximization (IM) is a combinatorial optimization problem. It studies a social network represented as a graph $G = (V, E, W)$, where V is the set of nodes in G, E is the set of directed edges in G and $W : E \to \mathbf{R}_+$ is a non-negative weight function. The goal of the problem is to find a $k \geq 1$ sized set of so-called seed nodes $v_1, \ldots, v_k \in V$ with the maximum influence in graph G in such a way that a weighted directed graph G, a diffusion (or spreading) model and the integer k are given [13].

The following notations will be used: $n = |V|$, for a node $j \in V$ the set of its out-neighbors is denoted by $N_{out}(j)$, and for $j \in V$ the set of in-neighbors is denoted by $N_{in}(j)$. Let $S \subset V$ of size $0 < k \leq n$ be the set of seeds and the function $\sigma(S)$ is the number of influenced nodes started from $S \subset V$ seeds by executing the diffusion model. The formal definition of the optimization problem is therefore

$$\max_{S \subset V, |S|=k} \sigma(S).$$

Diffusion models are usually used with stochastic parameters to solve influence maximization [13]. Thus, $\sigma(S)$ is the expected number of influenced nodes. The nodes with influenced and uninfluenced states will also be called as active and inactive nodes, respectively.

Diffusion models Several relevant diffusion models can be found in the literature, among which the independent cascade model [9], triggering model [13], time-aware model [16] and the linear threshold model [10] are the most popular ones. Our model is based on the linear threshold model (LTM) which solves the problem by iterating over a $t \in \mathbb{N}$ value, starting with $t = 1$. Let $0 < \theta_i < 1$ be the threshold value for each $i \in V$ that determines how easy it is to make the vertex active. The steps of the LTM is shown in Algorithm 1.

Algorithm 1 Linear threshold diffusion model

Step 1 Let $t = 1$, $0 < k \leq n$ be fixed and $\mathcal{V}_0$ be a seed set which containing k nodes, $\mathcal{V}_1 = \emptyset$.

Step 2 If

$$\sum_{j \in \tilde{N}(i)} b_{j,i} \geq \theta_i$$

holds for all $i \in V$ uninfluenced nodes, then put node i into the set $\mathcal{V}_t$. Mark those nodes as active at the end of this step.

Step 3 If there is no chance to influenced more node that is $\mathcal{V}_t = \emptyset$ holds, then STOP. $\mathcal{V} = \mathcal{V}_0 \cup \ldots \cup \mathcal{V}_{t-1}$ and $\sigma(\mathcal{V}_0) = |\mathcal{V}|$ by this time.

Otherwise, let $t := t + 1$ and go back to Step 2.

The value of σ in the LTM is determined by executing R runs using θ_i values uniform at random, then taking the average value of the influence to obtain the expected value of σ.

Deterministic linear threshold model In our work we used the deterministic linear threshold model (DLTM). It differs in that the θ_i values of the nodes are fixed, and thus no need for running the diffusion model multiple times. Note, that in [10] the original LT model is also deterministic. DLTM has been investigated also in the recent years, see e.g. in [1, 17, 12, 31, 19]. One of the key properties of DLTM is that the submodularity property does not hold [2], so there is no guarantee of the efficiency of the greedy algorithm.

Note that the IM under DLTM is a bilevel optimization problem since we need to find the maximum number of active nodes together with the minimum time t. Even using state-of-the-art optimization methods it is not possible to solve this kind of problem. Hence, an iterative solution method needs to be applied using the binary linear program which details are given in the next paragraph.

Integer LP model In the following, we present an integer linear programming (ILP) model, which is the result of our previous work [5]. This served as inspiration, since although it gives the global optimum of the problem, we only obtain results for small graphs within a reasonable run-time. Assuming that $\mathcal{T} > 1$ is a given integer constant and that $T = \{2, \ldots, \mathcal{T}\}$ is the set of time periods describing the diffusion process, the ILP model is as follows.

$$\max \sum_{i=1}^{n} x_{i,\mathcal{T}} \tag{1}$$

$$\sum_{i=1}^{n} x_{i,1} \leq k \tag{2}$$

$$\sum_{j \in N(i)} b_{j,i} x_{j,t-1} \geq \theta_i (x_{i,t} - x_{i,t-1}) \ \ \forall (i \in V, t \in T) \tag{3}$$

$$\sum_{j \in N(i)} b_{j,i} x_{j,t-1} \leq \theta_i + x_{i,t} - \varepsilon \ \ \forall (i \in V, t \in T) \tag{4}$$

$$x_{i,t-1} \leq x_{i,t} \ \ \forall (i \in V, t \in T) \tag{5}$$

$$\sum_{i=1}^{n} x_{i,\mathcal{T}-1} + 1 \leq \sum_{i=1}^{n} x_{i,\mathcal{T}} \tag{6}$$

$$\mathbf{x} \in \{0,1\}^{n \times \mathcal{T}} \tag{7}$$

The objective function (1) maximizes the number of active nodes in the last time epoch $\mathcal{T}$. Constraint (2) limits the number of seed nodes that can be initially selected. A node becomes active at a time t if the sum of the weights of the edges coming from its already active neighbours exceeds the threshold value of the vertex, defined by (3). Furthermore, the difference training on the right-hand side of this ensures that any node can be chosen as a seed at $t = 1$. This is related to the constraint (4) that the node is affected if the sum of the weights of the edges from the active neighbors exceeds the threshold of the vertex at time t. A small $\varepsilon > 0$ is required to ensure that the node is activated even if the sum of the weights of the edges arriving from the active neighbors equals the threshold value of the vertex[1]. The constraint (5) is that the state of the vertices cannot change backwards, i.e., once a node becomes active, it remains to be active. The equation (6) determines that at time $\mathcal{T}$ the number of influenced nodes should be greater than or equal to the number of active nodes at the preceding time in $\mathcal{T}-1$. Finally, constraint (7) describes that the solution matrix $\mathbf{x}$ is binary, i.e., one vertex is either influenced or uninfluenced at $t \in T$.

