



Transactions on Data Analysis in Social Science

Journal Homepage: <https://transoscience.ir>



HNoF: A Neighbors-of-Friends Heuristic for Influence Maximization in Social Networks

P. Nair^{1,*}

¹ Department of Computer Science, Faculty of Engineering, Indian Institute of Science (IISc), Bengaluru, India

ARTICLE INFO	ABSTRACT
Article History: Received 13 January 2025 Received in revised form 4 March 2025 Accepted 11 April 2025 Available online 5 September 2025	Influence maximization (IM) seeks a set of k seed users that maximizes the expected spread of influence in a social network. Because IM is NP-hard and computing the spread is #P-hard, simulation-based greedy algorithms achieve near-optimal spread but scale poorly, whereas fast heuristics often sacrifice much of the achievable spread. This paper proposes HNoF (Heuristic based on Neighbors of Friends), a greedy heuristic that restricts candidates to the two-hop neighbors of the current seed that are not its direct friends and ranks them by their marginal gain in the Expected Diffusion Value (EDV), a closed-form one-hop approximation of spread under the Independent Cascade (IC) model. This restriction prunes most of the graph at each iteration, and the EDV score replaces Monte Carlo simulation during seed selection. A derived closed-form expression for the EDV marginal gain shows how HNoF penalizes overlapping neighborhoods. HNoF was compared with General Greedy, LIR, SAEDV and DDSE on four real-world networks (NetHEPT, NetPHY, Epinions1 and Slashdot0902) under the IC model ($p = 0.01$, $k = 5-50$). On the two collaboration networks, it came within about 1–4% of the spread of General Greedy at $k = 50$ while running roughly four orders of magnitude faster. On all four networks, it achieved the largest spread among the scalable methods, exceeding DDSE by about 1–6%, although at a higher running time than DDSE and SAEDV on the largest networks. Combining a locality-constrained candidate set with an analytical spread estimate thus offers a favorable balance between spread and efficiency.
Keywords: Influence Maximization; Social Networks; Viral Marketing; Independent Cascade Model; Expected Diffusion Value; Information Diffusion; Greedy Heuristics	

1. INTRODUCTION

1.1. Background and motivation

Online social networks have become a primary channel through which people share opinions, emotions, news and ideas [1]. As both the number of users and the intensity of their engagement keep growing, these platforms have also turned into powerful instruments for advertising and marketing [2]. A firm with limited resources, however, cannot distribute a product or message to every user within a short period. The practical question is therefore whom to target first so that adoption spreads as widely and as quickly as possible through interpersonal ties. Viral marketing

* Corresponding Author: PriyaNair1982@gmail.com

Department of Computer Science, Faculty of Engineering, Indian Institute of Science (IISc), Bengaluru, India



answers this question by identifying a small number of influential individuals and relying on word-of-mouth propagation, much as a virus spreads from one host to another, to carry the message to the rest of the network [3, 4].

This targeting problem is formalized as the influence maximization (IM) problem. An IM instance has three essential components: (i) the seed set, the users who are activated at the outset and from whom diffusion starts; (ii) the influence spread, the expected number of users who are active at the end of the diffusion process; and (iii) the computational time required to identify the seed set. IM therefore seeks the seed set that maximizes the spread of information within a feasible time budget [5]. Beyond viral marketing, IM supports political-campaign analysis, opinion formation, link prediction and recommender systems [2], and a large body of work has addressed its algorithmic aspects [6–14].

Domingos and Richardson [4] first framed the problem algorithmically in a probabilistic setting. Kempe et al. [15] then formulated IM as a discrete stochastic optimization problem, proved that it is NP-hard under the Independent Cascade (IC) and Linear Threshold models, and showed that, because the spread function is monotone and submodular, a simple greedy algorithm (General Greedy) achieves a $(1 - 1/e - \epsilon)$ approximation [16]. General Greedy, however, estimates spread by Monte Carlo (MC) simulation and needs $O(k n R)$ simulations, where n is the number of nodes and R the number of MC rounds. This makes it impractical even for networks of moderate size [15].

1.2. Research gap

Subsequent research has followed two broad directions. Acceleration of the greedy framework (e.g., lazy evaluation, edge-sampling snapshots, and reverse-influence sampling) preserves approximation guarantees but still requires substantial computation or memory. Heuristic, centrality-based, community-based and metaheuristic methods are much faster but give up guarantees and often lose a considerable share of the achievable spread, particularly in sparse, low-probability regimes where influence remains local. As reviewed in Section 2, most methods improve either spread or running time. Achieving a balance between high spread and low running time therefore remains an open and practically important challenge [17, 18].

Two specific gaps motivate the present study. First, most fast heuristics select seeds from a global ranking of all nodes (e.g., by degree or a local index) and only a posteriori discount overlap between seeds. They do not exploit the structural fact that, under low propagation probabilities, the useful candidates for the next seed are those that are close enough to existing seeds to benefit from shared social context, yet far enough away to avoid redundant coverage. Second, analytical spread estimators such as the Expected Diffusion Value (EDV) [19] have mainly been used as fitness functions inside stochastic metaheuristics (SAEDV, DDSE), whose results vary from run to run and whose quality depends on parameter tuning. Their use inside a deterministic, candidate-restricted greedy procedure has received little attention.

1.3. Contributions

To address these gaps, this paper proposes HNoF (Heuristic based on Neighbors of Friends), a deterministic greedy heuristic for IM under the IC model. Its main contributions are as follows:

A locality-constrained candidate set. At every iteration, HNoF considers as candidates only the *neighbors of the friends* of the current seed, i.e., nodes at distance exactly two that are not direct friends of the seed. Such nodes share at least one mutual friend with an active seed, whose activation probability is therefore raised, while their own one-hop neighborhoods overlap less with that of the seed. Restricting the search to this set removes most nodes from the computation at each step.

Analytical greedy scoring. Candidates are ranked by the marginal gain in EDV, so no MC simulation is needed during seed selection. We derive a closed-form expression for this marginal gain (Equation 4) that makes explicit how HNoF rewards new coverage and penalizes overlap with already-covered nodes. It also gives an $O(d(v))$ evaluation cost per candidate.

Empirical evidence of a favorable spread–efficiency balance. On four real-world networks of different scales, HNoF attains spread close to General Greedy and the highest spread among the scalable baselines, at a running time

several orders of magnitude lower than General Greedy. We also report and analyze its running-time disadvantage relative to the fastest metaheuristics, which delimits where the method is most useful.

1.4. Organization

Section 2 reviews related work. Section 3 states the problem and the diffusion model. Section 4 describes HNoF, including a worked example and a complexity analysis. Section 5 presents the experimental setup and Section 6 the results. Section 7 discusses the findings, their implications and limitations, and Section 8 concludes the paper.

2. RELATED WORK

IM is NP-hard, so no polynomial-time exact algorithm is known, and researchers have proposed a variety of approximate and heuristic approaches. We group them into five families.

