TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/09 pp703–711
DOI: 10.26599/TST.2019.9010017
Volume 25, Number 6, December 2020

Approximating Special Social Influence Maximization Problems

Jie Wu* and Ning Wang

Abstract: Social Influence Maximization Problems (SIMPs) deal with selecting k seeds in a given Online Social Network (OSN) to maximize the number of eventually-influenced users. This is done by using these seeds based on a given set of influence probabilities among neighbors in the OSN. Although the SIMP has been proved to be NP-hard, it has both submodular (with a natural diminishing-return) and monotone (with an increasing influenced users through propagation) that make the problem suitable for approximation solutions. However, several special SIMPs cannot be modeled as submodular or monotone functions. In this paper, we look at several conditions under which non-submodular or non-monotone functions can be handled or approximated. One is a profit-maximization SIMP where seed selection cost is included in the overall utility function, breaking the monotone property. The other is a crowd-influence SIMP where crowd influence exists in addition to individual influence, breaking the submodular property. We then review several new techniques and notions, including double-greedy algorithms and the supermodular degree, that can be used to address special SIMPs. Our main results show that for a specific SIMP model, special network structures of OSNs can help reduce its time complexity of the SIMP.

Key words: influence maximization; online social networks; submodular function

1 Introduction

This section reviews the notion of the submodular function with associated properties. We discuss the general Social Influence Maximization Problem (SIMP) with a focus on the independent cascade model. We then introduce two special SIMPs.

1.1 Submodular functions

Many optimization problems in combinatorics, graphs, and game theory can be represented as non-negative submodular functions. A submodular function σ is a set function so that the difference in the incremental value of σ that an element makes when added to an input set S decreases as the size of the input set increases. That is to say, $\sigma(\cdot)$ is submodular with respect to S if $\sigma(S\cup\{v\})-\sigma(S) \leqslant \sigma(S'\cup\{v\})-\sigma(S')$ for $S'\subset S$. Submodular functions have a natural diminishing returns property that makes them suitable for approximation solutions of complex optimization problems.

More specifically, an optimization problem concerning a convex or concave function can be described as a problem of maximizing or minimizing a submodular function with or without constraints. The set cover problem (minimization without constraint) and maximum coverage problem (maximization with constraint) are both classic NP-complete problems: given a set of elements U and a collection of sets S, the set cover problem is to minimize the amount of subsets used to cover V. The maximum coverage problem is to identify the k elements of S whose union has the maximum cardinality. If we let $\sigma(S')$ denote the cardinality of S', a subset of S, then $\sigma(S')$ is submodular. $\sigma(S'\cup\{v\})-\sigma(S')$ is called the marginal gain of v (i.e., a subset of V) when it is added to S'. A greedy cover works by iteratively selecting a v with the maximum marginal gain (i.e., the maximum

• Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA. E-mail: jiewu@temple.edu.
• Ning Wang is with the Department of Computer Science, Rowan University, Glassboro, NJ 08028, USA. E-mail: wangn@rowan.edu.
* To whom correspondence should be addressed.
Manuscript received: 2019-04-19; accepted: 2019-04-29

number of uncovered elements). This greedy cover is a $\ln n + 1$ and $1 - 1/e$ approximation of the set cover and maximum coverage problems, respectively. Note that in this example, σ is monotone, i.e., for every $S' \subseteq S$, we have $\sigma(S') \leqslant \sigma(S)$. That is, coverage does not diminish as more subsets are included.

However, in many optimization problems, the corresponding functions are not necessarily both submoduler and monotone. Let us look at two cases:

• **Submodular but non-monotone**: The max-cut problem is as follows: in the unweighted version, we are given an undirected graph, and our goal is to partition the graph into two node sets to maximize the number of edges crossing these two sets. In the weighted version, each edge has a non-negative weight, and our goal is to maximize the weight of the edges crossing these two sets.

• **Monotone but non-submodular**: The welfare maximization problem is as follows: there is a set of players and a set of indivisible items. Each player has his own (monotone, non-decreasing) valuation for any subset of items. Note that the valuation function can be non-submodular. The goal is to distribute the items to the players in a way that maximizes social welfare (the sum of values of all players) by their personal valuations.

Figure 1 shows an example of the max-cut problem in a given graph using white and black nodes. If the black node set grows from an empty set, the cut value increases to a certain point and then decreases. The cut function is symmetric since the role of black and white nodes can be exchanged. More formally, σ is symmetric if for every $S \subset V$ we have $\sigma(S) = \sigma(V - S)$. An example to show the non-submodularity of the welfare maximization problem is that the value of a pair of shoes is much larger than the sum of the individual value of each shoe for a customer.

1.2 Social influence maximization problem

Motivated by applications for viral marketing[1] and personalized recommendations[2], research into the social influence propagation has received tremendous attention in the last decade, especially for the SIMP in Online Social Networks (OSNs). The original SIMP was proposed by Kempe et al.[3] The SIMP aims to select k initially-influenced seed users to maximize the number of eventually-influenced users. Under the independent cascade and linear threshold models, the SIMP has been proven to be NP-hard, monotone, and submodular. Consequently, a simple greedy algorithm that iteratively maximizes the marginal gain, obtains an approximation ratio of $1 - 1/e$ to the optimal algorithm.