Centrality in general A centrality measure shows the nodes' importance in a graph which is based on the location of the nodes within the graph. Accurately, if given a graph $G(V, E, W)$, a centrality measure $C : V \to \mathbb{R}_+$ which assigns a non-negative centrality value to every node. The order of the nodes formed by the centrality values is usually more important than the centrality value itself [6, 27]. There are two main categories of centralities. One is based on shortest path, which includes closeness [26] and betweenness [8], among others. The other one is based on neighborhoods, where the centrality metrics, just to mention the most frequently used ones, are degree [28], eigenvector [3] and PageRank [4]. The centrality metric we have created is also based on neighborhoods what we describe below in Section 3.

2 Related work

The influence maximization is still an actively researched area of great interest and thus has a huge scientific literature. In this section, we would like to present the related works that are relevant, i.e., those based on the deterministic linear threshold model.

A fast algorithm for finding the most influential people was proposed in [22]. First, they were looking for communities of the graph and examining their limited number to reduce implementation time. The nodes to be excluded were selected using centrality measurements, community detection and new set computation. They then bounded the search space of the input network using the new set. The main advantage of the algorithm is that it reduces the number of nodes to be tested without compromising quality in order to reduce execution time.

The threshold-based greedy approach was presented in [14] for solving IM under DLTM. The greedy approach allows to obtain very good results compared to other combinatorial algorithms. It can be also used with the genetic algorithm and significantly improve its efficiency.

[1] An important note: we assume that the sum of the weights of the incoming edges is at most 1.

A new hybrid centrality metric based on closeness (harmonic) and decay measures was proposed in [29]. Two main application areas were presented. One hybridization was used to solve the coverage problem, while the other tries to find the most ideal node to reach the most people in the population, using the deterministic threshold model. In both cases, the centrality metric is based on a formulation of the weighted centrality measure of nodes.

The complexity of the IM under DLTM is studied in [18]. It is shown that for DLTM, active nodes are not approximable in $n^{1-\varepsilon}$ -factor polynomial time unless P=NP. In contrast, they are well approximable in the linear threshold model and the independent cascade model. It has also been demonstrated that for a given set of seeds, the number of influenced nodes can be determined in polynomial time.

The fact that the DLTM has no polynomial time $n^{1-\varepsilon}$ approximation unless P=NP, even when a person needs at most two active neighbors to become active was shown in [17]. However, there exists an $\varepsilon/(\varepsilon-1)$ polynomial-time approximation in the case where one of the neighbours has already become active and the person is can be activated. It is shown to be the best approximation under plausible Complexity-Theoretic assumptions.

In [11], the authors have created Targeted and Budgeted Influence Maximization for DLTM. They advanced a scalable algorithm that allows some optional methods to solve the problem, it is the TArgeted and BUdgeted Potential Greedy (TABU-PG) algorithm. It is an iterative and heuristics algorithm that relies on investing in potential future gains when choosing seed nodes.

In [30], the selection of top-k nodes is investigated based on the measure corresponding to the social network under consideration. It relies on the Shapley measure to efficiently compute an approximate solution to the problem. Although explicitly not using DLTM, the algorithm is general in a sense that it does not exploit the submodular property of the function.

3 New centralities

Two new centrality measures are proposed. Both of them are specifically developed for solving IM problem under DLTM. What makes them distinguished from other centralities is that they take into account not only the direction and the weight of the graph edges, but also the weight, i.e. threshold, of the nodes. To make it easier to understand our metrics and calculation, we have made a small graph, which is shown in Figure 1.

3.1 Influenceability

Our first centrality is the influenceability, denoted by $\mathcal{I}_{in}$, which measures how to easy activate a node. To calculate this, we examine the incoming edges from the neighbours, namely which edges and combinations of edges are able to reach or exceed the threshold value of the node. We define the weighted incidence, denoted by w_{to} as follows. Take the number of edge combinations that are able to activate the node by dividing by the number of edges in the edge combination and all occurrences of a given number of edge combinations. Finally, sum the w_{to} values and obtain $\mathcal{I}_{in}^{(p)} = \sum w_{to}(i)$, where $i \in V(G)$.

Table 1 shows the calculation of $\mathcal{I}_{in}^{(p)}$ for vertex 8 of the graph in Figure 1. The in-neighbors of vertex 8 are nodes $9, 4$ and 3. Note that vertex 4 can influence vertex 8 by itself, so combinations where vertex 4 is included will certainly be able to influence vertex 8.

Table 1: The $\mathcal{I}_{in}^{(p)}$ value of node 8 of the graph in Fig. 1.

combinations of edges	sum	θ	$w_{to}(8)$
$9 \to 8$	0.05	0.27	0
$4 \to 8$	0.27	0.27	$1/(1 \cdot 3)$
$3 \to 8$	0.18	0.27	0
$9 \to 8, 4 \to 8$	0.31	0.27	$1/(2 \cdot 3)$
$9 \to 8, 3 \to 8$	0.23	0.27	0
$4 \to 8, 3 \to 8$	0.44	0.27	$1/(2 \cdot 3)$
$9 \to 8, 4 \to 8, 3 \to 8$	0.49	0.27	$1/(3 \cdot 1)$
			$\mathcal{I}_{in}^{(p)}(8) = 1$

Table 2 shows the calculated $\mathcal{I}_{in}^{(p)}$ values for the vertices of the small graph in Fig. 1.