2.1. Simulation-based greedy algorithms and their accelerations

Kempe et al. [15] introduced General Greedy, which iteratively adds the node with the largest MC-estimated marginal gain in spread. Despite its approximation guarantee and high spread, the large number of simulations makes it inefficient on large networks. Leskovec et al. [20] proposed CELF, which exploits submodularity through *lazy evaluation* to avoid recomputing marginal gains that cannot change the selection. CELF was reported to be up to 700 times faster than General Greedy but remains slow on large graphs; CELF++ [21] further reduced the number of spread evaluations. Chen et al. [22] proposed NewGreedyIC, which uses the equivalence between the IC model and *live-edge* graphs [15, 23] to evaluate the marginal gains of all nodes from a single sampled snapshot, and NewGreedyWC, which uses Cohen’s randomized reachability estimation for the weighted-cascade model. Both reduced the running time of General Greedy and CELF, but further improvement was still necessary. Cheng et al. [24] introduced StaticGreedy, which reuses a fixed set of snapshots across iterations; its variant StaticGreedyDU is efficient but was reported to give lower spread in some settings.

2.2. Reverse-influence-sampling algorithms

A more recent line of work estimates spread through *reverse reachable* sets [25]. TIM [26] and IMM [27] achieve a $(1 - 1/e - \epsilon)$ approximation in near-linear expected time. TipTop [3] pushes accuracy close to exact solutions on billion-scale networks. These methods dominate the theoretical landscape, but their memory consumption grows quickly as ϵ decreases and as p increases.

2.3. Heuristic and centrality-based algorithms

Chen et al. [22] proposed Degree Discount, which selects high-degree nodes but discounts the degree of nodes adjacent to already-selected seeds. It is extremely fast but offers no approximation guarantee. Narayanam and Narahari [28] proposed SPIN, which ranks nodes by their Shapley value in a cooperative game; it requires little memory but yields notably lower spread than greedy methods. Wang et al. [29] proposed MIA, which localizes spread computation to *maximum influence arborescences* whose paths have probability above a threshold θ . MIA avoids MC simulation and is fast, but its accuracy depends on the choice of θ . Goyal, Lu, and Lakshmanan [30] proposed SIMPATH for the Linear Threshold model, which builds on CELF and estimates spread by enumerating simple paths; it runs efficiently but does not guarantee the approximation ratio. Morone et al. [31] proposed Collective Influence (CI), grounded in optimal percolation, which restricts computation to a ball of radius ℓ around each node; it performs well on dense networks, but both its accuracy and its running time depend on ℓ and on k . Liu et al. [32] proposed LIR, which computes for each node a local index LI , the number of neighbors with a higher degree, and selects nodes with $LI = 0$ in descending order of degree; LIR is relatively fast but not optimal in selecting influential nodes. Rui et al. [33] improved on LIR with reversed node ranking (RNR), which is efficient but does not always avoid the *rich-club* effect, in which several seeds are chosen from the same densely interconnected core. Ahajjam and Badir [34] proposed Hybrid Rank, which combines centrality measures and explicitly avoids the rich-club effect, but without an approximation guarantee. Wu et al. [13] proposed LAIM, a linear-time iterative method that estimates node influence locally; it is very fast but likewise lacks guarantees. Chen et al. [35] proposed

LUGreedy for *robust* IM, in which edge probabilities are uncertain. This brings the problem closer to real conditions, but the quality of the robust solution depends on the seed budget and the width of the uncertainty intervals. Xie et al. [36] proposed IRR under the membership-based MBIC model, which uses an idea similar to Degree Discount and allows node states beyond active/inactive; its performance depends on k .

2.4. Community-based algorithms

Chen et al. [37] proposed CIM, which detects communities and selects seeds based on them; it is efficient but without approximation guarantees. Shang et al. [9] proposed CoFIM, which evaluates spread within and across communities and is efficient, but it does not account for overlapping nodes. Shang et al. [10] proposed IMPC, based on multi-neighbor potential in community networks. Banerjee et al. [6] proposed CombIM for the budgeted IM problem, which allocates a budget to each community and selects seeds based on node value; the allocation does not fully reflect community characteristics. Qiu et al. [38] proposed PHG, a three-phase improvement of CoFIM that selects some seeds heuristically and the rest greedily; its spread and running time depend on the split between the two phases.

2.5. Metaheuristic algorithms

Jiang et al. [19] proposed simulated annealing (SA) for IM, which achieved higher spread than DegreeDiscount at a high computational cost. They then introduced SAEDV, which replaces simulation with the analytical *Expected Diffusion Value* (EDV) to evaluate candidate solutions. SAEDV is much faster but does not reach the spread of SA. Cui et al. [8] proposed DDSE, an evolutionary algorithm based on differential evolution with a degree-descending search strategy that also uses EDV as its fitness function to accelerate the search; its results are competitive on some, but not all, datasets.

2.6. Positioning of the present study

Table 1 summarizes the trade-offs discussed above. Greedy and sampling-based methods provide guarantees at the cost of computation or memory. Heuristics are fast but rank nodes globally and handle overlap only indirectly. Metaheuristics use analytical estimators such as EDV but are stochastic and parameter-sensitive. HNoF combines the deterministic greedy principle with the analytical EDV estimator and a structurally motivated, two-hop candidate set. This combination has not, to our knowledge, been examined in the studies reviewed above.

Table 1. Summary of representative IM algorithms.

Family	Algorithms	Main strength	Main weakness
Simulation-based greedy	General Greedy, CELF, CELF++, NewGreedy, StaticGreedy	High spread; $(1 - 1/e - \epsilon)$ guarantee	High running time on large graphs
Reverse-influence sampling	TIM, IMM, TipTop	Guarantee with near-linear time	High memory for small ϵ or large p
Heuristic / centrality	DegreeDiscount, SPIN, MIA, SIMPATH, CI, LIR, RNR, HybridRank, LAIM, IRR	Very fast	No guarantee; sensitive to parameters or overlap
Community-based	CIM, CoFIM, IMPC, CombIM, PHG	Scales via decomposition	Depends on community quality; overlap handling
Metaheuristic	SA, SAEDV, DDSE	Flexible; analytical fitness (EDV)	Stochastic; parameter tuning; variable spread
Proposed	HNoF	Deterministic; two-hop candidate pruning; analytical marginal gain	No approximation guarantee (see Section 7.5)

3. PROBLEM FORMULATION

3.1. Notation

A social network is modeled as a graph $G = (V, E, w)$, where $V = \{v_1, \dots, v_n\}$ is the set of nodes (users), $E = \{e_1, \dots, e_m\}$ is the set of edges (relationships), and $w: E \rightarrow (0, 1]$ assigns to each edge (u, v) a propagation probability p_{uv} . In the experiments of this paper, a uniform probability $p_{uv} = p = 0.01$ is used for every edge, which is a standard setting in the IM literature [19, 22]. Undirected networks are treated as directed graphs with arcs in both directions. $N(v)$ denotes the (out-)neighbors of v , $d(v) = |N(v)|$ its degree, and $N(S) = \bigcup_{v \in S} N(v) \setminus S$ the neighbors of a set S outside S . The nodes at hop distance exactly l from v are denoted $N^l(v)$.

3.2. Independent Cascade model

The IC model is one of the most widely used diffusion models [11, 12, 15, 17, 18, 39]. Each node is either *active* or *inactive*, and activation is irreversible. Diffusion unfolds in discrete time steps. At $t = 0$ only the seed nodes in S are active. When a node u becomes active at step t , it has a single chance to activate each currently inactive neighbor v , succeeding with probability p_{uv} independently of all other attempts. If it succeeds, v becomes active at step $t + 1$. Only nodes activated at step t attempt activations at step $t + 1$. The process terminates when no further node can be activated [15].

