Theoretical Computer Science 764 (2019) 30–41

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Recommending links through influence maximization

Gianlorenzo D'Angelo [a,**], Lorenzo Severini [b,*], Yllka Velaj [c,*]

[a] Gran Sasso Science Institute (GSSI), Viale F. Crispi, 7, 67100, L'Aquila, Italy
[b] ISI Foundation, Via Chisola, 5, 10126, Torino, Italy
[c] University of Chieti-Pescara, Viale Pindaro, 42, 65127, Pescara, Italy

ARTICLE INFO

Article history:
Received 8 May 2017
Received in revised form 25 November 2017
Accepted 19 January 2018
Available online 2 April 2018

Keywords:
Approximation algorithm
Information diffusion
Complex networks
Independent cascade model
Network augmentation

ABSTRACT

The link recommendation problem consists in suggesting a set of links to the users of a social network in order to increase their social circles and the connectivity of the network. Link recommendation is extensively studied in the context of social networks and of general complex networks due to its wide range of applications. Most of the existing link recommendation methods estimate the likelihood that a link is adopted by users and recommend links that are likely to be established. However, most of such methods overlook the impact that the suggested links have on the capability of the network to spread information. Indeed, such capability is directly correlated with both the engagement of a single user and the revenue of online social networks.

In this paper, we study link recommendation systems from the point of view of information diffusion. In detail, we consider the problem in which we are allowed to spend a given budget to create new links so to suggest a bounded number of possible persons to whom become friend in order to maximize the influence of a given set of nodes. We model the influence diffusion in a network with the popular Independent Cascade model.

1. Introduction

Link recommendation is one of the most important features of online social networking sites. Recommending suitable connections to social network users has a twofold impact: it improves the user's experience by enlarging users' social circles and connectivity, and it increases social network revenue by enhancing the user engagement and retention rate.

Most of the existing link recommendation methods estimate the likelihood that a link is adopted by users and recommend links that are likely to be established [2,24–26]. In social networks such likelihood is estimated by considering the similarity of user profiles and some structural network properties. Friendship social networks, like Facebook, exploit similarity metrics that are based on the number of common neighbors. For example, the Friend-of-Friend (FoF) algorithm [25] recommends the users that have the highest number of common friends with the receiver of the recommendation. Other examples of such similarity metrics are the Adamic–Adar [1] index, the Jaccard's coefficient [15], and the preferential attachment index [3,28]. In content-centric social networks such as Twitter and Google+ the link recommender systems take also into account the similarity of the users' interests.

Most of the known methods aim at having a high accuracy in the prediction of the suggested links, without considering the impact of the new links on the capability of the network to spread information. This approach speeds up the network

* Corresponding author.
** Principal corresponding author.
E-mail addresses: gianlorenzo.dangelo@gssi.infn.it (G. D'Angelo), lorenzo.severini@isi.it (L. Severini), yllka.velaj@unich.it (Y. Velaj).

https://doi.org/10.1016/j.tcs.2018.01.017
0304-3975/

growth but is only able to infer links that will likely occur in the near future. Another drawback of the existing methods is that, in most of the cases, they suggest links to a short range of users and this does not necessarily lead to a network growth and an improved user engagement [35].

On the other hand, the capability of a social network to spread information is directly correlated with both the engagement of a single user and the revenue of an online social network [4]. From a user's perspective, being able to quickly and effectively disseminating information is highly desirable as it helps the user to share contents so to reach a large number of other users. This in turn helps the user to build its own social reputation, to express and diffuse its own opinion, and to discover novel contents and information. From the social network point of view, the effectiveness of the information spreading capabilities helps in improving the user engagement, which in turn increases the retention rate and the number of new subscriptions. Moreover, being able to quickly deliver diverse contents to a large portion of the users, increases the opportunities of making revenue from advertisement.

Most of the existing link recommendation systems overlook this aspect, which is crucial for both users and social networks. In this paper, we consider the problem of recommending links to a given set of users in a social network, without exceeding a given budget, in such a way that the number of users reached by the contents generated by such set of users is maximized. Our main objective is to improve the capability of the given users to diffuse their own contents, this in turn drives the network evolution towards an increment of the spreading capability of the whole network.

Several models of information diffusion have been introduced in the literature, two widely studied models are: the *Linear Threshold Model* (LTM) [13,17,31] and the *Independent Cascade Model* (ICM) [11,12,16,17]. In both models, we can distinguish between *active*, or *infected*, nodes which spread the information and inactive ones. At the beginning of the process a small percentage of nodes of the graph is set to active in order to let the information diffusion process start. Such nodes are called *seeds*. Recursively, currently infected nodes can infect their neighbors with some probability. After a certain number of such cascading cycles, a large number of nodes might becomes infected in the network. The process terminates when no further node gets activated. In this paper we adopt ICM to model the way in which the contents are propagated in the network.