Table 2: The $\mathcal{I}_{in}^{(p)}$ values for the graph in Fig. 1.

node:	1	2	3	4	5	6	7	8	9
$\mathcal{I}_{in}^{(p)}$:	0	1	1	0	1	1.5	1.5	1	0

The final centrality metrics are obtained by combining with the measure of node and its neighbors. The influenceability value of a node is obtained by adding to the value of $\mathcal{I}_{in}^{(p)}$ the approximation of the influenceability of its in-neighbours:

$$\mathcal{I}_{in}(i) = \mathcal{I}_{in}^{(p)}(i) + \sum_{j \in N_{in}(i)} \frac{\mathcal{I}_{in}^{(p)}(j)}{|N_{out}(j)| - 1} \quad \forall (i \in V) \quad (8)$$

Note that if $|N_{out}(j)| \leq 1$, then let $|N_{out}(j)| = 2$ for the divisor to be 1.

3.2 Ability-to-influence

The second centrality is the ability-to-influence, denoted by $\mathcal{I}_{out}$. This indicates the influencing role of the node on its neighbors. Specifically, we look at all the combinations of incoming edges to the neighbourhood which include the edge from the investigated node. Of these, we count the ones whose sum of weights reaches the threshold value of the node and calculate the weighted incidence value for this case, denote w_{from}. As calculated for the influenceability, we divide the number of infecting edges by the number of edges in the edge combination and all occurrences of a

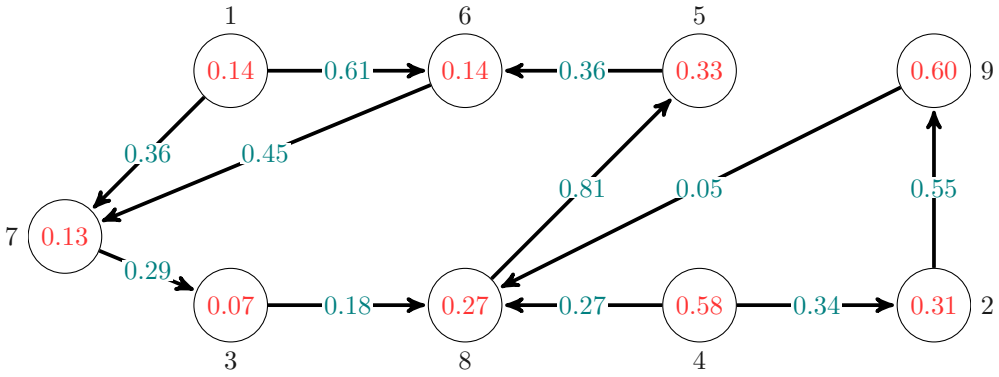

Figure 1: Example graph

given number of edge combinations. Finally, summarized w_{from} for each investigated combinations of edges.

The calculation of $\mathcal{I}_{out}^{(p)}$ value for node 4 is shown in Table 3. To do this, we need to look at the neighbours of node 4, i.e., the edges coming into the vertices 8 and 2. We have seen the combinations of edges coming into vertex 8 in Table 1, but only those that involve the edge coming from vertex 4 are needed.

To calculate this efficiently, we create a table with rows and columns representing the vertices of the graph. Row i and column j show the role of vertex j in the contamination of vertex i. The column sum of the table gives the $\mathcal{I}_{out}^{(p)}$ value of the vertices. We see this counting table in Table 4 which also shows the calculated $\mathcal{I}_{out}^{(p)}$ values for the vertices of the small graph in Fig. 1.

Table 3: The $\mathcal{I}_{out}^{(p)}$ value for node 4 of the graph in Fig. 1.

combinations of edges	sum	θ	$w_{from}(4)$
$4 \to 8$	0.27	0.27	$1/(1 \cdot 1)$
$9 \to 8, 4 \to 8$	0.31	0.27	$1/(2 \cdot 2)$
$4 \to 8, 3 \to 8$	0.44	0.27	$1/(2 \cdot 2)$
$9 \to 8, 4 \to 8, 3 \to 8$	0.49	0.27	$1(3 \cdot 1)$
$4 \to 2$	0.34	0.31	$1/(1 \cdot 1)$
			$\mathcal{I}_{out}^{(p)}(4) = 2.83$

The ability-to-influence value of a vertex is obtained by adding to the value of $\mathcal{I}_{out}^{(p)}$ the approximation of the ability-to-influence of its out-neighbors:

$$\mathcal{I}_{out}(i) = \mathcal{I}_{out}^{(p)}(i) + \sum_{j \in N_{out}(i)} \frac{\mathcal{I}_{out}^{(p)}(j)}{|N_{in}(j)| - 1} \quad \forall (i \in V) \quad (9)$$

Note that if $|N_{in}(j)| \leq 1$, then let $|N_{out}(j)| = 2$ for the divisor to be 1.

3.3 Potential seed selection

Using the two centrality values, we want to determine which vertices can be seeds. Therefore, first, the centrality values are normalized between 0 and 1 in a way that all the elements are divided with the maximum. Such normalization is denoted in each case by $||.||$. Then, we sort the nodes according to their centrality value. We put them in descending order according to their ability-to-influence value, since seed vertices should have good ability-to-influence's value. Conversely, we rank the vertices in ascending order according to their influenceability value, since seed vertices are unlikely to be easily infected. We take the weighted sum of the two order values for each node to get $\mathcal{I}$. This is shown in equation (10):

Table 4: The $\mathcal{I}_{out}^{(p)}$ values for the graph in Fig. 1.

	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	1.00	0	0	0	0	0
3	0	0	0	0	0	0	1.00	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1.00	0
6	1.50	0	0	0	1.50	0	0	0	0
7	1.50	0	0	0	0	1.50	0	0	0
8	0	0	0.58	1.83	0	0	0	0	0.58
9	0	0	0	0	0	0	0	0	0
$\mathcal{I}_{out}^{(p)}$	3.00	0.00	0.58	2.83	1.50	1.50	1.00	1.00	0.58

$$\mathcal{I}(i) = \alpha \cdot \text{ord}(||\mathcal{I}_{out}(i)||) + (1-\alpha) \cdot \text{ord}(||\mathcal{I}_{in}(i)||), \quad (10)$$

for all $i \in V$.