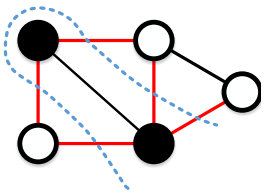


Fig. 1 An example of max-cut with black and white nodes.

1.3 Independent cascade model of SIMP

The independent cascade model[3] is a classic model that simulates influence propagations in OSNs. The network can be modeled as a directed graph, $G = (V, E)$, where the nodes V represent users and the edges E represent the connections among users. For each edge there is a weight w representing the influence propagation probability $(0 \leqslant w \leqslant 1)$. The influence spread process starts with a set S of nodes. All nodes in S are initially active and are also called seed users[3]. In contrast, all other nodes are initially inactive. The independent cascade unfolds in discrete steps according to the following randomized process. When a node v first becomes influenced, it has a single chance to activate its neighbors who are not yet influenced with a probability of w. If an inactive node has received multiple activation attempts, these activation attempts can be sequenced in an arbitrary order. If an inactive node is successfully activated in step t, it then becomes active in step $t+1$. Whether or not an activation attempt succeeds, it has no further impacts in subsequent steps. The above process iterates step-by-step and terminates when no more activations are possible. We use $\sigma(S)$ to denote the expected number of eventually-influenced nodes. $\sigma(S)$ is also called the influence spread of S.

1.4 Two special SIMPs

Some variations of the SIMP are not submodular and monotone. Reference [4] considers a SIMP variation that is not monotone, but symmetric: instead of specifying k seed users, the objective is changed to a maximization problem without constraints. The profit of a seed set, S, denoted as σ', is given as the influence spread ($\sigma(S)$ minus the cost of selection ($c(S)$), i.e., $\sigma'(S) = \sigma(S) - c(S)$. The corresponding independent cascade model is called profit maximization. This problem is submodular but not monotone because the

marginal profit gained by adding a new seed can be negative.

This paper considers another SIMP variation which is not submodular but monotone due to the phenomenon of crowd influence in addition to individual single seed influence. Figure 2, which shows three people (Alice, Bob, and Charlie), provides an example of both crowd and single influence. Directed edges represent the influence from Alice or Bob on Charlie. The influences Charlie receives from Alice or Bob are independent of each other. According to crowd psychology, if both Alice and Bob are influenced, there should exist a crowd influence in addition to Alice's and Bob's influences. Figure 2 shows how a combined influence on Charlie is calculated using both individual influence and the crowd influence from Alice and Bob. A hyperedge (of a hypergraph) is used to depict such a crowd influence. Note that influences through hyperedges are not submodular since seed user selections in the SIMP are no longer diminishing returns. Consequently, solving the SIMP in hypergraphs poses unique challenges. The first challenge is to deal with non-submodularity. The problem hardness and approximability both need to be explored. New algorithms are needed, since a simple greedy algorithm can no longer guarantee an approximation ratio. Another challenge is scalability. Since hyperedges change the scalability of the SIMP, it is difficult to reduce their complexities.

1.5 Overview

In this paper, we discuss both the profit-maximization SIMP and the crowd-influence SIMP. Both are recent results extended from known theoretical results. Section 2 focuses on a solution for profit-maximization SIMP which is based on the idea of double-greedy algorithms[5]. Section 3 discusses a new notion of supermodular degree in a solution for the crowd-influence SIMP[6]. Section 4 reviews relevant work. Section 5 shows some simulation results of several solutions to the crowd-influence SIMP[7]. The paper concludes in Section 6.

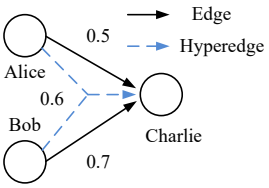
2 Profit-Maximization SIMP

This section starts with a special algorithmic approach called double-greedy and then uses this approach to find an approximation solution for the profit-maximization SIMP.

2.1 Double-greedy algorithms

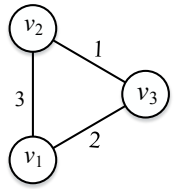
Double-greedy algorithms has been recently proposed in Ref. [5] to solve unconstrained submodular maximization functions. Consider a non-negative submodular function σ. Consider the complement of σ, denoted by $\bar{\sigma}$, defined as $\bar{\sigma}(S) : \sigma(V/S)$ for any $S \subseteq V$. Since σ is submodular, $\bar{\sigma}$ is also submodular. Given an optimal solution $S' \subseteq V$ with input σ, V/S' is an optimal solution for $\bar{\sigma}$. σ starts from an empty set and iteratively adds elements greedily. $\bar{\sigma}$ starts from V and iteratively removes elements greedily. Correlated execution on both σ and $\bar{\sigma}$ is applied and the searching dimension reduced by one after each iteration. In the i-th iteration, the deterministic double-greedy algorithm either adds v_i to set S or removes v_i from V. The decision is conducted greedily based on the marginal gain of σ and $\bar{\sigma}$, denoted as δ and $\bar{\delta}$, respectively.