1.1. Related works

The problem of recommending links to the users of a social network has been widely studied, we refer to [24] and [26] for surveys on the link recommendation and link prediction problems, respectively. The problem of recommending links by taking into account the information spreading capability, instead, has received little attention in the literature. In the following we focus on such problem and on the problems of modifying a graph in order to maximize or minimize the spread of information through a network under LTM and ICM models.

Yu et al. [35] propose a recommendation algorithm called ACR-FoF (algebraic connectivity regularized friends-of-friends) that uses the algebraic connectivity of a connected network to estimate its capability for spreading contents. The ACR-FoF takes also into account the success rate of the suggested links. The authors give experimental evidence that ACR-FoF improves the spread of contents in a social networks but do not prove any approximation guarantees. Chaoji et al. [4] consider a model in which each node is associated with an independent probability to share a content with all its neighbors and with a set of contents that are generated by the node itself. The problem they study consists in maximizing the expected number of nodes influenced by some of the contents by adding a set of edges under the constraint that each node has at most k incident new edges. They show that the problem is NP-hard and propose an information diffusion model called Restricted Maximum Probability Path Model in which a content is propagated between two users along the path with maximum probability among those containing a recommended edge. They show that under this model the objective function is submodular and hence the problem can be approximated within a constant bound. Li et al. [23] introduce the notion of user diffusion degree which is a measure of the influence that a user has and is computed by combining community detection algorithms with information diffusion models. They propose an algorithm that suggests links by combining the diffusion degree with the FoF algorithm and, by means of experiments on two networks, show that it outperforms some known baseline in terms of number of affected nodes under the ICM and LTM models.

To the best of our knowledge, under LTM, the graph modification problems that have been studied are those outlined in what follows. Khalil et al. [18] consider two types of graph modification, adding edges to or deleting edges from the existing network to minimize the information diffusion and they show that this network structure modification problem has a supermodular objective and therefore can be solved by algorithms with provable approximation guarantees. Zhang et al. [36] consider arbitrarily specified groups of nodes, and edge and node removal from the groups. They develop algorithms with rigorous performance guarantees and good empirical performance. Kimura et al. [20] use a greedy approach to delete edges under the LTM but do not provide any rigorous approximation guarantees. Kuhlman et al. [22] propose heuristic algorithms for edge removal under a simpler deterministic variant of LTM which is not only hard, but also has no approximation guarantee. Papagelis [29] and Crescenzi et al. [7] study the problem of augmenting the graph in order to increase the connectivity or the centrality of a node, respectively and experimentally show that this increases the expected number of eventual active nodes. Parotsidis et al. [30] study the problem of recommending links with the objective of improving the centrality of a node within a network. Cordasco et al. consider the problem of activating all the nodes in a network by adding links [6]. The authors study the Minimum Links Problem whose the aim is to find the minimum number of links that a set of external influencers should form with people in the network, in order to activate the entire social network. They prove that the Minimum Links problem cannot be approximated to within a ratio of $O(2^{log^{1-\epsilon} n})$ or any fixed $\epsilon \geq 0$

unless $NP \subseteq DTIME(n^{polylog})$, where n is the number of nodes in the network. They give linear time algorithms to solve the problem for trees, cycles, and cliques. For general graphs, instead, they design a polynomial time algorithm to compute size-efficient link sets that can activate the entire graph.

Under ICM, Wu et al. [34] consider graph modifications other than edge addition, edge deletion and source selection, such as increasing the probability that a node infects its neighbors. They proved that optimizing the selection of such modifications with a limited budget is NP-hard and is neither submodular nor supermodular. Sheldon et al. [32] study the problem of node addition to maximize the spread of information, and provide a counterexample showing that the objective function is not submodular. Kimura et al. [21] propose methods for efficiently finding good approximate solutions on the basis of a greedy strategy for the edge deletion problem under the ICM, but do not provide any approximation guarantees.

In [8] the authors introduced a preliminary version of the edge addition problem where new edges are incident to a given initial set of active nodes. They focus on the case in which the initial set of seeds is a singleton and investigate the existence of a constant approximation algorithm.

In this paper, we extend the results presented in [8]. In detail, we adopt the independent cascade model and investigate the problem of adding a limited number of edges incident to an arbitrary set of initial seeds, without exceeding a given budget k, in order to maximize the spread of information in terms of number of nodes that eventually become active. The problem we analyze differs from above mentioned ones since we make the reasonable restriction that the edges to be added can only be incident to the seed nodes and that to add such edges there is a cost to be paid. To our knowledge, similar problems have never been studied for the independent cascade model. We refer to this problem as the *Cost Influence Maximization with Augmentation problem* (CostIMA).

A generalization of the CostIMA problem has been proposed in [9] where the initial set of seed nodes is not given and the budget k is used for selecting the set of initial seeds and for buying a small number of edges incident to the seeds in order to maximize the expected number of nodes that become active. The results presented in [9] depend on those for the CostIMA problem since it has been used as a subroutine for the more general problem of seed selection and link addition.