The normalized values of influenceability and ability-to-influence for the graph in Figure 1 are shown in Table 5. Also, the $\mathcal{I}$ values are calculated from them.

Finally, to form the set of potential seed nodes, choose the subset of $V(G)$ according to $\mathcal{I}$. This is controlled by a parameter $0 < r < 1$, thus the cardinality of the candidate seeds set is $r \cdot |V(G)|$.

Table 5: The value of $||\mathcal{I}_{in}||$, $||\mathcal{I}_{out}||$ and $\mathcal{I}(i)$ with $\alpha = 0.0$ and $\alpha = 0.8$ for the graph in Figure 1.

node	$\|\|\mathcal{I}_{in}\|\|$	$\|\|\mathcal{I}_{out}\|\|$	$\mathcal{I}(i), \alpha = 0.0$	$\mathcal{I}(i), \alpha = 0.8$
1	0.0	1.0	1	1.0
2	0.33	0.11	3	7.8
3	0.83	0.20	7	7.0
4	0.0	0.61	2	2.0
5	0.67	0.55	5	3.4
6	0.83	0.45	8	4.8
7	1.0	0.29	9	6.6
8	0.67	0.45	6	5.2
9	0.33	0.20	4	7.2

4 Algorithms

Before details are given about our new heuristic method it needs to be emphasized that due to the nature of our problem, namely finding the maximum number of influenced nodes with minimum diffusion time steps, all algorithms work iteratively and use the ILP model $(1) - (7)$ and their stopping criteria is to run until infeasibility. This usually leads to a long running time.

4.1 Proposed heuristic: IAtI

Here we describe our proposal for a heuristic which selects a candidate seeder set of graph nodes based on the new centrality metrics introduced in Section 3. Since it is using the Influenceability and the Ability-to-Influence measures we refer to it as IAtI-heuristic.

The method is described in Algorithm 2. In its precondition phase, in Step 1 and 2 it calculates the two centrality metrics and the combination of them using the formulae given in Section 3. In Step 3 the algorithm collects the set of possible seed nodes. It is a parameter $0 < r < 1$ which controls the ratio of the nodes to be selected. Higher r value leads to higher probability for the seed nodes corresponding to the global optimum to be selected into the candidate set. However, high r value also leads to higher execution time, thus it needs to be set up with care. Steps $4 - 6$ describe the iterative part of the algorithm. This is essentially the same as the algorithm we proposed in our recent work [5]. Note that the algorithm usually iterates until the ILP model becomes infeasible, as there is no other stopping criteria, unless all the nodes become active; the reader is referred to [5] for the theoretical proofs.

Discussion on parameters' choice for IAtI Now that we have Algorithm 2, let us see how to set up its parameters in order to have high chance to include the seed nodes of the global optimum and exclude those which might not be good candidates. As we have calculated the centrality metrics for the small test graph from Figure 1 as well as the globally optimal solution using $k = 2$ seed nodes, the scatterplot shown in Figure 2 clearly suggests that we shall aim

Algorithm 2 IAtI-heuristic(r, α)

Input A directed graph G with edge weights and node threshold values.

Step 1 Calculate $\mathcal{I}_{in}^{(p)}$ and $\mathcal{I}_{out}^{(p)}$ for all $i \in V$ and then $\mathcal{I}_{in}$ and $\mathcal{I}_{out}$ using Eq. (8) and (9), respectively.

Step 2 Form $\mathcal{I}$ for each vertex according to the Equation (10) using the input parameter α.

Step 3 Define $S \subseteq V(G)$ to be the set of possible seeds: choose the top $r \cdot |V(G)|$ number of nodes from $\mathcal{I}$.

Step 4 Let $\mathcal{T} := 2$ and start the iteration.

Step 5 Solve the ILP defined by $\{(1)$ - $(7)\}$ for the diffusion time value $\mathcal{T}$, so that the seed vertices can be chosen exclusively from the set S.

Step 6 If all the nodes are influenced or the solution becomes infeasible then stop the iteration. Otherwise, let $\mathcal{T} = \mathcal{T} + 1$ and go back to Step 5.

for larger ability-to-influence value together with low influenceability. The explanation is that the seed nodes of the global optimum are colored red (node 4 and 1). Larger circle represents later activation in the diffusion in the globally optimal solution[2]. Obviously, this landscape of the centrality values is quite simple as our small test graph is of special structure. Discussion some larger graphs will be given in Section 5.3.

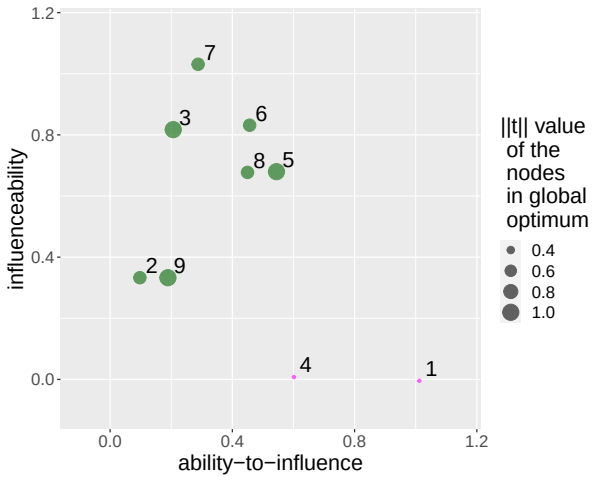

Figure 2: Visualization of the new centralitites for the small test graph

[2]It must be emphasized here that the size of the circles on Figure 2 represents the activation time in the globally optimal solution. For non-optimal $\mathcal{T}$ value we might obtain different times.