3.3. The influence maximization problem

Let $A(S)$ be the (random) set of nodes that are active when the IC process started from S terminates. The influence spread of S is its expected size, $\sigma(S) = \mathbb{E}[|A(S)|]$. Given a budget k , the IM problem is to find

$$S^* = \operatorname{argmax}_{S \subseteq V, |S|=k} \sigma(S). \quad (1)$$

Under the IC model, $\sigma(\cdot)$ is monotone and submodular, so the greedy algorithm that repeatedly adds the node with the largest marginal gain $\sigma(S \cup \{v\}) - \sigma(S)$ achieves $\sigma(S) \geq (1 - 1/e) \sigma(S^*)$ when σ is evaluated exactly [15, 16]. Exact evaluation of σ is #P-hard [23], however, so it is usually estimated by MC simulation, which is the main computational bottleneck.

3.4. Expected Diffusion Value

To avoid simulation, Jiang et al. [19] proposed the Expected Diffusion Value, a one-hop approximation of $\sigma(S)$:

$$\text{EDV}(S) = k + \sum_{i \in N(S)} (1 - (1 - p)^{\tau(i)}), \quad (2)$$

where $k = |S|$, p is the propagation probability, $N(S)$ is the set of neighbors of the nodes in S excluding S itself, and $\tau(i) = |N(i) \cap S|$ (more precisely, the number of in-neighbors of i that belong to S) is the number of seeds that can directly attempt to activate i . The term $1 - (1 - p)^{\tau(i)}$ is the exact probability that i is activated *in the first diffusion step*. EDV therefore counts the seeds plus the expected number of their direct neighbors activated at step one. When p is small, multi-hop activations are rare and EDV is a close and inexpensive surrogate for σ .

4. THE PROPOSED METHOD: HNOF

4.1. Rationale

HNOF rests on two observations. The first concerns *where* good candidates are. A node v at distance two from an active seed s shares at least one mutual friend u with s . If both s and v are seeds, u receives two independent activation attempts and is activated in the first step with probability $1 - (1 - p)^2 \approx 2p$ rather than p . This exploits the reinforcing role of mutual friends emphasized in the social-contagion literature [40]. At the same time, v is not a direct friend of s . Most of its own neighborhood therefore lies outside the area already covered by s , which limits the redundancy that arises when adjacent hubs are selected together (the rich-club effect) [34]. Direct friends of a seed, by contrast, are already partially covered: selecting them wastes part of their own activation probability and adds mainly overlapping coverage.

The second observation concerns *how* to score candidates. Because σ is expensive to evaluate, HNOF uses the marginal gain of EDV, which can be computed in closed form from local information.

4.2. Algorithm description

The general framework of HNoF is shown in Figure 1. It consists of six steps: (1) initial selection, (2) identification of candidate nodes for the current seed, (3) filtering of candidates, (4) computation of the expected diffusion value, (5) seed selection and (6) the termination check.

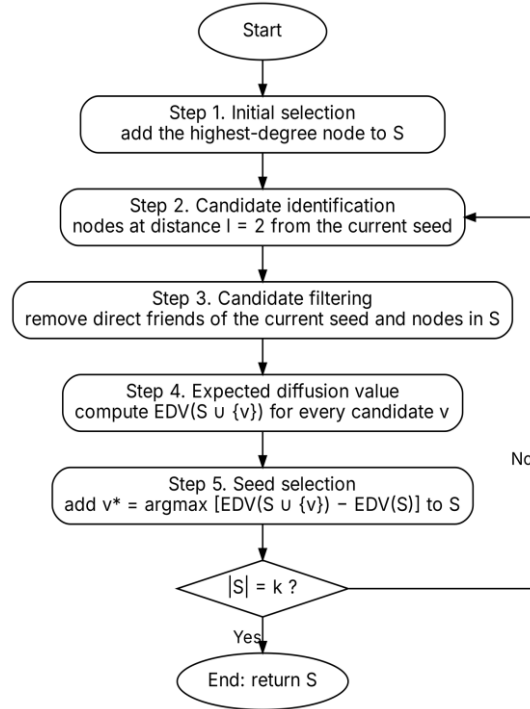


Fig.1. General framework of the HNoF algorithm.

Step 1: Initial selection. All nodes are sorted in descending order of degree, and the node with the highest degree is added to S as the first seed. This node also becomes the *current seed*.

Step 2: Candidate identification. The nodes located at distance $l = 2$ from the current seed are identified as candidates. These are the *neighbors of the friends* of the current seed. As explained in Section 4.1, if a node is active and a neighbor of its friend is also active, their mutual friend is very likely to be influenced. For this reason only neighbors of friends of the current seed are allowed to become candidates and take part in the computation at this step.

Step 3: Candidate filtering. A node found in Step 2 may be both a neighbor of a friend of the current seed and a direct friend of that seed, for example when the seed, a friend and the node form a triangle. Such nodes, as well as nodes already in S , are removed, so that the candidate set becomes

$$C = \left(\bigcup_{u \in N(s_{cur})} N(u) \right) \setminus (N(s_{cur}) \cup S). \quad (3)$$

Step 4: Expected diffusion value. For each remaining candidate $v \in C$, $EDV(S \cup \{v\})$ is computed with Equation (2). Rather than recomputing Equation (2) from scratch, the marginal gain can be obtained directly. Adding v to S increases the seed count by one. If v was already a neighbor of S , its first-step activation term is no longer counted. Each neighbor u of v receives one more activation attempt, which raises its term from $1 - (1 - p)^{\tau(u)}$ to $1 - (1 - p)^{\tau(u)+1}$. Hence

$$\Delta(v | S) = EDV(S \cup \{v\}) - EDV(S) = 1 - \mathbb{1}[v \in N(S)](1 - (1 - p)^{\tau(v)}) + p \sum_{u \in N(v) \setminus S} (1 - p)^{\tau(u)} \quad (4)$$

Equation (4) makes the behavior of HNoF transparent. A candidate gains p for every neighbor that is not yet covered ($\tau(u) = 0$) and a *geometrically discounted* amount $p(1 - p)^{\tau(u)}$ for every neighbor that is already covered. If the candidate itself is already a neighbor of the seed set, it is additionally penalized. Overlap with existing coverage is thus discouraged without any MC simulation.

Step 5: Seed selection. In the manner of General Greedy, the candidate that maximizes the marginal influence $\Delta(v | S)$ is added to S and becomes the new current seed.

Step 6: Termination check. Steps 2–5 are repeated until $|S| = k$. If the candidate set of the current seed becomes empty before k seeds have been selected (for instance, in a small component or when all two-hop nodes have already been chosen or excluded), the candidate search restarts from the earliest seed in S , then from the next one, and so on, in insertion order. After termination, the influence spread of the final seed set is evaluated by MC simulation under the IC model (Section 5.3). This simulation is used only for evaluation, not for selection.

Algorithm 1. gives the pseudocode.