In Fig. 3, we show an example of applying the deterministic double greedy to the weighted max-cut problem, where $\sigma(S)$ and $\bar{\sigma}(V)$ are the cut value with a set S and V, respectively. The decision result, i.e., whether v_1, v_2, and v_3 are in set S or V, is denoted by a 3-tuple. For example, $(1, 0, 0)$ means that only v_1 is in the set. Initially, $S = \varnothing$ and $V = \{v_1, v_2, v_3\}$. v_1 is



Influenced?		Probability to propagate the influence to Charlie
Alice	Bob	
Yes	No	0.5
No	Yes	0.7
Yes	Yes	1–(1–0.5)(1–0.6)(1–0.7)=0.94

Fig. 2 Social influences through edges and hyperedges.



Iteration	S	σ	V	$\bar{\sigma}$
0	(0,0,0)	0	(1,1,1)	0
v_1	(1,0,0)	5	(1,1,1)	0
v_2	(1,0,0)	5	(1,0,1)	4
v_3	(1,0,0)	5	(1,0,0)	5

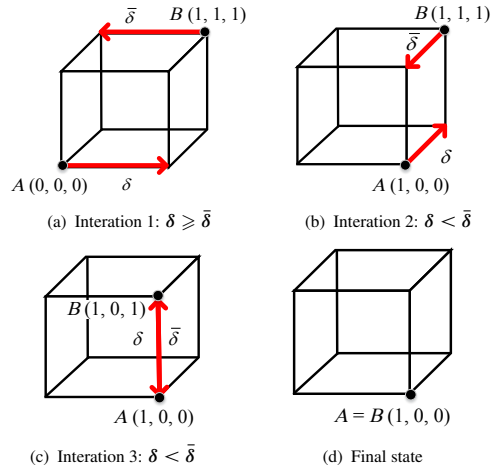
Fig. 3 Max-cut illustration.

added to set S in the first iteration since $(\delta = 5) \geqslant (\bar{\delta} = 5)$. Similarly, v_2 is deleted during the second iteration because $(\delta = -2) < (\bar{\delta} = 4)$. Finally, v_3 is deleted during the third iteration since $(\delta = -1) < (\bar{\delta} = 1)$; V equals S, and the algorithm terminates. There is a geometric interpretation shown in Fig. 4. The deterministic double-greedy algorithm is proved to have an approximation ratio of $1/3$[5]. A randomized double-greedy algorithm is further proposed by changing the greedy decision to a "smoother" decision. That is, in each iteration, v_i is added to S with a probability of $\delta/(\delta + \bar{\delta})$; otherwise, v_i is removed from V. The randomized double-greedy algorithm can improve the approximation ratio to $1/2$[5]. Note that the running time of the double-greedy algorithm is $O(|V|)$.

2.2 Profit-maximization SIMP

The major difference between the profit-maximization SIMP and existing works is that there is a seed selection cost for each node. The profit of a seed set S, denoted by $\sigma'(S)$, is given as the influence spread $(\sigma(S))$ minus the cost of selection $(c(S))$, i.e., $\sigma'(S) = \sigma(S) - c(S)$. $\sigma'(\cdot)$ is a submodular function but might not monotone anymore.

We can apply the double-greedy algorithm to the profit-maximization SIMP. One limitation of the double-greedy algorithms is that they only establish approximation guarantees for non-negative submodular functions. For a general cost function $c(\cdot)$, $\sigma'(\cdot)$ can be negative for some node sets. To address this problem, Tang et al.[4] extended the double-greedy algorithms by adding an iterative pruning procedure to reduce the search space without losing optimality. Their idea is



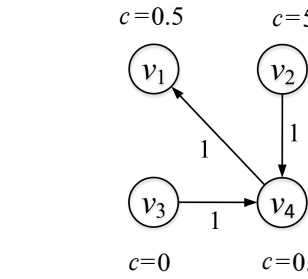
(a) Interation 1: $\delta \geqslant \bar{\delta}$ (b) Interation 2: $\delta < \bar{\delta}$

(c) Interation 3: $\delta < \bar{\delta}$ (d) Final state

Fig. 4 Geometric interpretation of a deterministic double-greedy algorithm.

to find the nodes that must be selected as seeds and eliminate the nodes that are impossible to be chosen as seeds in an optimal solution. By the submodularity of the influence metric, the marginal profit for adding a new seed node decreases with the base seed set. Thus, the smallest possible profit gain of adding this seed node is generated by adding the node into an universal set except itself, i.e., $S_1 = \{v : \sigma'(v|V/v) > 0\}$. The largest possible profit gain is produced by adding a node into an empty seed set, i.e., $S'_1 = \{v : \sigma'(v|\varnothing) \geqslant 0)\}$. Therefore, nodes in S_1 must be selected in the optimal solution since adding nodes in S_1 to any set can further increase the profit. Similarly, nodes outside S'_1 cannot be selected for the optimal set since adding nodes in S'_1 will definitely decrease the profit. After finding a must-select node, i.e., v, we can iteratively reduce the search space by $S_i = \{v : \sigma'(v|S'_{i-1}/v) > 0\}$ and $S'_i = \{v : \sigma'(v|S_{i-1}) \geqslant 0)\}$ until the pruning procedure is converged, i.e., $S_i = S_{i-1}$ and $S'_i = S'_{i-1}$.