Two different α values were used from which potential seeds were selected in equal proportions. In the first case, the weight of the ability-to-influence property is set to a higher value, so $\alpha = 0.8$ and select elements with a ratio $r = 0.2$ which are the best elements according to $\mathcal{I}$. Moreover, we add the vertices with ability-to-influence value equal to 1. This is necessary because it has the highest chance of infection, yet in many cases it will not be the seed. There could be several reasons for this, for example another node could reach the same global optimum or such a node could be a neighbour of the seed that could infect the seed. In any case, choosing these nodes gives a better chance for finding the global optimum. Denote this set by S_1. In the second case, we select based on the influenceability value alone, so $\alpha = 0$ and added nodes with 0 influenceability value. This is because nodes with an influence value of 0 can never be influenced unless they become seed nodes themselves. Therefore, any vertex with an influence value of 0 is chosen to be a seed vertex. The resulting set is denoted by S_2. Finally, take the intersection of S_1 and S_2 and obtain the set S. The seed nodes are chosen from this set S.

4.2 Greedy

Although the so-called submodularity property does not hold for the DLTM, the greedy approach is still a favorable method for solving the IM problem. We have adapted the greedy strategy into our ILP framework. For the formal description of the method the reader is referred to the Algorithm 3 in the Appendix.

5 Numerical experiments

5.1 Computational environment

The implementation of the above proposed new centralities was done in R version 4.1.2 using its `igraph` 1.3.5 package. The numerical experiments were executed with Gurobi 10.0 called from AMPL [7] using non-default options: `threads=8 lpmethod=0 cuts=0 mipgapabs=1e-2`. The computer used had Intel Core i7-10700 CPU at 2.90GHz with 16GB memory running Ubuntu 22.04.2. Note that we used multi-core setup of Gurobi (i.e., for solving the ILP models), whereas R was used with single threading. In the R implementation of the new centrality metrics we used the different `apply` functionalities to process matrices and lists efficiently.

5.2 Test graphs

Synthetic random graphs To generate random test graphs we used the LFR scheme, which creates networks with prescribed community structures [15]. Note that this procedure only provides the graphs of social-network type, the edge weights and node thresholds needed to be assigned in the second phase. Similarly to our earlier work [5], the following procedure was used.

Table 6: A selection of real-world test graphs

name	N	M	$d_{\max}$
soc-dolphins [20]	62	159	5
ca-sandi-auths	86	124	12
retweet [25, 24]	96	117	17
ca-netscience [21]	379	914	34

– Regarding the edge weights: nodes with in-weights larger than 1 (given by the LFR method) were normalized to 1; and we applied a multiplication with a factor r_w which was a uniform at random number in the interval $[0.6, 1]$.

– The nodes' threshold values were generated uniform at random in the interval $[0.05, 0.6]$.

– From the LFR method three parameters were fixed: mixing parameter $\mu_w = 0.1$, minimal community size $minc = 5$, maximal community size $maxc = 42$.

– Three parameters were varied in the experiments: number of nodes n, the average degree $avgk$, and maximum degree $maxk$.

Real-world graphs We have tested our proposed method on a small set of real-world graphs [23], see Table 6 for the details. Note that the last column indicates the maximum degree of the given graph. For our problem setup $d_{\max}$ needs to be relatively small, since the exact ILP solver as well as our heuristic does not scale efficiently. Nevertheless, for these four social networks we did the following experiments.

The real graphs were undirected and unweighted graphs. The `mutual` parameter was used which makes the graphs directed by doubling the undirected edges. We then created 3 groups according to the percentage of edges that were deleted randomly and how we generated weight of edges and threshold values. These groups and their corresponding generating parameters are shown in Table 7. The edges and weights were generated uniform at random in the given interval. As for the weights of the edges, similar to the LFR graphs, nodes with weights greater than 1 were normalized to 1 and multiplied by a factor r_w which were randomly chosen from the interval $[0.6, 1]$.

Table 7: The parameters used to generate real-world test graphs

group	% of edges deleted	threshold	weights
#1	45	$[0.05, 0.5]$	$[0.01, 0.6]$
#2	50	$[0.1, 0.5]$	$[0.05, 0.5]$
#3	50	$[0.07, 0.55]$	$[0.075, 0.55]$

5.3 Results

For all the test graphs we ran the new IAtI-heuristic and compare its results with those obtained by Greedy. For reference we also solved the smaller sized problems with Gurobi to get the global optimal solution. Note that we fixed the seed set size to 2.

LFR graphs The results are reported in Table 8 (including the globally optimal solution) and 9 (without global optima[3]). If the obtained results (with respect to σ and $\mathcal{T}$) were different, then the better solution is emphasized as boldface. For the smaller sized problems we can see that there were 3 cases when neither our heuristic nor the greedy approach was able to find the global optimum, see Table 8. Greedy found better solutions than our heuristic in seven cases. Also in 7 cases (that is 23%) our heuristic missed the global optimum. On the other hand, Greedy lost the global optimum in 11 cases (37% of the cases). Regarding the running time, the greedy approach is far the quickest method. On the other hand, IAtI-heuristic is faster that the exact method, usually around 4−12 times.