Algorithm 1. HNoF

Input : graph $G = (V, E)$, propagation probability p , budget k

Output: seed set S , $|S| = k$

```

1:  $s \leftarrow \operatorname{argmax}_{\{v \in V\}} d(v)$ ;  $S \leftarrow \{s\}$ ;  $s_{\text{cur}} \leftarrow s$  ▷ Step 1
2: initialise  $\tau(u) \leftarrow 1$  for  $u \in N(s)$ , else 0
3: while  $|S| < k$  do
4:    $C \leftarrow (\cup_{u \in N(s_{\text{cur}})} N(u)) \setminus (N(s_{\text{cur}}) \cup S)$  ▷ Steps 2–3
5:   if  $C = \emptyset$  then
6:      $s_{\text{cur}} \leftarrow$  next seed of  $S$  in insertion order; continue
7:   for each  $v \in C$  do compute  $\Delta(v | S)$  by Eq. (4) ▷ Step 4
8:    $v^* \leftarrow \operatorname{argmax}_{\{v \in C\}} \Delta(v | S)$  ▷ Step 5
9:    $S \leftarrow S \cup \{v^*\}$ ;  $s_{\text{cur}} \leftarrow v^*$ 
10:  for each  $u \in N(v^*)$  do  $\tau(u) \leftarrow \tau(u) + 1$  ▷ update coverage
11: return  $S$  ▷ Step 6

```

4.3. Illustrative example

Figure 2 illustrates one iteration of HNoF, in which the third seed is selected. Suppose that $S = \{9, 6\}$, where node 9 (the highest-degree node, $d = 7$) was selected in Step 1 and node 6 is the current seed. The direct friends of node 6 are $N(6) = \{1, 2, 7, 13, 14\}$. Their neighbors at distance two from node 6 are $\{3, 8, 9, 15, 20\}$. Removing the seed 9 yields the candidate set $C = \{3, 8, 15, 20\}$, marked in red in Figure 2(a). Panels (b)–(e) show the neighborhoods that enter the EDV computation for each of the four candidates. With $p = 0.01$, Equation (4) gives $\Delta(15 | S) \approx 1.0496$, $\Delta(8 | S) \approx 1.0396$, $\Delta(20 | S) \approx 1.0298$ and $\Delta(3 | S) \approx 1.0196$. Node 15 is not adjacent to any seed and has five neighbors, four of which are already covered once (8, 7, 16, 14) and one of which is uncovered (21). It therefore yields the largest marginal gain. Node 8 has a comparable number of neighbors, but it is itself a neighbor of seed 9 and is penalized accordingly. As shown in Figure 2(f), node 15 is added to S as the third seed and becomes the current seed for the next iteration.

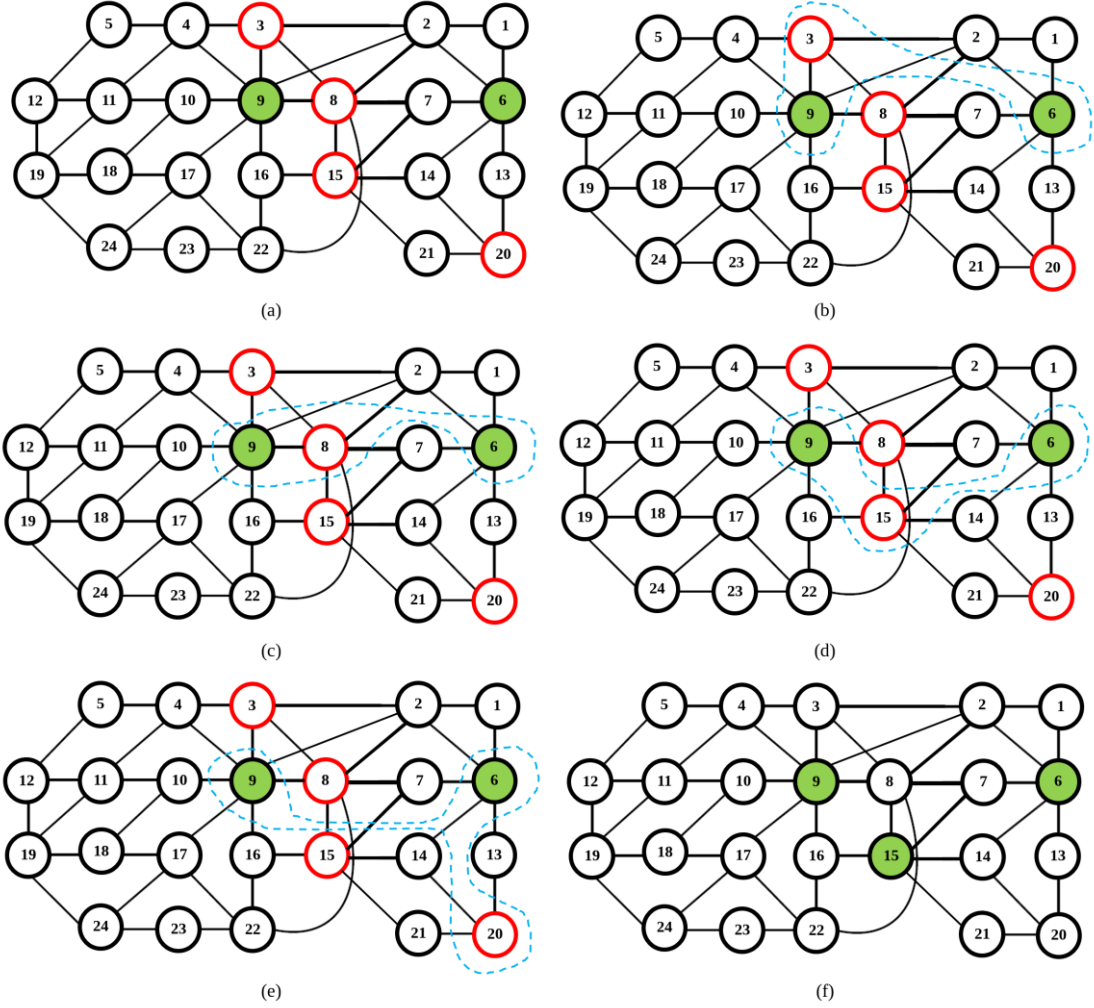


Fig. 2. An illustrative iteration of HNoF. (a) Seeds (green) and candidate nodes (red) for current seed 6; (b)–(e) EDV evaluation for each candidate; (f) node 15 selected as the third seed.

4.4. Computational complexity

Step 1 requires $O(n \log n)$ time for sorting, or $O(n)$ to find the maximum-degree node. At iteration t , Steps 2–3 enumerate at most $\sum_{u \in N(s \text{ cur})} d(u) \leq d_{\max}^2$ two-hop nodes. Using Equation (4) with maintained τ values, Step 4 evaluates each candidate v in $O(d(v))$ time, and the update in line 10 costs $O(d(v^*))$. The total cost of seed selection is therefore

$$O\left(n + \sum_{t=1}^k \sum_{v \in C_t} d(v)\right) \subseteq O(n + k d_{\max}^3), \quad (5)$$

which is independent of the number of MC rounds and, for networks with moderate maximum degree, much smaller than the $O(k n R \cdot m)$ worst-case cost of General Greedy. If $\text{EDV}(S \cup \{v\})$ is instead recomputed from scratch with Equation (2), each evaluation costs $O(\sum_{u \in S \cup \{v\}} d(u))$. This cost grows with $|S|$, and candidate sets around hubs can be large in heavy-tailed networks. The memory requirement is $O(n + m)$ for the graph and the τ array.

5. EXPERIMENTAL SETUP

HNoF was evaluated on four real-world datasets and compared with representative algorithms from three families, in terms of influence spread and the running time required to select the k seeds of S . All algorithms were run in the same simulation environment.