Figure 5 shows a toy example of iterative pruning in the profit-maximization SIMP. To simplify the influence spread calculation of $\sigma(\cdot)$, we assume that all edge weights are 1. The cost of selecting $V = \{v_1, v_2, v_3, v_4\}$ as a seed node is $\{0.5, 5, 0.5, 0\}$, respectively. Before iterative pruning, the solution space is the power set of $\{v_1, v_2, v_3, v_4\}$, i.e., any combination of these four nodes. If we denote $S^\star$ as the optimal seed set, then $S_0 \subseteq S^\star \subseteq S'_0$, where $S_0 = \varnothing$ and $S'_0 = V$. Then, we check every node in S'_0 by adding each to S_0 and calculating the corresponding marginal profit increase. In the first round, we find that even adding v_2 into the empty solution set, i.e., $\sigma'(v_2|\varnothing)$, will lead to a profit increase of -2. v_1, v_2, and v_4 will be influenced but the cost of selecting v_2 is 5. Therefore, v_2 cannot belong to the optimal solution. However, we can always remove v_2 to get a better result. Then, we update $S'_1 =$



Round 0: $S_0 = \{\}$, $S'_0 = \{v_1, v_2, v_3, v_4\}$
Round 1: $S_1 = \{v_3\}$, $S'_1 = \{v_1, v_3, v_4\}$
Round 2: $S_2 = \{v_3\}$, $S'_2 = \{v_1, v_3, v_4\}$

Fig. 5 An illustration of iterative pruning.

$\{v_1, v_3, v_4\}$. On the other hand, we find that adding v_3 to the seed set $\{v_1, v_2, v_4\}$ can further increase the total profit by 1, i.e., $\sigma'(v_3|S'_0/v_3) = 1$, and v_3 must be selected in the optimal solution. Then, we update S_0 to S_1, i.e., $S_1 = \{S_0 \cup v_3\}$. As a result, we shrink the solution space to be $S_1 \subseteq S^\star \subseteq S'_1$. Similarly, we can continue to prune the solution space based on S_1 and S'_1. However, we cannot further reduce the solution space, i.e., $S_2 = S_1$ and $S'_2 = S'_1$ and the pruning ends in this example.

Tang et al.[4] proved that applying the iterative pruning approach prior to applying double-greedy algorithms will maintain the same approximation guarantee, i.e., 1/2, with the condition where $\sigma'(S_\star) + \sigma'(S'_\star) > 0$, and $S_\star$ and $S'_\star$ are the final node sets after iterative pruning.

3 Crowd-influence SIMP

This section reviews the notion of supermodular degree and applies it to solutions for the crowd-influence SIMP.

3.1 Supermodular degree

Considering crowd-influence, influence propagation is not submodular with respect to S, meaning that $\sigma(S \cup \{v\}) - \sigma(S) > \sigma(S' \cup \{v\}) - \sigma(S')$ for $\exists\, v \in V, S' \subset S, S \subseteq V$. Supermodular degree is proposed in Ref. [6] to evaluate the degree to which a function violates the submodular function.

Specifically, we define the modularity set of a node v as M_v, which is a set of nodes including all nodes that might increase the marginal gain of v. The supermodular degree of a node v is the cardinality of the corresponding modularity set. An example is shown in Fig. 6. The corresponding modularity set of v_1 is $\{v_2\}$. Similarly, The corresponding modularity set of v_2 is $\{v_1\}$. This is because v_1 and v_2 together can influence all the remaining nodes. We call v_1 and v_2 boosting nodes for each other. In general, the modularity of v is denoted as M_v. The supermodular degree, denoted as Δ, is the maximum supermodular degree of any v, i.e., $\Delta = \max_v |M_v|$. In Fig. 6, we have $\Delta = 1$. The supermodularity comes from the non-diminishing return effects of influence propagation through hyperedges, since a single node cannot activate the hyperedge. Note that a submodular function's supermodular degree is 0.

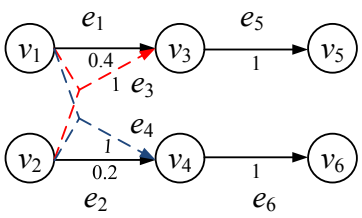


Fig. 6 Supermodular in social influences.