Table 8: Benchmarking results for the LFR graphs; time in seconds

graph parameters			IAtI-heuristic			greedy			global opt.		
n	$avgk$	$maxk$	σ	$\mathcal{T}$	time	σ	$\mathcal{T}$	time	σ^*	$\mathcal{T}^*$	time
100	3	8	87	13	22.2	87	13	11.9	87	13	164.2
100	4	8	74	17	77.1	74	17	7.6	74	17	1442.3
100	4	10	100	15	11.7	100	15	5.5	100	15	20.6
100	5	8	88	24	339.1	88	24	12.4	88	24	4287.9
100	5	10	89	10	168.7	89	10	5.9	89	10	1742.7
100	6	8	88	17	348.1	88	17	9.6	88	17	4353.3
105	3	8	58	11	8.9	58	11	5.4	58	11	92.0
105	4	8	57	10	15.5	57	10	2.4	57	10	169.0
105	4	10	**80**	**15**	16.0	79	14	6.2	80	15	42.2
105	5	8	85	16	20.1	85	16	7.4	85	16	86.9
105	5	10	66	20	41.7	66	20	8.7	66	20	108.0
105	6	8	61	14	27.7	61	14	4.8	**61**	**12**	119.5
110	3	8	75	20	110.0	75	20	10.7	**75**	**15**	655.8
110	4	8	**96**	**23**	411.9	87	15	9.4	96	23	5510.2
110	4	10	91	16	20.9	91	16	10.1	91	16	188.0
110	5	8	90	18	38.9	90	18	9.4	90	18	313.7
110	5	10	70	15	236.5	70	15	7.1	70	15	430.7
110	6	8	**95**	**15**	221.0	95	19	16.6	95	15	1085.0
115	3	8	56	13	6.6	56	13	4.6	56	13	59.3
115	4	8	49	18	13.5	45	14	6.4	49	18	170.5
115	4	10	43	11	12.3	**49**	**13**	4.2	49	13	64.3
115	5	8	**25**	**18**	24.9	21	7	1.1	25	18	219.5
115	5	10	**78**	**20**	81.7	68	14	6.3	78	20	193.1
115	6	8	**49**	**13**	18.2	46	11	4.3	49	13	292.7
120	3	8	86	26	59.7	**86**	**25**	23.4	86	25	1402.6
120	4	8	**77**	**17**	30.9	75	15	6.9	77	17	307.2
120	4	10	113	20	36.4	**113**	**14**	18.2	113	14	274.9
120	5	8	70	12	97.0	70	14	9.6	**70**	**11**	469.5
120	5	10	82	11	67.6	**83**	**17**	9.9	83	17	326.7
120	6	8	**98**	**15**	113.7	98	17	12.5	98	15	1160.0

For the larger graphs, reported in Table 9 we can see that the in terms of running time Greedy is much faster than our heuristic. Greedy was only 7 cases found better solution than our heuristic. On the other hand IAtI-heuristic was in 13 cases more successful than the greedy approach and among these cases there are several ones where it obtained much better solution.

[3]Due to the long running time for the larger graphs the globally optimal solutions are not available.

Table 9: Benchmarking results for the LFR graphs, globally optimal solution not available; time in seconds

graph parameters			IAtI-heuristic			greedy		
n	$avgk$	$maxk$	σ	$\mathcal{T}$	time	σ	$\mathcal{T}$	time
125	3	8	116	17	35.5	116	17	13.4
125	4	8	102	24	407.1	102	24	25.6
125	4	10	93	22	115.0	**98**	**27**	25.1
125	5	8	81	15	303.7	81	15	16.9
125	5	10	99	19	220.9	**99**	**18**	29.4
125	6	8	**104**	**13**	406.1	104	14	15.2
130	3	8	121	28	604.7	121	28	90.4
130	4	8	**96**	**14**	266.5	84	10	11.7
130	4	10	**51**	**9**	80.6	51	10	5.9
130	5	8	130	20	116.5	**130**	**19**	16.3
130	5	10	114	16	355.2	**116**	**16**	27.1
130	6	8	130	15	54.2	130	15	11.9
135	3	8	129	19	22.7	**135**	**18**	10.6
135	4	8	117	13	215.9	117	13	22.7
135	4	10	93	14	65.4	93	14	15.8
135	5	8	**74**	**21**	864.4	55	16	12.1
135	5	10	81	13	234.5	81	13	15.5
135	6	8	**113**	**23**	1059.6	97	18	36.3
140	3	8	57	11	17.6	57	11	18.4
140	4	8	**116**	**20**	247.9	114	16	13.3
140	4	10	82	17	32.5	82	17	14.0
140	5	8	**93**	**18**	369.2	72	16	22.7
140	5	10	111	17	172.8	111	17	21.9
140	6	8	**130**	**32**	1283.5	103	17	17.5
145	3	8	70	16	34.6	70	16	14.0
145	4	8	100	21	273.4	100	21	45.2
145	4	10	60	18	123.8	**76**	**19**	17.2
145	5	8	**58**	**20**	134.7	44	14	7.8
145	5	10	**110**	**21**	621.5	108	20	30.0
145	6	8	101	25	885.1	101	25	31.1
150	3	8	72	12	107.1	72	12	17.2
150	4	8	66	13	167.3	**66**	**12**	9.8
150	4	10	**84**	**16**	78.6	83	17	13.9
150	5	8	**123**	**21**	290.5	118	19	28.2
150	5	10	89	27	1067.1	89	27	61.5
150	6	8	**130**	**22**	1207.6	124	18	20.6

To discuss a particular example at which the IAtI-heuritic lost the globally optimal solution, whereas Greedy was able to find it, see Figure 3. The scatterplot show the two centralities of the nodes together with their activation time at the optimal $\mathcal{T}^*$. The empty circles correspond to those nodes which were not selected by our heuristic to be candidates for seeds. As shown by the red circles, we can see that IAtI dropped the optimal seed set, as only those were selected to be candidates which are shown with dark-blue color. Note that Greedy, instead, found those red-colored nodes. The seed nodes chosen by the IAtI-heuristic are shown as yellow colored circles. It would be possible to parameterize the IAtI-heuristic in such a way that it would include the optimal seeds in the candidate set. Figure 3 suggests that in that case the cardinality of the candidate seed set would be necessary larger which would result in much longer running time.