5.1. Datasets

The datasets differ in scale, density and type of relationship (Table 2). NetHEPT is the arXiv collaboration network from the “High Energy Physics – Theory” section, covering papers from 1991 to 2003; each node is an author and each edge indicates co-authorship of at least one paper [22]. NetPHY is the arXiv co-authorship network from the “Physics” section, with the same node and edge semantics [22]. Epinions1 is the who-trusts-whom network of the consumer-review site Epinions.com; each node is a member and each directed edge represents a trust relation [41]. Slashdot0902 is the network of the technology-news site Slashdot, collected in February 2009, in which users tag one another as friends or foes [42]. The last two datasets were obtained from the Stanford Large Network Dataset Collection [43].

Table 2. Statistics of the datasets used in the experiments.

Dataset	Type	Nodes	Edges	Average degree ^a
NetHEPT	Undirected, collaboration	15,233	58,891	7.73
NetPHY	Undirected, collaboration	37,149	231,507	12.46
Epinions1	Directed, trust	75,879	508,837	6.71
Slashdot0902	Directed, friend/foe	82,168	948,464	11.54

^a $2m/n$ for undirected networks; average out-degree m/n for directed networks.

5.2. Baseline algorithms and parameter settings

HNoF was compared with General Greedy (a simulation-based greedy method), LIR (a heuristic), and SAEDV and DDSE (metaheuristics). The parameter settings are listed in Table 3. They follow the settings reported in the original studies or common practice.

Table 3. Parameter settings of the compared algorithms.

Algorithm	Family	Parameters
General Greedy [15]	Simulation-based greedy	MC rounds $R = 10,000$; $p = 0.01$; IC model
LIR [32]	Heuristic	Nodes with $LI = 0$ selected by descending degree; spread evaluated with $R = 10,000$, $p = 0.01$
SAEDV [19]	Metaheuristic (simulated annealing)	Random initial solution; initial temperature $T_0 = 500,000$; final temperature $T_f = 100,000$; cooling decrement $\Delta T = 2,000$; inner-loop iterations $q = 10$; EDV fitness
DDSE [8]	Metaheuristic (differential evolution)	Crossover rate $CR = 0.4$; mutation rate $F = 0.1$; diversity threshold = 0.6; maximum iterations $MaxIt = 200$; EDV fitness
HNoF (proposed)	Greedy heuristic	Candidate distance $l = 2$; $p = 0.01$; EDV marginal gain (Eq. 4)

General Greedy was not run on Epinions1 and Slashdot0902 because its running time on these networks was prohibitive (Section 6.2).

5.3. Evaluation metrics and protocol

Two metrics were used, consistent with the objective of HNoF to achieve high spread with acceptable running time. Influence spread is the expected number of nodes activated by the k initially active nodes in S under the IC model with $p = 0.01$. For every algorithm and every $k \in \{5, 10, \dots, 50\}$, it was estimated as the average over $R = 10,000$ MC simulations, so that all algorithms are evaluated with the same estimator. Running time is the wall-clock time, in seconds, required to find the final $k = 50$ nodes of S .

5.4. Experimental environment

All experiments were conducted on a machine running Windows 10 with an Intel® Core™ i5-5200U processor (2.20 GHz) and 8 GB of RAM.

6. RESULTS

6.1. Influence spread

Figure 3 shows the influence spread of the compared algorithms on the four datasets. The x-axis is the number of initially active nodes, from $k = 5$ to $k = 50$, and the y-axis is the resulting spread. Table 4 reports the values at $k = 50$.

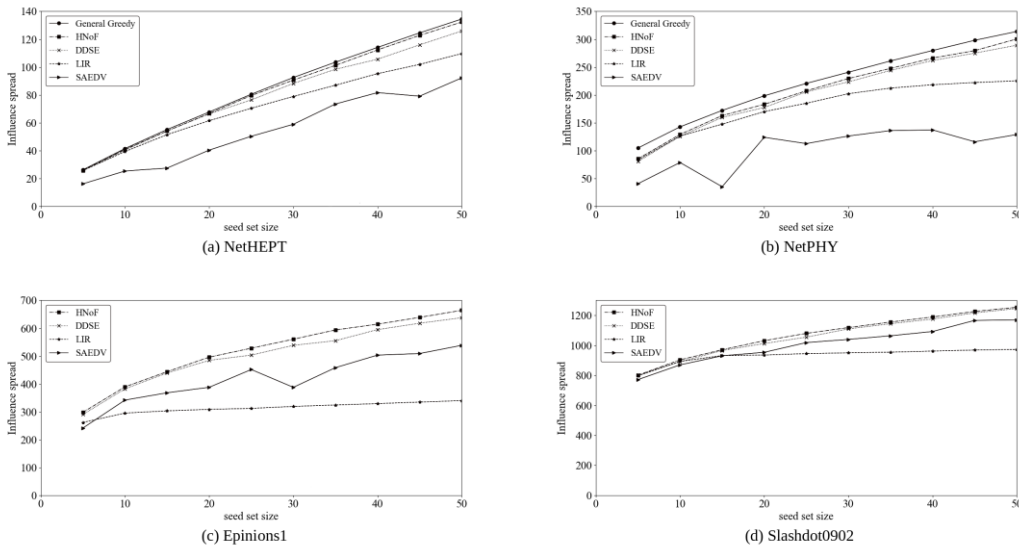


Fig. 3. Influence spread under the IC model with $p = 0.01$ on (a) NetHEPT, (b) NetPHY, (c) Epinions1 and (d) Slashdot0902.

NetHEPT (Figure 3a). General Greedy achieves the largest spread, followed very closely by HNoF; the two curves almost coincide over the entire range of k (≈ 134 vs. ≈ 133 at $k = 50$). DDSE, LIR and SAEDV follow in that order. DDSE is close to HNoF for $k = 5-20$ but falls progressively behind for $k = 20-50$ (≈ 125 at $k = 50$).

NetPHY (Figure 3b). General Greedy is again the best, followed by HNoF, DDSE, LIR and SAEDV. The gap between HNoF and General Greedy is largest for small k (≈ 86 vs. ≈ 105 at $k = 5$) and narrows to about 4% at $k = 50$ (≈ 301 vs. ≈ 314). SAEDV performs very poorly on this dataset, and its curve is non-monotonic in k .

Epinions1 (Figure 3c). Since General Greedy was not run, HNoF achieves the largest spread (≈ 663 at $k = 50$), followed by DDSE (≈ 640), SAEDV (≈ 540) and LIR. LIR saturates early at around 340 nodes.

Slashdot0902 (Figure 3d). HNoF again attains the largest spread ($\approx 1,255$ at $k = 50$), closely followed by DDSE ($\approx 1,245$), and then SAEDV ($\approx 1,170$) and LIR (≈ 975), whose curve flattens after $k \approx 15$.

6.2. Running time

Running time, the time required to find the k seed nodes of S , is shown in Figure 4 on a logarithmic scale for $k = 50$ (MC rounds $R = 10,000$ where applicable). Across all datasets the ranking from fastest to slowest is SAEDV, DDSE, HNoF, LIR and General Greedy. General Greedy required on the order of 10^6 seconds even on the two smallest networks ($\approx 1.4 \times 10^6$ s on NetHEPT and $\approx 4.6 \times 10^6$ s on NetPHY). HNoF required ≈ 45 s and $\approx 2.8 \times 10^2$ s, respectively, a reduction of about four orders of magnitude. On NetHEPT and NetPHY the running times of DDSE and HNoF are of the same order (≈ 33 s vs. ≈ 45 s and $\approx 1.3 \times 10^2$ s vs. $\approx 2.8 \times 10^2$ s). On the larger Epinions1 and Slashdot0902 networks the gap widens: HNoF is roughly 7 and 13 times slower than DDSE, respectively, while remaining about 2 and 3.5 times faster than LIR.