For supermodular problems, the traditional greedy algorithm, which iteratively selects a node to maximize the marginal gain, may not work well. Figure 6, in which we want to select two seed nodes to influence the remaining nodes, shows such an example. The edge weights of $\{e_1, e_2, e_3, e_4, e_5, e_6\}$ are $\{0.4, 0.2, 1, 1, 1, 1\}$. Note that edges e_3 and e_4 are hyperedges. The traditional greedy algorithm has two iterations: it selects v_3 in the first iteration and v_4 in the second iteration. This is because the influence gain of selecting v_1 is $0.4 \times 1 + 0.4 \times 1 \times 1 = 0.8$, which is the expected active probability for nodes v_3 and v_5. The influence gain of selecting v_4 is 1×1, which is the active probability of the node v_6. Similarly, the influence gain of selecting v_2, v_3, v_5, and v_6 individually is $0.4, 1, 0$, and 0, respectively. As a result, the nodes v_3 and v_4 are selected in the first and second rounds, i.e., $\sigma(\{v_3, v_4\}) = 2$. However, a better idea is to select a set of nodes rather than a single one in one greedy iteration. In Fig. 6, if we select $\{v_1, v_2\}$ in one greedy iteration, the result can be significantly improved. It is because the active probability of nodes v_3 and v_4 are $1 - (1 - 0.4)(1 - 1) = 1$ and $1 - (1 - 0.2)(1 - 1) = 1$, respectively. Nodes v_3 and v_4 can further propagate influence to nodes v_5 and v_6 with a probability of 1. Therefore, v_1 and v_2 together can influence all the other nodes, leading to $\sigma(\{v_1, v_2\}) = 4$.

3.2 Crowd-influence SIMP

The SIMP under the independent cascade model in hypergraphs is proven to be NP-hard, and cannot be approximated within a ratio of $|V|^{\epsilon-1}$ for any $\epsilon > 0$. $|V|$ is the number of nodes in the hypergraph, meaning that no algorithm can do better than a random seed user selection in terms of the asymptotic approximation ratio. Recent advances in network science show that user connections in OSNs are not truly random[8]. The degree distribution in OSNs is acknowledged to follow the power-law distribution[8]: a majority of users are inactive with a small number of connections, while a minority of users are active with a large number of connections. Based on Kumar et al.[9], OSNs are known to have small diameters (about 6), high clustering coefficients (larger than 0.1), and community structures. These structural properties can be incorporated into

algorithmic designs. Zheng et al.[7] proved that the supmodular degrees, denoted as Δ, of most OSNs have the following property $\lim_{|V|\to\infty}\dfrac{\Delta}{O(|V|)}=0$, i.e., $\Delta\in O(|V|)$.

Zheng et al.[7] leveraged the structural properties of OSNs to solve the non-submodular SIMP in hypergraphs, and two approximation algorithms[6] are applied with ratios of $1/(\Delta+2)$ and $1-\mathrm{e}^{-\frac{1}{\Delta+1}}$. In the existing Naive Greedy (NG) seed node selection shown in Algorithm 1, the node (with the maximum marginal gain, i.e., which increases the overall influence maximally) is selected at a time. In contrast, the first greedy algorithm (called Improved Greedy (IG) in Algorithm 2) includes the selected node together with a subset of its boosting nodes at each round, subject to the limit of the total number of seed nodes allowed. The second greedy algorithm (called Capped Greedy (CG) in Algorithm 3) makes two major changes. First, every node, together with a subset of its boosting set, is selected as the first round of seeds. Then the subsequent rounds of seeds together with their boosting subsets are selected based on maximum marginal gain. Second, the size of the seed node together with its boosting subset is controlled through another round of iterations from 1 to Δ (supermodular degree). The detailed algorithm is shown in Algorithms 2 and 3, respectively. The key ideas of these three algorithms are illustrated in Fig. 7. The major contribution in Ref. [7] is showing that the supermodular degree is bound in OSNs. Thus, the optimization technique using supermodular degree can be applied to OSN-related problems.

Algorithm 1 NG

Input: a hypergraph G and a constant k.
Output: a set of seed nodes S, initiated $\varnothing$.

while $|S|<k$ **do**
  Find $v=\arg\max_{v\in V}(\sigma(S\cup\{v\})-\sigma(S))$.
  Update $S=S\cup\{v\}$.
end while

Algorithm 2 IG

Input: a hypergraph G and a constant k.
Output: a set of seed nodes S, initiated $\varnothing$.

while $|S|<k$ **do**
  Find $\arg\max_{v\in V,M'_v\subseteq M_v}(\sigma(S\cup\{v\}\cup M'_v)-\sigma(S))$, s.t. $|S\cup\{v\}\cup M'_v|\leqslant k$.
  Update $S=S\cup\{v\}\cup M'_v$.
end while