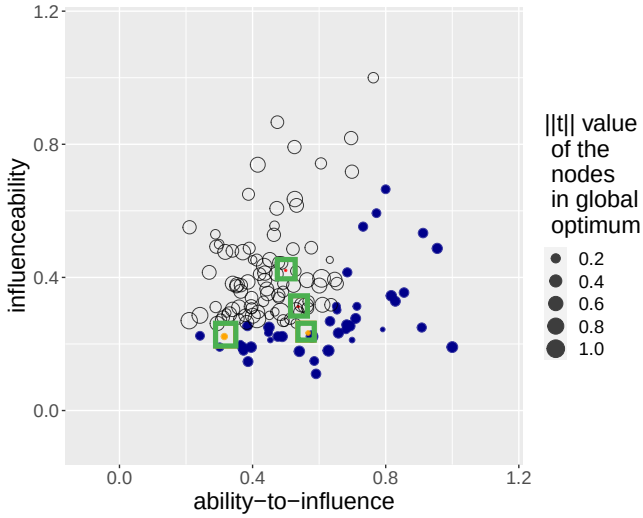

Figure 3: Visualization of the new centralitites for the LFR.135.3.8 graph

Real-world graphs Our experiments on some selected real world graphs are reported in Table 10. As in the earlier tables it is shown in boldface if a method obtained a better solution than the other one. We can observe that the trend of the IAtI-heuristic being much slower than Greedy remains to be the case also here. There was only once case when the greedy approach found a better solution than our heuristic. On the other hand, IAtI-heuristic was able to find better solution in 5 cases (out of 12), which corresponds to 42%.

Table 10: Results for the small real-world graphs; time in second

	IAtI-heuristic			greedy		
graph	σ	$\mathcal{T}$	time	σ	$\mathcal{T}$	time
soc-dolphins #1	**45**	**18**	11.9	41	10	5.1
ca-sandi-auths #1	**34**	**9**	6.9	30	9	1.7
retweet #1	22	7	5.5	22	7	0.9
ca-netscience #1	63	9	319.5	63	9	183.6
soc-dolphins #2	**34**	**9**	6.9	24	5	0.5
ca-sandi-auths #2	**13**	**5**	5.5	12	4	0.3
retweet #2	19	6	4.8	19	6	0.7
ca-netscience #2	25	7	3047.7	**25**	**6**	13.8
soc-dolphins #3	27	8	5.5	27	8	1.5
ca-sandi-auths #3	14	5	7.6	14	5	0.5
retweet #3	17	6	4.9	17	6	0.6
ca-netscience #3	**41**	**11**	300.1	30	5	10.2

Finally, let us demonstrate again the seed node selection strategy of the IAtI-heuristic on the ca-sandi-auths graph as shown in Figure 4. The red colored circles represent the nodes found by the Greedy approach, leading to a suboptimal result. The red circle on the left was exluded by the IAtI-heuristic from the possible set of seed nodes. As before, the candidate set of seed nodes are colored with dark-blue. Among them, there are the two optimal seed nodes shown as yellow circles. Note that there are quite many nodes with 0 influenceability value. Those are definitely selected by IAtI as possible seed nodes as they are impossible to be activated by other nodes. As we can observe, one of them is indeed part of the optimal seed set.

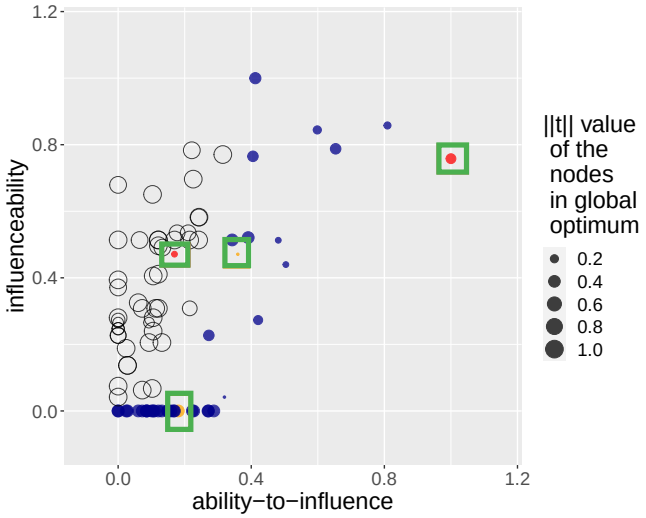

Figure 4: Visualization of the new centralitites for the ca-sandi-auths #1 real graph

6 Conclusion

We proposed two new centrality metrics for the influence maximization under deterministic linear threshold model. These metrics take into account the structure of the input network, more precisely the weight of edges, the combinations of edges and the threshold value of vertices. This is a great advantage of our method, because it is not usual to include the node's threshold in the centrality measure. Using the two centrality metrics, we selected vertices that have a high probability of being seed nodes. The solver now selects seed vertices only among these. This reduces the computational complexity of the task and therefore, compared to running the ILP solver on the unrestricted model, it speeds up the procedure. Using those metrics, we created the so-called IAtI algorithm. This was compared with Greedy and in some cases with the global optimum, also. The IAtI algorithm is slower than Greedy, but in many cases it gives a better solution and in most cases it finds the global

optimum. Although most real-world networks are sparse graphs, for which our method works well, the disadvantage of this method is that it takes a lot of computing time to generate and use edge combinations for large degree of nodes. Thus, it may take a long time for pre-processing and is therefore, in its current form, not scalable. Our future work is to develop a version of the method that can handle graphs including nodes with relatively large degree.

Acknowledgement

The research leading to these results has received funding from the national project TKP2021-NVA-09. Project no. TKP2021-NVA-09 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development, and Innovation Fund, financed under the TKP2021-NVA funding scheme. The work was also supported by the grant SNN-135643 of the National Research, Development, and Innovation Office, Hungary.