1.2. Our results

We first focus on the unit-cost version of the problem that we call *Influence Maximization with Augmentation problem* (IMA). In such problem the cost of adding any edge is constant and equal to 1. We show that IMA is NP-hard to be approximated within a constant factor greater than $1 - (2e)^{-1}$ (Section 3.1). We then provide an approximation algorithm that almost matches such upper bound by guaranteeing an approximation factor of $1 - (e)^{-1} - \epsilon$, where ϵ is any positive real number (Section 3.2). The algorithm is based on a greedy technique and the approximation factor is proven by showing that the expected number of activated nodes is monotonically increasing and submodular with respect to the possible set of edges incident to the seeds.

Then, we study the more general CostIMA problem where we are given a budget k and the cost of edges is in $[0, 1]$. We propose an algorithm that combines greedy and enumeration techniques and that achieves an approximation guarantee of $1 - (e)^{-1}$, for any $\epsilon > 0$ (Section 4).

Both the IMA and CostIMA problems are interesting: even though the former represents a limited number of real scenarios, there is a greedy approximation algorithm that guarantees an approximation factor of $1 - (e)^{-1}$ exploiting the properties of submodular functions. The latter problem, instead, introduces a more flexible and general model for the application to link recommendation. Even if the greedy heuristic of IMA can not be trivially generalized to achieve the same approximation for the budgeted version of the problem, we will show how to obtain an approximation factor to $1 - (e)^{-1}$ by using the enumeration technique.

We observe that, even if the two proposed algorithms have the same worst-case computational complexity and approximation factor, the algorithm presented for the IMA problem is simpler because it does not require an enumeration technique and, therefore, it should perform better in practical scenarios.

2. Preliminaries

A social network is represented by a weighted directed graph $G = (V, E, p, c)$, where V represents the set of nodes, E represents the set of relationships, $p : V \times V \to [0, 1]$ is the propagation probability of an edge, that is the probability that the information is propagated from u to v if $(u, v) \in E$, and $c : V \times V \to [0, 1]$ is the cost of adding an edge to E.

In ICM, each node can be either *active* or *inactive*. If a node is active (or infected), then it is already influenced by the information under diffusion, if a node is inactive, then it is unaware of the information or not influenced. The process runs in discrete steps. At the beginning of the ICM process, few nodes are given the information, they are known as *seed nodes*. Upon receiving the information these nodes become active. In each discrete step, an active node tries to influence one of its inactive neighbors. The success of node u in activating the node v depends on the propagation probability of the edge (u, v), independently of the history so far. In spite of its success, the same node will never get another chance to activate the same inactive neighbor. The process terminates when no further nodes became activated from inactive state.

We define the influence of a set $A \subseteq V$ in the graph G, denoted $\sigma(A, G)$, to be the expected number of active nodes in G at the end of the process, given that A is the initial set of seeds. Given a set S of edges not in E, we denote by $G(S)$ the graph augmented by adding the edges in S to G, i.e. $G(S) = (V, E \cup S)$. We denote by $\sigma(A, S)$ the influence of A in $G(S)$.

In this paper, given a set of seeds A, we look for a set of edges S, to be added to G, incident to such seeds that maximize $\sigma(A, S)$. We assume that each edge $e \in (V \times V) \setminus E$ can be selected according to cost function c. In detail, the CostIMA problem is defined as follows: given a graph $G = (V, E)$, a budget k and a set A of seeds, find a set S of edges such that $S \subseteq (A \times V) \setminus E$, $c(S) \leq k$, and $\sigma(A, S)$ is maximum, where $c(S) = \sum_{e \in S} c(e)$.

Moreover, we consider also the unit-cost version of our problem, we refer to it as the IMA problem: it is a particular case of CostIMA where each edge $e \in (V \times V) \setminus E$ has cost $c(e) = 1$ and where $c(S) = |S|$.

We give now some definitions useful to prove our results. We will use the definition of *live-edge graph* $X = (V, E_X)$ which is a directed graph where the set of nodes is equal to V and the set of edges is a subset of E. More specifically, the edge set E_X is given by a edge selection process in which each edge in E is either *live* or *blocked* according to its propagation probability. We can assume that, for each edge $e = (u, v)$ in the graph, a coin of bias p_e is flipped and the edges for which the coin indicated an activation are live, the remaining are blocked.

We denote by $\chi(S)$ the probability space in which each sample point specifies one possible set of outcomes for all the coin flips on the edges of $G(S)$, that is the set of all possible live-edge graphs of $G(S)$. For a node $a \in V$ and a live-edge graph X in $\chi(S)$, let $R(a, X)$ denote the set of all nodes that can be reached from a in graph X, that is for each node $v \in R(a, X)$, there exists a path from a to v consisting entirely of live edges with respect to the outcome of the coin flips that generates X.