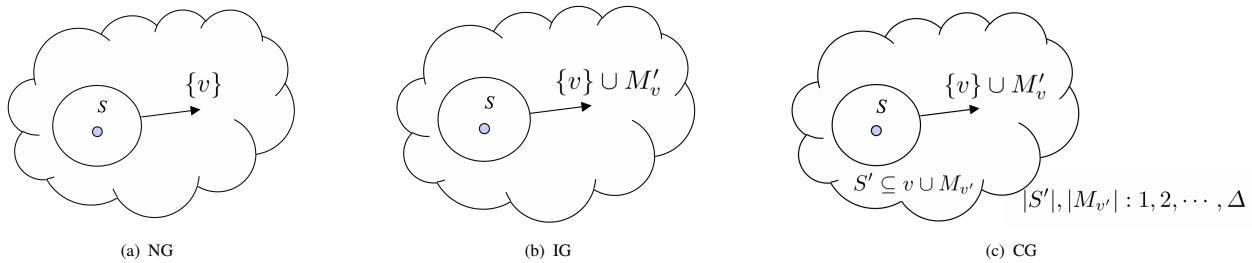
Algorithm 3 CG

Input: a hypergraph G and a constant k.
Output: a set of seed nodes S, initiated $\varnothing$.

for each $v'\in V$ **do**
  for each Δ' from 1 to Δ **do**
    for each $S'\subseteq\{v'\}\cup M_{v'}$, s.t. $|S'|\leqslant\min\{k,\Delta'\}$ **do**
      while $|S'|<k$ **do**
        Find $\arg\max_{v\in V,M'_v\subseteq M_v}(\sigma(S'\cup\{v\}\cup M'_v)-\sigma(S'))$, s.t. $|S'\cup\{v\}\cup M'_v|\leqslant k$ and $M'_v\leqslant\Delta'$.
        Update $S'=S'\cup\{v\}\cup M'_v$.
      end while
      if $\sigma(S')>\sigma(S)$ **then**
        Update $S=S'$.
      end if
    end for
  end for
end for

4 Related Work

Optimizing submodular functions can be classified on two axes: constrained / unconstrained and maximization / minimization. For unconstrained submodular minimization problems, we can use the Lovász extension[10] to get a convex closure, and thus, optimally solve them. Constrained submodular optimization problems have different approximation ratios based on different constraints. For example, there is an approximation ratio of 2 for the vertex cover[11] constraint. For an unconstrained submodular maximization problem, the double-greedy algorithm achieves a tight $1/2$ approximation ratio according to Feige et al.[12]



(a) NG (b) IG (c) CG

Fig. 7 Key ideas of three algorithms.

Unfortunately, there are some problems that are not simple non-negative submodular functions. For example, the minimum submodular cover problem with linear cost[13], negative submodular functions[4], and non-submodular problems, e.g., Refs. [6, 14–16], are all more complex. Hung et al.[14] studied a variation of the SIMP with multiple items. Their problem is NP-hard and non-submodular, and thus, only heuristic algorithms are provided because the problem of non-submodular function maximization[6] has not been perfectly solved in Ref. [17]. Although the problem of supermodular function maximization can be optimally solved by the minimum-norm-point algorithm[15], non-submodular functions are not the same. The latest approach is based on the curvature[16], which assumes that the marginal gain of the non-submodular function varies within a given curvature. This paper can be viewed as a curvature-based approach that is specially designed for the SIMP in OSNs.

Recent studies in network science show that many networks exhibit special structures and thus, the handy information of network structure can be applied in the network optimization. OSNs are scale-free networks[8], meaning that the degree distribution in an OSN follows the power-law distribution[18]. The supermodular degree analysis of OSNs in this paper is based on the OSNs' scale-free structure. Zheng and Wu[19] further found that the publish/subscribe systems[20] based on unstructured P2P networks have the nested scale-free architectures. The 'nested' indicates that the scale-free architecture is preserved when low-degree peers and their associated connections are removed.

5 Performance Evaluation

This section shows some simulation results[7] using three datasets and thoroughly compares the details performance of three solutions for the crowd-influence SIMP on one dataset: citation network.

5.1 Dataset validation

Our experiments are based on three datasets (Forum, Board, and Citation) from Tore Opsahl[21]. Forum records user activities in a forum with different topics. Board records directors belonging to the boards of some companies. Citation records collaborations among paper authors. Figure 8 shows the network topology in the Citation network, where each node is an author and the edge weight sum of joint papers. It shows the multiple clustered groups that correspond to different subdomains of expertise.

Figures 9 and 10 show the distributions of node degree (d_v) and modularity set cardinality ($|M_v|$). For the above three datasets, flags (triangles, circles, and squares) represent the real distributions based on statistics, and lines (dotted, dashed, and solid) are the fitting curves. Figure 9 validates the power-law distribution. It can be seen that the fraction of nodes with hyperdegree d is proportional to $d^{-\gamma}$ in each of

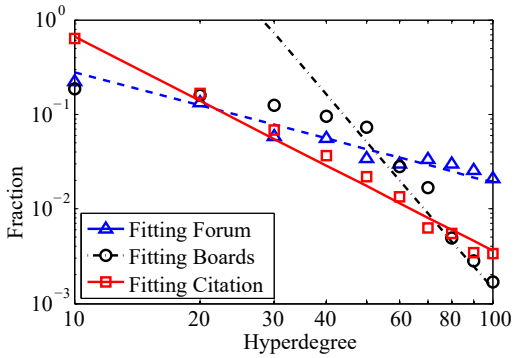


Fig. 9 Distribution of d_v among v.