References

[1] D. Acemoglu, A. Ozdaglar, and E. Yildiz. Diffusion of innovations in social networks. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 2329–2334, 2011. https://doi.org/10.1109/cdc.2011.6160999

[2] F. Altarelli, A. Braunstein, L. Dall'Asta, and R. Zecchina. Optimizing spread dynamics on graphs by message passing. *Journal of Statistical Mechanics: Theory and Experiment*, page P09011, 2013. https://doi.org/10.1088/1742-5468/2013/09/p09011

[3] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology*, 2:113–120, 1972. https://doi.org/10.1080/0022250x.1972.9989806

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998. https://doi.org/10.1016/s0169-7552(98)00110-x

[5] E. Csókás and T. Vinkó. An exact method for influence maximization based on deterministic linear threshold model. *Central European Journal of Operations Research*, 31:269–286, 2023. https://doi.org/10.1007/s10100-022-00807-3

[6] K. Das, S. Samanta, and M. Pal. Study on centrality measures in social networks: a survey. *Social Network Analysis and Mining*, 8, 2018. https://doi.org/10.1007/s13278-018-0493-2

[7] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL. A modeling language for mathematical programming*. Thomson, 1993. https://doi.org/10.1007/978-3-642-83724-1_12

[8] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977. https://doi.org/10.2307/3033543

[9] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001. https://doi.org/10.1023/a:1011122126881

[10] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83:1420–1443, 1978. https://doi.org/10.1086/226707

[11] F. Gursoy and D. Gunnec. Influence maximization in social networks under deterministic linear threshold model. *Knowledge-Based Systems*, 161:111–123, 2018. https://doi.org/10.1016/j.knosys.2018.07.040

[12] P.D. Karampourniotis, B.K. Szymanski, and G. Korniss. Influence maximization for fixed heterogeneous thresholds. *Scientific Reports*, 9:5573, 2019. https://doi.org/10.1038/s41598-019-41822-w

[13] 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*, pages 137–146, 2003. https://doi.org/10.1145/956750.956769

[14] S. Kochemazov. Comparative study of combinatorial algorithms for solving the influence maximization problem in networks under a deterministic linear threshold model. *Procedia Computer Science*, 136:190–199, 2018. 7th International Young Scientists Conference on Computational Science, YSC2018. https://doi.org/10.1016/j.procs.2018.08.252

[15] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78:046110, 2008. https://doi.org/10.1103/physreve.78.046110

[16] B. Liu, G. Cong, D. Xu, and Y. Zeng. Time constrained influence maximization in social networks. In *2012 IEEE 12th International Conference on Data Mining*, pages 439–448, 2012. https://doi.org/10.1109/icdm.2012.158

[17] Z. Lu, W. Zhang, W. Wu, B. Fu, and D. Du. Approximation and inapproximation for the influence maximization problem in social networks under deterministic linear threshold model. In *2011 31st International Conference on Distributed Computing Systems*

Workshops, pages 160–165, 2011. https://doi.org/10.1109/icdcsw.2011.33

[18] Z. Lu, W. Zhang, W. Wu, and J. Kim. The complexity of influence maximization problem in the deterministic linear threshold model. *Journal of Combinatorial Optimization*, 24:374–378, 2012. https://doi.org/10.1007/s10878-011-9393-3

[19] Z. Lu, W. Zhang, W. Wu, J. Kim, and B. Fu. The complexity of influence maximization problem in the deterministic linear threshold model. *Journal of Combinatorial Optimization*, 24(3):374–378, 2012. https://doi.org/10.1007/s10878-011-9393-3

[20] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003. https://doi.org/10.1007/s00265-003-0651-y

[21] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006. https://doi.org/10.1103/physreve.74.036104

[22] K. Rahimkhani, A. Aleahmad, Ma. Rahgozar, and A. Moeini. A fast algorithm for finding most influential people based on the linear threshold model. *Expert Systems with Applications*, 42:1353–1361, 2015. https://doi.org/10.1016/j.eswa.2014.09.037

[23] R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. https://doi.org/10.1609/aaai.v29i1.9277

[24] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, and M. A. Patwary. What if clique were fast? maximum cliques in information networks and strong components in temporal networks. *arXiv preprint arXiv:1210.5802*, pages 1–11, 2012. https://doi.org/10.48550/arXiv.1210.5802

[25] R. A. Rossi, D. F. Gleich, A. H. Gebremedhin, and M. A. Patwary. Fast maximum clique algorithms for large graphs. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*, 2014. https://doi.org/10.1145/2567948.2577283

[26] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966. https://doi.org/10.1007/bf02289527

[27] S. Segarra and A. Ribeiro. Stability and continuity of centrality measures in weighted graphs. *IEEE Transactions on Signal Processing*, 64:543–555, 2016. https://doi.org/10.1109/tsp.2015.2486740

[28] M. E. Shaw. Group structure and the behavior of individuals in small groups. *The Journal of Psychology*, 38:139–149, 1954. https://doi.org/10.1080/00223980.1954.9712925

[29] A. Singh, R. Singh, and S. Iyengar. Node-weighted centrality: a new way of centrality hybridization. *Computational Social Networks*, 7:Article nr. 6, 2020. https://doi.org/10.1186/s40649-020-00081-w

[30] N. Rama Suri and Y. Narahari. Determining the top-k nodes in social networks using the Shapley value. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, pages 1509–1512, 2008. https://dl.acm.org/doi/10.5555/1402821.1402911

[31] R. Xu. An L_p norm relaxation approach to positive influence maximization in social network under the deterministic linear threshold model. In *Algorithms and Models for the Web Graph*, pages 144–155. Springer, 2013. https://doi.org/10.1007/978-3-319-03536-9_12

Appendix

The description of the greedy approach is given in Algorithm 3.

Algorithm 3 Greedy algorithm using the ILP model

Step 1 Let $k := 1$ be the number of seed nodes.

Step 2 Let $\mathcal{T} := 2$ and start the iteration with diffusion time.

Step 3 Solve the ILP defined by $\{(1)\text{-}(7)\}$ for the diffusion time value $\mathcal{T}$.

Step 4 If all the nodes are influenced or the model becomes infeasible then the optimum is found and stop the iteration with $\mathcal{T}$ and go to Step 5. Otherwise, let $\mathcal{T} = \mathcal{T}+1$ and go back to Step 3.

Step 5 If $k = |S|$ where S is set of seeds, then the stop the algorithm. Otherwise, fix the selected seed nodes for the remaining iterations. Let $k := k+1$ and go to Step 2.