It is easy to show that the information diffusion process is equivalent to a reachability problem in live-edge graphs: given any seed set A, the distribution of active node sets after the diffusion process ends is the same as the distribution of node sets reachable from A in live-edge graphs. Indeed, let

$$R(A, X) = \bigcup_{a \in A} R(a, X),$$

then $\sigma(A, S)$ can be computed as

$$\sigma(A, S) = \sum_{X \in \chi(S)} \mathbb{P}[X] \cdot |R(A, X)|.$$

Note that, the influence function $\sigma(A, S)$ cannot be evaluated exactly in polynomial time since it has been proven that it is generally $\#P$-complete for the Independent Cascade Model [5]. However, by simulating the diffusion process sufficiently many times and sampling the resulting active sets, it is possible to obtain arbitrarily good approximations to $\sigma(A, S)$. The next proposition bounds the number of times that the diffusion process must be simulated to obtain a good approximation of $\sigma(A, S)$.

Proposition 1 *([17]). If the diffusion process starting with A on graph $G(S)$ is simulated at least $\Omega\left(\frac{|V|^2}{\lambda^2} \ln \frac{1}{\delta}\right)$ times, then the average number of activated nodes over these simulations is a $(1+\lambda)$-approximation to $\sigma(A, S)$, with probability at least $1 - \delta$.*

Therefore, in the rest of the paper we can assume that we can compute $\sigma(A, S)$ within an arbitrary bound. This reflects to an additional factor $1 - \epsilon$, for any $\epsilon > 0$, to all algorithms presented in this paper.

Given a set of edges S, for each graph $X \in \chi(S)$ and subset of edges $T \subseteq S$, we denote by X^T the graph obtained by removing edges in T from X. To avoid cumbersome notation, when $T = \{e\}$ we denote $X^e = X^{\{e\}}$. Given two feasible solutions S_1 and S_2, such that $S_2 \subseteq S_1$, we denote with $\delta(S_1, S_2)$ the expected number of nodes affected by S_1 and not affected by S_2, formally:

$$\delta(S_1, S_2) = \sum_{X \in \chi(S_1)} \mathbb{P}[X] \cdot \left(|R(A, X)| - |R(A, X^T)| \right),$$

where $T = S_1 \setminus S_2$.

3. The IMA problem

In this section we present our results for the IMA problem, i.e. the unit-cost version of the CostIMA problem.

3.1. Hardness of approximation

We first show that the IMA problem does not admit a PTAS, unless $P = NP$. The result holds even when there is only one seed node, i.e. when $|A| = 1$. Moreover, since IMA is a special case of costIMA, any upper bound on the approximation of IMA also holds for CostIMA. In the next section, and in Section 4, we give an algorithm that almost matches the following upper bound on approximation.

Theorem 2. *It is NP-hard to approximate* IMA *within a factor greater than $1 - (2e)^{-1}$ for any A such that $|A| \geq 1$.*

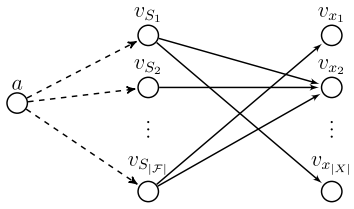

Fig. 1. Reduction used in Theorem 2. The dashed arcs denote those added in a solution.

Proof. The proof is based on a reduction from the *maximum set coverage problem* (MSC) which has been shown to be NP-hard to approximate within a factor greater than $1 - \frac{1}{e}$ [10]. In detail, in the MSC problem, we are given a finite set X, a finite family $\mathcal{F}$ of subsets of X, and an integer k', and we aim at finding a family $\mathcal{F}' \subseteq \mathcal{F}$ such that $|\mathcal{F}'| \leq k'$ and $s(\mathcal{F}') = |\cup_{S_i \in \mathcal{F}'} S_i|$ is maximum.

We follow the scheme of L-reductions [33, Chapter 16], since it has been shown that if there is an L-reduction with parameters a and b from maximization problem Π to maximization problem Π', and there is an α-approximation algorithm for Π', then there is an $(1 - ab(1-\alpha))$-approximation algorithm for Π [33, Chapter 16]. In our specific case, if there exists an α-approximation algorithm A_{IMA} for IMA and the following two conditions are satisfied for some values a and b:

$$OPT(I_{\text{IMA}}) \leq a\, OPT(I_{\text{MSC}}) \tag{1}$$

$$OPT(I_{\text{MSC}}) - s(S_{\text{MSC}}) \leq b\,(OPT(I_{\text{IMA}}) - \sigma(A, S_{\text{IMA}}))\,, \tag{2}$$

where OPT denotes the optimal value of an instance of an optimization problem, then there exists an approximation algorithm A_{MSC} for MSC with approximation an factor of $1 - ab(1-\alpha)$. Since it is NP-hard to approximate MSC within a factor greater than $1 - \frac{1}{e}$ [10], then the approximation factor of A_{MSC} must be smaller than $1 - \frac{1}{e}$, unless $P = NP$. This implies that $1 - ab(1-\alpha) < 1 - \frac{1}{e}$ that is, the approximation factor α of A_{IMA} must satisfy $\alpha < 1 - \frac{1}{abe}$, unless $P = NP$.