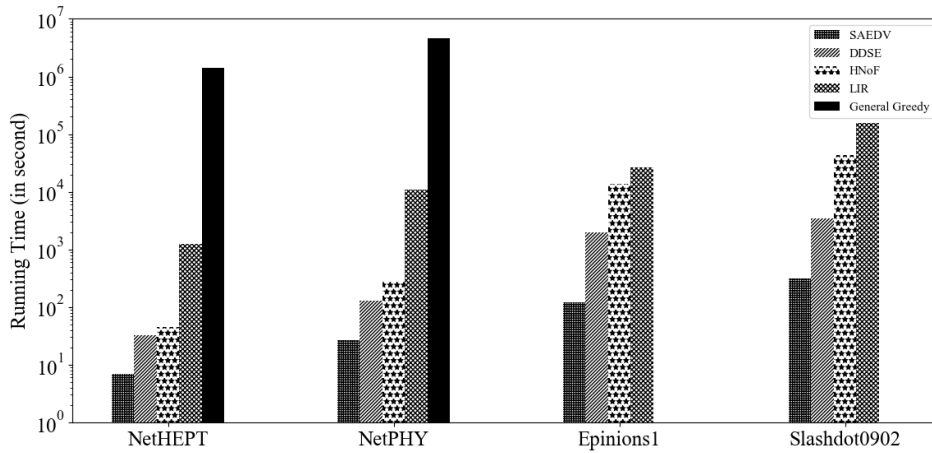


Fig. 4. Running time (seconds, logarithmic scale) of the compared algorithms for $k = 50$.

Table 4. Influence spread at $k = 50$ and running time of the compared algorithms. Values are read from Figures 3 and 4 and rounded; “—” indicates that the algorithm was not run.

Dataset	Metric	General Greedy	HNoF	DDSE	LIR	SAEDV
NetHEPT	Spread	≈ 134	≈ 133	≈ 125	≈ 110	≈ 92
	Time (s)	$\approx 1.4 \times 10^6$	≈ 45	≈ 33	$\approx 1.3 \times 10^3$	≈ 7
NetPHY	Spread	≈ 314	≈ 301	≈ 290	≈ 226	≈ 129
	Time (s)	$\approx 4.6 \times 10^6$	$\approx 2.8 \times 10^2$	$\approx 1.3 \times 10^2$	$\approx 1.1 \times 10^4$	≈ 27
Epinions1	Spread	—	≈ 663	≈ 640	≈ 342	≈ 540
	Time (s)	—	$\approx 1.4 \times 10^4$	$\approx 2.0 \times 10^3$	$\approx 2.7 \times 10^4$	$\approx 1.2 \times 10^2$
Slashdot0902	Spread	—	$\approx 1,255$	$\approx 1,245$	≈ 973	$\approx 1,170$
	Time (s)	—	$\approx 4.5 \times 10^4$	$\approx 3.5 \times 10^3$	$\approx 1.6 \times 10^5$	$\approx 3.2 \times 10^2$

6.3. Spread–efficiency trade-off

Taken together, Figures 3 and 4 locate HNoF on the efficient frontier of the spread–time trade-off on all four datasets. No compared method obtains a larger spread in less time. General Greedy obtains slightly more spread only at a cost four orders of magnitude higher. DDSE and SAEDV run faster but obtain less spread, and LIR is both slower and less effective. The size of the advantage is dataset-dependent. HNoF’s spread advantage over DDSE is largest on NetHEPT ($\approx 6\%$ at $k = 50$) and smallest on Slashdot0902 ($< 1\%$), whereas its running-time disadvantage relative to DDSE is smallest on NetHEPT and largest on Slashdot0902.

7. DISCUSSION

7.1. Interpretation of the results

Why HNoF approaches General Greedy on collaboration networks. At $p = 0.01$ the IC process is strongly subcritical on all four networks: the expected number of activations per edge attempt is small, and most of a seed's influence is exerted on its immediate neighborhood. In this regime the one-hop EDV is a close surrogate for σ , so greedy maximization of EDV marginal gains closely mimics greedy maximization of σ . This explains why HNoF almost coincides with General Greedy on NetHEPT. Collaboration networks are also highly clustered, because co-authors of a paper form cliques. Equation (4) penalizes candidates whose neighborhoods are already covered, and the two-hop restriction steers selection away from the dense cliques around existing seeds. The larger gap on NetPHY at small k is consistent with this interpretation. NetPHY has a higher average degree, so multi-hop cascades, which EDV ignores, contribute relatively more to σ for the first few high-degree seeds. As k grows, overlap avoidance becomes more important than raw reach, and the gap narrows.

Why HNoF outperforms DDSE and SAEDV. All three algorithms optimize the same objective (EDV). The differences therefore stem from the search strategy rather than the objective. SAEDV starts from a random solution and explores the solution space stochastically. With a finite cooling schedule (200 temperature levels $\times$ 10 inner iterations), it cannot reliably find high-EDV solutions in large spaces. This is reflected in its poor spread on NetPHY and in its non-monotonic curves, since SAEDV re-solves the problem independently for each k , so a larger budget does not guarantee a superset of the previous solution. DDSE uses a degree-descending initialization and an evolutionary search, which explains its competitiveness. Its population-based search, however, still samples only part of the space. HNoF evaluates *every* candidate in a structurally relevant subset at each step and deterministically selects the best one. Its spread curves are therefore smooth and monotone, and it outperforms DDSE consistently, if modestly.

Why LIR saturates. LIR selects nodes that are local degree maxima ($LI = 0$) in descending order of degree. In heavy-tailed directed networks such as Epinions1 and Slashdot0902, after the first few hubs the remaining $LI = 0$ nodes are either low-degree peripheral nodes or nodes whose neighborhoods heavily overlap with those of the hubs. This is a form of the rich-club effect [33, 34]. LIR has no mechanism to account for this overlap, so its marginal spread vanishes and its curve becomes flat. HNoF's explicit overlap penalty in Equation (4) prevents this saturation.

Why HNoF is slower than DDSE on the large networks. According to Equation (5), the cost of HNoF depends on the size of the two-hop neighborhoods it explores, which grows quadratically with the degree of the current seed. In Epinions1 and Slashdot0902, seeds selected by HNoF are often hubs with large two-hop neighborhoods, so each iteration evaluates many candidates. DDSE's cost is governed by its population size and iteration limit, which do not grow with node degree. The observed widening of the running-time gap from NetHEPT to Slashdot0902 is consistent with this analysis. It also suggests that efficient incremental evaluation (Equation 4), candidate sampling, or capping the candidate set for very high-degree seeds would substantially reduce HNoF's running time without changing its selection logic.

7.2. Relation to previous studies

The near-optimality of greedy EDV maximization in the low- p regime agrees with Jiang et al. [19], who found that EDV correlates closely with simulated spread for small propagation probabilities, and with Chen et al. [22], who showed that degree-discount heuristics, which also reason one hop away, perform well in this regime. The improvement of HNoF over DDSE complements Cui et al. [8]. Using the same objective, a deterministic, structurally restricted greedy search can match or exceed an evolutionary search, at the cost of more evaluations on hub-dominated graphs. The saturation of LIR is in line with the motivation of RNR [33] and HybridRank [34], which were proposed precisely to address the rich-club weakness of local-index ranking. The emphasis HNoF places on neighbors of friends is conceptually related to the role of mutual friends and wide bridges in social contagion [40]. It shows that this sociological insight can be translated into an effective search-space restriction for IM.