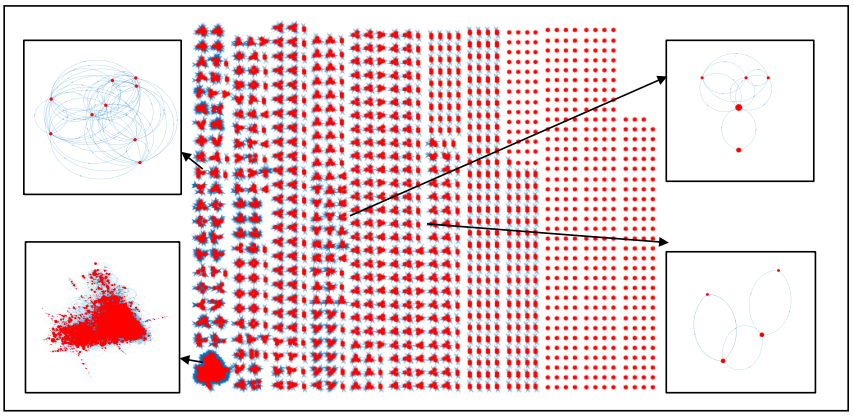


Fig. 8 Citation network topology.

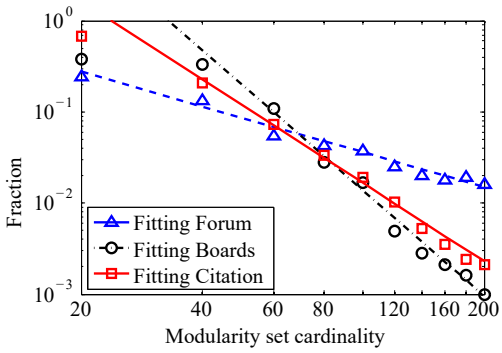


Fig. 10 Distribution of $|M_v|$ among v.

these three datasets. The distribution of the modularity set cardinality also follows the power-law, as shown in Fig. 10. However, the power-law exponents for d_v and $|M_v|$ may not be the same in a given dataset as they represent two different notions. Figure 10 further shows that only a small fraction of nodes have modularity sets with cardinalities larger than 100, which is relatively small compared with the number of nodes in the Citation dataset: 16 726. Both figures are plotted in a log-log scale where the y-axis represents the fraction of nodes corresponding to the data in the x-axis.

5.2 Algorithm performance

This subsection focuses on evaluating the performances of proposed algorithms, in terms of maximizing the number of eventually-influenced users. The evaluation result in the Citation dataset is shown in Fig. 11. A larger result represents a better performance, since seed nodes could eventually influence more nodes on expectation. Among all the algorithms, CG achieves the best performances, while NG has the worst performances. This is simply because CG considers the impact of crowd influences. Compared to other algorithms, CG has at least 15% more eventually-influenced users. CG, IG, and NG become identical when only one seed node is selected (i.e., $k = 1$).

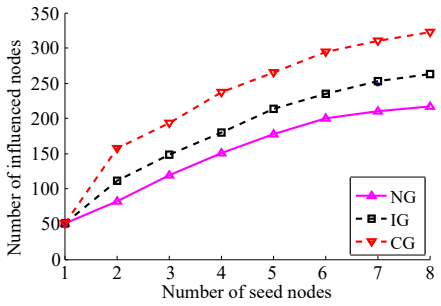


Fig. 11 Performance comparison of different algorithms.

6 Conclusion

Submodular function is an important property in solving combinatorial problems with bounded results. When submodule functions are non-negative and monotone, there are well-known approximation bounds. However, many real applications are not modeled as submodular and monotone functions. In this paper, we investigate two submodular function variations, i.e., non-monotone submodular functions and monotone non-submodular functions, in SIMP in OSNs. For the non-monotone submodular function, we use the existing double-greedy algorithm[5] to achieve an approximation of $1/2$ on expectation. The social influence maximization problem (called the profit-maximization SIMP), which is non-monotone, is discussed in Ref. [14]. For monotone non-submodular functions, we use the supermodular degree Δ to evaluate its violation of submodularity. We verify the structural properties of OSNs and show that the supermodular degree is bounded[7]. Furthermore, the two approximation algorithms for the other special social influene maximization problem (called the crowd-influence SIMP)[6] are applied with ratios of $1/(\Delta+2)$ and $1-\mathrm{e}^{-1/\Delta+1}$, respectively. Thus, the optimization technique using supermodular degrees can be applied to OSN-related problems with relatively low complexity.

Acknowledgment

This research was supported in part by the National Science Foundation (NSF) grants Computer and Network Systems (CNS) 1824440, CNS 1828363, CNS 1757533, CNS 1618398, CNS 1651947, and CNS 1564128.