Therefore, in what follows we give a polynomial-time algorithm that transforms any instance $I_{\text{MSC}} = (X, \mathcal{F}, k')$ of MSC into an instance $I_{\text{IMA}} = (G, A, k)$ of IMA and a polynomial-time algorithm that transforms any solution S_{IMA} for I_{IMA} into a solution S_{MSC} for I_{MSC} such that Conditions (1) and (2) are satisfied.

Given $I_{\text{MSC}} = (X, \mathcal{F}, k')$, we define $I_{\text{IMA}} = (G, A, k)$ where $G = (V, E, p)$ is a directed graph containing a node v_{x_i} for each element $x_i \in X$, a node v_{S_j} for each set $S_j \in \mathcal{F}$, a seed node a, and a directed edge (v_{S_j}, v_{x_i}), whenever $x_i \in S_j$. We define $k = k'$ and $A = \{a\}$. We denote by $V_{\mathcal{F}}$ the set of nodes corresponding to sets in $\mathcal{F}$. The propagation probability p_e is equal to 1 if $e \in E \cup (A \times V_{\mathcal{F}})$ and 0 otherwise. See Fig. 1 for a visualization. Note that in I_{IMA}, the information diffusion is a deterministic process, as all probabilities are 0 or 1. Moreover, any solution S for I_{IMA} contains only edges (a, v_{S_j}), for some $S_j \in \mathcal{F}$, since any other edge would have probability 0 of being activated. Given a solution $S_{\text{IMA}} = \{(a, v_{S_j}) \mid S_j \in \mathcal{F}\}$ to I_{IMA}, we construct the solution $S_{\text{MSC}} = \{S_j \mid (a, v_{S_j}) \in S_{\text{IMA}}\}$ to I_{MSC}. W.l.o.g., we can assume that $|S_{\text{IMA}}| = k$ and since by construction $|S_{\text{MSC}}| = |S_{\text{IMA}}|$, then $|S_{\text{MSC}}| = k = k'$.

The nodes influenced by node a in S_{IMA} are all nodes v_{S_j} such that $(a, v_{S_j}) \in S_{\text{IMA}}$ and all the nodes v_{x_i} such that $x_i \in S_j$, for some S_j such that $(a, v_{S_j}) \in S_{\text{IMA}}$. The nodes of the former type are k, while the nodes of the latter type are all those nodes v_{x_i} such that $x_i \in \cup_{(a, v_{S_j}) \in S_{\text{IMA}}} S_j$. Therefore, $\sigma(A, S_{\text{IMA}}) = k + |\cup_{(a, v_{S_j}) \in S_{\text{IMA}}} S_j| = k + s(S_{\text{MSC}})$.

It follows that Conditions (1) and (2) are satisfied for $a = 2$, $b = 1$ since: $OPT(I_{\text{IMA}}) = OPT(I_{\text{MSC}}) + k \leq 2\, OPT(I_{\text{MSC}})$ and $OPT(I_{\text{MSC}}) - s(S_{\text{MSC}}) = (OPT(I_{\text{IMA}}) - k) - (\sigma(A, S_{\text{IMA}}) - k) = OPT(I_{\text{IMA}}) - \sigma(A, S_{\text{IMA}})$, where the first inequality is due to the fact that $OPT(I_{\text{MSC}}) \geq k$, as otherwise the greedy algorithm finds an optimal solution for MSC. The statement follows by plugging the values of a and b into $\alpha < 1 - \frac{1}{abe}$. This concludes the proof for $|A| = 1$. We can extend to cases in which $|A| > 1$ by adding nodes in A that can only form edges with propagation probability equal to 0. □

3.2. Greedy approximation algorithm

In this section, we propose an algorithm that guarantees a constant approximation ratio for the IMA problem. The algorithm exploits the results of Nemhauser et al. on the approximation of monotone submodular objective functions [27]. Let us consider the following optimization problem: given a finite set N, an integer k', and a real-valued function z defined on the set of subsets of N, find a set $S \subseteq N$ such that $|S| \leq k'$ and $z(S)$ is maximum. If z is *monotone and submodular*,[1] then the following greedy algorithm exhibits an approximation of $1 - \frac{1}{e}$ [27]: start with the empty set, and repeatedly add an element that gives the maximal marginal gain, that is if S is a partial solution, choose the element $j \in N \setminus S$ that maximizes $z(S \cup \{j\})$.

[1] For a ground set N, a function $z : 2^N \to \mathbb{R}$ is submodular if for any pair of sets $S \subseteq T \subseteq N$ and for any element $e \in N \setminus T$, $z(S \cup \{e\}) - z(S) \geq z(T \cup \{e\}) - z(T)$.

Algorithm 1: GREEDYIMA algorithm.