7.3. Theoretical implications

This study contributes to the IM literature in three ways. First, it shows that a *structural* restriction of the candidate set, based on hop distance and friendship, can be combined with a greedy criterion without a large loss of spread in sparse diffusion regimes. Candidate pruning is therefore a viable alternative to global ranking. Second, the closed-

form EDV marginal gain (Equation 4) makes explicit an overlap-penalty mechanism that is implicit in the EDV-based metaheuristics. It also provides a principled link between degree-discount heuristics, which apply a discount per covered neighbor, and simulation-based greedy algorithms. Third, the results characterize *when* one-hop analytical surrogates are adequate (low p , clustered networks) and when they are not (multi-hop cascades, small k on denser graphs), which can guide the choice of estimator in future algorithm design.

7.4. Practical implications

For practitioners in viral marketing, information campaigns or public-health messaging, HNoF offers a deterministic and parameter-light method: apart from p and k , only the candidate distance $l = 2$ must be set. It yields reproducible seed sets of high quality, which is valuable when campaign decisions must be audited or justified. On networks with tens of thousands of nodes, it selects 50 seeds in seconds to minutes on commodity hardware, compared with days to weeks for General Greedy. When the network is very large and dominated by hubs and time is the binding constraint, a faster metaheuristic such as DDSE may be preferable, at the cost of a small reduction in spread. The choice between the two methods can therefore be guided by the relative value of additional reach versus computation time.

7.5. Limitations and future research

Several limitations should be considered when interpreting the results. (1) HNoF does not provide an approximation guarantee. EDV ignores multi-hop activations, and the two-hop restriction may exclude the globally best candidate. (2) Experiments were limited to the IC model with a uniform probability $p = 0.01$. Performance under the weighted-cascade and Linear Threshold models, under heterogeneous or learned probabilities, and at larger p , where multi-hop cascades matter more, remains to be investigated. (3) The baselines did not include state-of-the-art reverse-influence-sampling algorithms (e.g., IMM, TIM) or lazy-evaluation greedy methods (CELF, CELF++), which are the standard benchmarks for spread and efficiency. (4) The stochastic baselines (SAEDV, DDSE) were reported from single runs; repeated runs with mean and dispersion, and statistical comparison (e.g., Wilcoxon signed-rank tests across k values), would make the comparisons more robust. (5) Running times depend on implementation language and data structures, and all algorithms were measured on a single commodity machine. (6) The largest network contains fewer than 10^6 edges, so scalability to networks with hundreds of millions of edges was not demonstrated. (7) The sensitivity of HNoF to the candidate distance l and to the choice of the first seed was not examined.

Future work should address these points by: incorporating CELF-style lazy evaluation and incremental τ updates; capping or sampling candidate sets around hubs; extending the candidate definition to l -hop neighborhoods with adaptive l ; adapting HNoF to the Linear Threshold model and to community-aware, budgeted, competitive and time-critical IM variants; and benchmarking against IMM and TIM on billion-edge graphs.

8. CONCLUSION

The influence maximization problem in social networks seeks k nodes with the greatest influence over a short period. To balance the number of finally activated nodes against the time needed to find the seeds, this paper proposed HNoF, a heuristic that selects seeds greedily using an analytical estimate of the expected diffusion value of candidate nodes. The key idea is to consider as candidates only the nodes that are neighbors of the friends of the current seed. Identifying these nodes and involving only them in the spread computation substantially reduces the number of evaluations and increases the speed and scalability of the algorithm. We also derived a closed-form EDV marginal gain that explains how HNoF favors new coverage and penalizes overlap. Experiments on four real-world networks under the IC model showed that HNoF attained spread within about 1–4% of General Greedy on the collaboration networks, at a running time roughly four orders of magnitude lower, and achieved the highest spread among the scalable baselines on all datasets. Its running time was higher than that of DDSE and SAEDV, particularly on large hub-dominated networks, which identifies a clear direction for future optimization. Overall, HNoF provides a simple, deterministic and effective option for practitioners who need high-quality seed sets at acceptable computational cost.

Authors' Contributions

All authors contributed to the conception and design of the study, the interpretation of the results and the preparation of the manuscript. All authors read and approved the final manuscript.

Declaration of Generative AI Use

During the preparation of this work, the authors used generative AI tools to improve the language and readability of the manuscript. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgments

The authors are grateful to all those who contributed to this project.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Ethics Considerations

Not applicable. This study used publicly available, anonymized network datasets and did not involve human participants or the collection of personal data.

Data Availability

The network datasets used in this study are publicly available from their original sources. The source code of HNoF and the experimental scripts are available from the corresponding author upon reasonable request.

REFERENCES

- [1] Talukder, A., et al. (2019). Knapsack-based reverse influence maximization for target marketing in social networks. *IEEE Access*, 7, 44182–44198. <https://doi.org/10.1109/ACCESS.2019.2908412>
- [2] He, Q., et al. (2019). TIFIM: A two-stage iterative framework for influence maximization in social networks. *Applied Mathematics and Computation*, 354, 338–352. <https://doi.org/10.1016/j.amc.2019.02.056>
- [3] Li, X., Smith, J. D., Dinh, T. N., & Thai, M. T. (2019). TipTop: (Almost) exact solutions for influence maximization in billion-scale networks. *IEEE/ACM Transactions on Networking*, 27(2), 649–661. <https://doi.org/10.1109/TNET.2019.2898413>
- [4] Domingos, P., & Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 57–66). Association for Computing Machinery. <https://doi.org/10.1145/502512.502525>
- [5] Wu, K. (2015). Influence maximization in social networks [Master's thesis, Concordia University].
- [6] Banerjee, S., Jenamani, M., & Pratihari, D. K. (2019). ComBIM: A community-based solution approach for the budgeted influence maximization problem. *Expert Systems with Applications*, 125, 1–13.