References

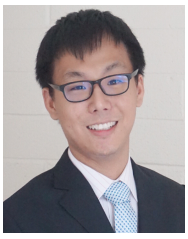
[1] H. Nguyen and R. Zheng, On budgeted influence maximization in social networks, *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1084–1094, 2013.

[2] X. Yang, H. Steck, and Y. Liu, Circle-based recommendation in online social networks, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*, Beijing, China, 2014, pp. 1267–1275.

[3] D. Kempe, J. Kleinberg, and É. Tardos, Maximizing the spread of influence through a social network, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*, Washington, DC, USA, 2003, pp. 137–146.

[4] J. Tang, X. Tang, and J. Yuan, Profit maximization for viral marketing in online social networks, in *Proceedings of the IEEE International Conference on Network Protocols (ICNP'16)*, Singapore, 2016. pp. 1095–1108.
[5] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, A tight linear time (1/2)-approximation for unconstrained submodular maximization, *SIAM Journal on Computing*, vol. 44, no. 5, pp. 1384–1402, 2015.
[6] M. Feldman and R. Izsak, Constrained monotone function maximization and the supermodular degree, arXiv preprint arXiv: 1407.6328, 2014.
[7] H. Zheng, N. Wang, and J. Wu, Non-submodularity and approximability: Influence maximization in online social networks, in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'19)*, Washington, DC, USA, 2019, pp. 1–10.
[8] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, Measurement and analysis of online social networks, in *Proceedings of the ACM SIGCOMM Conference on Internet Measurement (IMC'07),* San Diego, CA, USA, 2007, pp. 29–42.
[9] R. Kumar, J. Novak, and A. Tomkins, Structure and evolution of online social networks, in *Link Mining: Models, Algorithms, and Applications*. New York, NY, USA: Springer, 2010, pp. 337–357.
[10] L. Lovász, Submodular functions and convexity, in *Mathematical Programming the State of the Art*, Heidelberg, German: Springer, 1983, pp. 235–257.
[11] S. Iwata and K. Nagano, Submodular function minimization under covering constraints, in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS'09)*, Atlanta, GA, USA, 2009, pp. 671–680.
[12] U. Feige, V. S. Mirrokni, and J. Vondrak, Maximizing non-monotone submodular functions, *SIAM Journal on Computing*, vol. 40, no. 4, pp. 1133–1153, 2011.
[13] P.-J. Wan, D.-Z. Du, P. Pardalos, and W. Wu, Greedy approximations for minimum submodular cover with submodular cost, in *Computational Optimization and Applications*, Springer, 2010, pp. 463–474.
[14] H.-J. Hung, H.-H. Shuai, D.-N. Yang, L.-H. Huang, W.-C. Lee, J. Pei, and M.-S. Chen, When social influence meets item inference, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, San Francisco, CA, USA, 2016, pp. 915–924.
[15] S. Fujishige and S. Isotani, A submodular function minimization algorithm based on the minimum-norm base, presented at the Fourth Sino-Japanese Optimization Meeting, Tainan, China, 2008.
[16] M. Sviridenko, J. Vondrák, and J. Ward, Optimal approximation for submodular and supermodular optimization with bounded curvature, *Mathematics of Operations Research*, vol. 42, no. 4, pp. 1197–1218, 2017.
[17] S. Dughmi, Algorithmic information structure design: A survey, *ACM SIGecom Exchanges*, vol. 15, no. 2, pp. 2–24, 2017.
[18] T. Gradowski and A. Krawiecki, Majority-vote model on scale-free hypergraphs, *Acta Physica Polonica A*, vol. 127, no. 3A, pp. 1–4, 2015.
[19] H. Zheng and J. Wu, NSFA: Nested scale-free architecture for scalable publish/subscribe over p2p networks, in *Proceedings of the IEEE International Conference on Network Protocols (ICNP'16)*, Singapore, 2016, pp. 1–10.
[20] J. Leskovec, J. Kleinberg, and C. Faloutsos, Graph evolution: Densification and shrinking diameters, *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1–41, 2007.
[21] O. Tore, tnet dataset, https://toreopsahl.com/datasets/\#newman2001, 2001.

Jie Wu is the director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the director of International Affairs at College of Science and Technology. He served as chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and associate vice provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Service Computing*, *Journal of Parallel and Distributed Computing*, and *Journal of Computer Science and Technology*. Dr. Wu was general co-chair for *IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016*, and *IEEE CNS 2016*, as well as program cochair for *IEEE INFOCOM 2011* and *CCF CNCC 2013*. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a fellow of the AAAS and a fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.



Ning Wang received the PhD degree from Temple University in 2018. Before that, he received the BS degree from University of Electronic Science and Technology of China in 2013. He is currently an assistant professor in the Department of Computer Science, Rowan University, Glassboro, New Jersey, USA. His research focuses on mobile edge networks, data offloading, and pub/sub systems.