Input : A directed graph $G = (V, E)$; a set of vertices $A \subseteq V$; and an integer $k \in \mathbb{N}$
Output: Set of edges $S \subseteq (A \times V) \setminus E$ such that $|S| \leq k$

$S := \emptyset$;
for $i = 1, 2, \ldots, k$ **do**
  $\hat{e} = \arg\max\{\sigma(A, S \cup \{e\}) \mid e = (a, v) \in (A \times V) \setminus (E \cup S)\}$;
  $S := S \cup \{\hat{e}\}$;
return S;

Theorem 3 *([27]). For a non-negative, monotone submodular function z, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the value of z. Then S provides a* $\left(1 - \frac{1}{e}\right)$*-approximation.*

In this paper, we exploit such results by showing that $\sigma(A, \cdot)$ is monotone and submodular with respect to the possible set of edges incident to nodes in A. In fact, assuming that for a set $S \subseteq (A \times V) \setminus E$ we are able to compute $\sigma(A, S)$,[2] then Algorithm 1 provides a $\left(1 - \frac{1}{e}\right)$-approximation. Algorithm 1 iterates k times and, at each iteration, it adds to an initially empty solution S an edge $\hat{e} = (\hat{a}, \hat{v})$ s.t. $(\hat{a}, \hat{v}) \in (A \times V) \setminus E$ that, when added to S, gives the largest marginal increase in the value of $\sigma(A, S)$, that is $\sigma(A, S \cup \{\hat{e}\})$ is maximum among all the possible edges in $(A \times V) \setminus (E \cup S)$ to be added to S. The next theorem shows that $\sigma(A, \cdot)$ is monotone and submodular.

Theorem 4. *Given a graph $G = (V, E, p)$, $\sigma(A, S)$ is a monotonically increasing submodular function of sets $S \subseteq (A \times V) \setminus E$.*

In order to prove the theorem, we first show, in the next lemma, that function $R(A, \cdot)$ is submodular with respect to the insertion of edges outgoing nodes in A to live-edge graphs. Note that this is a deterministic property.

Lemma 5. *Given a set of nodes $A \subseteq V$, two live-edge graphs X and Y such that $E_X \subseteq E_Y$ and $E_Y \setminus E_X \subseteq A \times V$, and an edge $e \in (A \times V) \setminus E$, let X^+ and Y^+ denote the live-edge graphs obtained by adding e to X and Y, respectively, that is $X^+ = (V, E_X \cup \{e\})$ and $Y^+ = (V, E_Y \cup \{e\})$. Then, $|R(A, Y^+)| - |R(A, Y)| \leq |R(A, X^+)| - |R(A, X)|$.*

Proof. Let $r(X, e) = R(A, X^+) \setminus R(A, X)$, that is, $r(X, e)$ is the set of nodes that are reachable from A in X^+ by means of edge e and that are not reachable in X. Similarly, let $r(Y, e) = R(A, Y^+) \setminus R(A, Y)$. Since $E_X \subseteq E_Y$ and $E_Y \setminus E_X \subseteq A \times V$, then $R(A, X) \subseteq R(A, Y)$ and the set of nodes that are reachable from A *only* by means of edge e is smaller in Y^+ than in X^+. It follows that $|r(X, e)| \geq |r(Y, e)|$. Therefore, $|R(A, X^+)| - |R(A, X)| = |r(X, e)| \geq |r(Y, e)| = |R(A, Y^+)| - |R(A, Y)|$. □

We can now prove Theorem 4.

Proof of Theorem 4. To prove that σ is a monotonically increasing function, we show that for each $S \subseteq (A \times V) \setminus E$ and $e = (a, v) \in (A \times V) \setminus (E \cup S)$, $\sigma(A, S \cup e) - \sigma(A, S) \geq 0$, that is

$$\sum_{X \in \chi(S \cup \{e\})} \mathbb{P}[X] \cdot |R(A, X)| - \sum_{X \in \chi(S)} \mathbb{P}[X] \cdot |R(A, X)| \geq 0. \tag{3}$$

For each live-edge graph X in $\chi(S)$), there are two different corresponding live-edge graph X^+ and X^- in $\chi(S \cup \{e\})$, whose edge sets depend on the outcome of the coin flipped for e, that is $E_{X^+} = E_X \cup \{e\}$ and $E_{X^-} = E_X$, respectively. The probabilities for the live-edge graph X^+ and X^- to occur, are: $\mathbb{P}[X^+] = \mathbb{P}[X] \cdot p_e$ and $\mathbb{P}[X^-] = \mathbb{P}[X] \cdot (1 - p_e)$, while the set of reachable nodes are such that $R(A, X) \subseteq R(A, X^+)$ and $R(A, X) = R(A, X^-)$, because in X^+ there is one more edge e and $X^- = X$. Therefore, $|R(a, X^+)| \geq |R(a, X)|$ and we can prove Inequality (3) as follows:

$$\sum_{X \in \chi(S)} \left(\mathbb{P}[X^+] \cdot |R(A, X^+)| + \mathbb{P}[X^-] \cdot |R(A, X^-)|\right) - \sum_{X \in \chi(S)} \mathbb{P}[X] \cdot |R(A, X)| =$$

$$\sum_{X \in \chi(S)} \left(\mathbb{P}[X] \cdot p_e \cdot |R(A, X^+)| + \mathbb{P}[X] \cdot (1 - p_e) \cdot |R(A, X^-)| - \mathbb{P}[X] \cdot |R(A, X)|\right) \geq$$

$$\sum_{X \in \chi(S)} \left(\mathbb{P}[X] \cdot p_e \cdot |R(A, X)| + \mathbb{P}[X] \cdot (1 - p_e) \cdot |R(A, X)| - \mathbb{P}[X] \cdot |R(A, X)|\right) = 0.$$

This shows that σ is a monotonically increasing.