<https://doi.org/10.1016/j.eswa.2019.01.070>

- [7] Chang, T.-C., et al. (2019). Seed selection and social coupon allocation for redemption maximization in online social networks. 2019 IEEE 35th International Conference on Data Engineering (ICDE). <https://doi.org/10.1109/ICDE.2019.00044>
- [8] Cui, L., Hu, H., Yu, S., Yan, Q., Ming, Z., Wen, Z., & Lu, N. (2018). DDSE: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. *Journal of Network and Computer Applications*, 103, 119–130. <https://doi.org/10.1016/j.jnca.2017.12.003>
- [9] Shang, J., Zhou, S., Li, X., Liu, L., & Wu, H. (2017). CoFIM: A community-based framework for influence maximization on large-scale networks. *Knowledge-Based Systems*, 117, 88–100. <https://doi.org/10.1016/j.knosys.2016.09.029>
- [10] Shang, J., et al. (2018). IMPC: Influence maximization based on multi-neighbor potential in community networks. *Physica A: Statistical Mechanics and Its Applications*, 512, 1085–1103. <https://doi.org/10.1016/j.physa.2018.08.045>
- [11] Singh, S. S., et al. (2019). MIM2: Multiple influence maximization across multiple social networks. *Physica A: Statistical Mechanics and Its Applications*, 526, Article 120902. <https://doi.org/10.1016/j.physa.2019.04.138>
- [12] Wang, W., & Street, W. N. (2018). Modeling and maximizing influence diffusion in social networks for viral marketing. *Applied Network Science*, 3, Article 6. <https://doi.org/10.1007/s41109-018-0062-7>
- [13] Wu, H., et al. (2018). LAIM: A linear time iterative approach for efficient influence maximization in large-scale networks. *IEEE Access*, 6, 44221–44234. <https://doi.org/10.1109/ACCESS.2018.2864240>
- [14] Zhu, W., et al. (2019). Location-based seeds selection for influence blocking maximization in social networks. *IEEE Access*, 7. <https://doi.org/10.1109/ACCESS.2019.2900708>
- [15] Kempe, D., Kleinberg, J., & Tardos, É. (2003). 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). Association for Computing Machinery. <https://doi.org/10.1145/956750.956769>
- [16] Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1), 265–294. <https://doi.org/10.1007/BF01588971>
- [17] Li, Y., Fan, J., Wang, Y., & Tan, K.-L. (2018). Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10), 1852–1872. <https://doi.org/10.1109/TKDE.2018.2807843>
- [18] Peng, S., Zhou, Y., Cao, L., Yu, S., Niu, J., & Jia, W. (2018). Influence analysis in social networks: A survey. *Journal of Network and Computer Applications*, 106, 17–32. <https://doi.org/10.1016/j.jnca.2018.01.005>
- [19] Jiang, Q., Song, G., Cong, G., Wang, Y., Si, W., & Xie, K. (2011). Simulated annealing based influence maximization in social networks. In *Proceedings of the twenty-fifth AAAI conference on artificial intelligence* (pp. 127–132). AAAI Press. <https://doi.org/10.1609/aaai.v25i1.7838>
- [20] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the thirteenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 420–429). Association for Computing Machinery. <https://doi.org/10.1145/1281192.1281239>

- [21] Goyal, A., Lu, W., & Lakshmanan, L. V. S. (2011). CELF++: Optimizing the greedy algorithm for influence maximization in social networks. In Proceedings of the twentieth international conference companion on World Wide Web (pp. 47–48). Association for Computing Machinery. <https://doi.org/10.1145/1963192.1963217>
- [22] Chen, W., Wang, Y., & Yang, S. (2009). Efficient influence maximization in social networks. In Proceedings of the fifteenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 199–208). Association for Computing Machinery. <https://doi.org/10.1145/1557019.1557047>
- [23] Chen, W., Lakshmanan, L. V. S., & Castillo, C. (2013). Information and influence propagation in social networks. Morgan & Claypool Publishers.
- [24] Cheng, S., Shen, H., Huang, J., Zhang, G., & Cheng, X. (2013). StaticGreedy: Solving the scalability-accuracy dilemma in influence maximization. In Proceedings of the 22nd ACM international conference on Information and knowledge management (pp. 509–518). Association for Computing Machinery. <https://doi.org/10.1145/2505515.2505541>
- [25] Borgs, C., Brautbar, M., Chayes, J., & Lucier, B. (2014). Maximizing social influence in nearly optimal time. In Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms (pp. 946–957). Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9781611973402.70>
- [26] Tang, Y., Xiao, X., & Shi, Y. (2014). Influence maximization: Near-optimal time complexity meets practical efficiency. In Proceedings of the 2014 ACM SIGMOD international conference on Management of data (pp. 75–86). Association for Computing Machinery. <https://doi.org/10.1145/2588555.2593670>
- [27] Tang, Y., Shi, Y., & Xiao, X. (2015). Influence maximization in near-linear time: A martingale approach. In Proceedings of the 2015 ACM SIGMOD international conference on Management of data (pp. 1539–1554). Association for Computing Machinery. <https://doi.org/10.1145/2723372.2723734>
- [28] Narayanam, R., & Narahari, Y. (2011). A Shapley value-based approach to discover influential nodes in social networks. IEEE Transactions on Automation Science and Engineering, 8(1), 130–147. <https://doi.org/10.1109/TASE.2010.2052042>
- [29] Wang, C., Chen, W., & Wang, Y. (2012). Scalable influence maximization for independent cascade model in large-scale social networks. Data Mining and Knowledge Discovery, 25(3), 545–576. <https://doi.org/10.1007/s10618-012-0262-1>
- [30] Goyal, A., Lu, W., & Lakshmanan, L. V. S. (2011). SIMPATH: An efficient algorithm for influence maximization under the linear threshold model. In Proceedings of the 2011 IEEE 11th international conference on Data mining (pp. 211–220). IEEE. <https://doi.org/10.1109/ICDM.2011.132>
- [31] Morone, F., Min, B., Bo, L., Mari, R., & Makse, H. A. (2016). Collective influence algorithm to find influencers via optimal percolation in massively large social media. Scientific Reports, 6, Article 30062. <https://doi.org/10.1038/srep30062>
- [32] Liu, D., Jing, Y., Zhao, J., Wang, W., & Song, G. (2017). A fast and efficient algorithm for mining top-k nodes in complex networks. Scientific Reports, 7, Article 43330. <https://doi.org/10.1038/srep43330>
- [33] Rui, X., et al. (2019). A reversed node ranking approach for influence maximization in social networks. Applied Intelligence, 49(7), 2684–2698. <https://doi.org/10.1007/s10489-018-01398-w>
- [34] Ahajjam, S., & Badir, H. (2018). Identification of influential spreaders in complex networks using HybridRank algorithm. Scientific Reports, 8, Article 11932. <https://doi.org/10.1038/s41598-018-30310-2>

- [35] Chen, W., Lin, T., Tan, Z., Zhao, M., & Zhou, X. (2016). Robust influence maximization. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 795–804). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939745>
- [36] Xie, G., et al. (2019). MBIC: A novel influence propagation model for membership-based influence maximization in social networks. IEEE Access, 7, 75696–75707. <https://doi.org/10.1109/ACCESS.2019.2922474>
- [37] Chen, Y.-C., Zhu, W.-Y., Peng, W.-C., Lee, W.-C., & Lee, S.-Y. (2014). CIM: Community-based influence maximization in social networks. ACM Transactions on Intelligent Systems and Technology, 5(2), Article 25. <https://doi.org/10.1145/2532549>
- [38] Qiu, L., et al. (2019). PHG: A three-phase algorithm for influence maximization based on community structure. IEEE Access, 7, 62511–62522. <https://doi.org/10.1109/ACCESS.2019.2912628>
- [39] Zhu, R., et al. (2018). QuickIM: Efficient, accurate and robust influence maximization algorithm on billion-scale networks [Preprint]. arXiv. arXiv:1805.12320.
- [40] Centola, D., & Macy, M. (2007). Complex contagions and the weakness of long ties. American Journal of Sociology, 113(3), 702–734. <https://doi.org/10.1086/521848>
- [41] Richardson, M., Agrawal, R., & Domingos, P. (2003). Trust management for the semantic web. In The Semantic Web—ISWC 2003 (Lecture Notes in Computer Science, Vol. 2870, pp. 351–368). Springer. https://doi.org/10.1007/978-3-540-39718-2_23
- [42] Leskovec, J., Lang, K. J., Dasgupta, A., & Mahoney, M. W. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Mathematics, 6(1), 29–123. <https://doi.org/10.1080/15427951.2009.10129177>
- [43] Leskovec, J., & Krevl, A. (2014). SNAP datasets: Stanford large network dataset collection [Data set]. Stanford University. <http://snap.stanford.edu/data>