[2] We showed in Section 2 how to find an arbitrarily good approximation of $\sigma(A, S)$ in polynomial time.

To prove the submodularity, we show that for each pair of sets S, T such that $S \subseteq T \subset (A \times V) \setminus E$ and for each $e = (a, v) \in (A \times V) \setminus T$, the increment in expected number of influenced nodes that edge e causes in $S \cup \{e\}$ is larger than the increment it produces in $T \cup \{e\}$, that is $\sigma(A, S \cup \{e\}) - \sigma(A, S) \geq \sigma(A, T \cup \{e\}) - \sigma(A, T)$, or

$$\sum_{X \in \chi(S \cup \{e\})} \mathbb{P}[X] \cdot |R(A, X)| - \sum_{X \in \chi(S)} \mathbb{P}[X] \cdot |R(A, X)| \geq \tag{4}$$

$$\sum_{X \in \chi(T \cup \{e\})} \mathbb{P}[X] \cdot |R(A, X)| - \sum_{X \in \chi(T)} \mathbb{P}[X] \cdot |R(A, X)|. \tag{5}$$

For each live-edge graph X in $\chi(S)$ let us denote by $\chi(T, X)$ the set of live-edge graphs in $\chi(T)$ that have X as a subgraph and possibly contain other edges in $T \setminus S$. In other words, a live-edge graphs in $\chi(T, X)$ has been generated with the same outcomes as X on the coin flips in the edges of $E \cup S$ and it has other outcomes for edges in $T \setminus S$.

As in the proof for monotonicity, for each live-edge graph X in $\chi(T)$, let X^+ and X^- be the live-edge graphs in $\chi(T \cup \{e\})$ such that $E_{X^+} = E_X \cup \{e\}$ and $E_{X^-} = E_X$, respectively. Again, $\mathbb{P}[X^+] = \mathbb{P}[X] \cdot p_e$, $\mathbb{P}[X^-] = \mathbb{P}[X] \cdot (1 - p_e)$, $R(A, X) \subseteq R(A, X^+)$, and $R(A, X) = R(A, X^-)$.

Then, Formula (4) is equal to

$$\sum_{X \in \chi(S)} \left(\mathbb{P}[X^+] \cdot |R(A, X^+)| + \mathbb{P}[X^-] \cdot |R(A, X^-)| - \mathbb{P}[X] \cdot |R(A, X)|\right) =$$

$$\sum_{X \in \chi(S)} \left(\mathbb{P}[X] \cdot p_e \cdot |R(A, X^+)| + \mathbb{P}[X] \cdot (1 - p_e) \cdot |R(A, X^-)| - \mathbb{P}[X] \cdot |R(A, X)|\right) =$$

$$\sum_{X \in \chi(S)} \mathbb{P}[X] \cdot p_e \cdot (|R(A, X^+)| - |R(A, X)|).$$

For each $X \in \chi(T)$, there exists a unique pair $X' \in \chi(S)$, $Y \in \chi(T, X')$ such that $X = Y$. Similarly, for each $X \in \chi(T \cup \{e\})$, there exists a unique pair $X' \in \chi(S)$, $Y \in \chi(T, X')$ such that $\mathbb{P}[X] \cdot |R(A, X)| = \mathbb{P}[Y^+] \cdot |R(A, Y^+)| + \mathbb{P}[Y^-] \cdot |R(A, Y^-)|$. It follows that Formula (5) is equal to

$$\sum_{X \in \chi(S)} \sum_{Y \in \chi(T, X)} \left(\mathbb{P}[Y^+] \cdot |R(A, Y^+)| + \mathbb{P}[Y^-] \cdot |R(A, Y^-)| - \mathbb{P}[Y] \cdot |R(A, Y)|\right) =$$

$$\sum_{X \in \chi(S)} \sum_{Y \in \chi(T, X)} \left(\mathbb{P}[Y] \cdot p_e \cdot (|R(A, Y^+)| - |R(A, Y)|)\right).$$

By Lemma 5, $|R(A, Y^+)| - |R(A, Y)| \leq |R(A, X^+)| - |R(A, X)|$. Moreover, $\sum_{Y \in \chi(T, X)} \mathbb{P}[Y] = \mathbb{P}[X]$ and then

$$\sum_{X \in \chi(S)} \sum_{Y \in \chi(T, X)} \mathbb{P}[Y] \cdot p_e \cdot (|R(A, Y^+)| - |R(A, Y)|) \leq \sum_{X \in \chi(S))} \mathbb{P}[X] \cdot p_e \cdot (|R(A, X^+)| - |R(A, X)|),$$

which concludes the proof. □

Algorithm 2: GreedyIMA algorithm with approximate estimation of marginal increment.

Input : A directed graph $G = (V, E)$; a set of vertices $A \subseteq V$; and an integer $k \in \mathbb{N}$
Output: Set of edges $S \subseteq (A \times V) \setminus E$ such that $|S| \leq k$

```
S := ∅;
for i = 1, 2, ..., k do
    foreach e ∈ (A × V) \ (E ∪ S) do
        Use repeated sampling to estimate a (1 + λ)-approximation of σ(A, S ∪ {e}) with prob. 1 − δ;
        Let σ̃(A, S ∪ {e}) be the estimation;
    ê = arg max{σ̃(A, S ∪ {e}) | e = (a, v) ∈ (A × V) \ (E ∪ S)};
    S := S ∪ {ê};
return S;
```

Theorem 3 can be generalized to the case in which each step of the greedy algorithm, at each iteration, selects an element whose marginal increment is an α-approximation to the maximum one. In this case, it guarantees an approximation factor of $1 - \frac{1}{e^{\alpha}}$ [14, Chapter 3]. Moreover, thanks to Proposition 1, we can estimate the maximum marginal increment to within a factor $(1 + \lambda)$ to the maximal one in polynomial time. In this case, the greedy algorithm guarantees a $\left(1 - \frac{1}{e} - \epsilon\right)$-approximation, where ϵ depends on λ and goes to 0 as $\lambda \to 0$. By combining these results, we can formally define Algorithm 2 that differs from Algorithm 1 on how it computes $\sigma(A, S)$.

Theorem 6. *Algorithm 2 guarantees an approximation factor of* $\left(1-\frac{1}{e}-\epsilon\right)$ *for the* IMA *problem, where* ϵ *is any positive real number.*

We notice that the computational complexity of Algorithm 2 is $O(k\cdot n\cdot g(n,m+k))$, where $g(n,m+k)$ is the complexity of computing an approximation $\tilde{\sigma}(A,S)$ of $\sigma(A,S)$ in a graph with n nodes and $m+k$ edges. Specifically, it runs in k iterations, each of which requires estimating the expected spread of $O(n)$ node sets running a breadth first search on the live-edge graphs. Since $g(n,m+k)=O((n+m+k)\cdot R)$ where R is the number of simulations, then the complexity of the greedy algorithm is $O(k\cdot n\cdot(n+m+k)\cdot R)$.

4. Approximation algorithms for the CostIMA problem

In this section we introduce our approximation algorithm for the CostIMA problem. First we propose a greedy algorithm that achieves an approximation factor of $\frac{1}{2}\left(1-\frac{1}{e}\right)$ for the CostIMA problem, then we improve such approximation factor to $\left(1-\frac{1}{e}\right)$ by using an enumeration technique.

Algorithm 3: GreedyCostIMA algorithm.

Input : A directed graph $G=(V,E)$, a budget $k\geq 0$ a seed set A
Output: A set of edges $S\subseteq(A\times V)\setminus E$ such that $c(S)\leq k$

1 $S:=\emptyset$;
2 $T:=(A\times V)\setminus E$;
3 $e_M:=\arg\max_{e\in(A\times V)\setminus E}\{\sigma(A,\{e\})\}$;
4 **while** $T\neq\emptyset$ **do**
5 $\quad\hat{e}:=\arg\max_{e\in T}\left\{\frac{\delta(S\cup\{e\},S)}{c(e)}\right\}$;
6 $\quad$**if** $k-c(\hat{e})\geq 0$ **then**
7 $\quad\quad S:=S\cup\{\hat{e}\}$;
8 $\quad\quad k:=k-c(\hat{e})$;
9 $\quad T:=T\setminus\{\hat{e}\}$;
10 **return** $\arg\max\{\sigma(A,S),\sigma(A,\{e_M\})\}$;

Our first algorithm, whose pseudocode is reported in Algorithm 3, outputs a solution that maximizes the expected number of affected nodes between two possible solutions described in the following. The first solution is found at line 3 and is made of a single edge (a_M,v_M) for which $\sigma(A,\{(a_M,v_M)\})$ is maximized; the second solution is obtained by a greedy algorithm at lines 4–9.

In particular, the greedy phase, selects at each step an edge $\hat{e}$ to be added to the solution S obtained at the previous iteration, such that the ratio between $\delta(S\cup\{\hat{e}\},S)$, that is the expected number of nodes affected by $S\cup\{\hat{e}\}$ and not affected by S, and $c(\hat{e})$ is maximized (see line 5). Then, if cost $c(S\cup\{\hat{e}\})$ does not violate the budget, the edge $\hat{e}$ is added to S, otherwise the edge is discarded (lines 6–8).

Next, we analyze the performance guaranteed by Algorithm 3. We denote by S^* an optimal solution to the problem. Let l be the number of edges added to the solution by the greedy algorithm until an edge from S^* is considered (i.e. it maximizes the ratio at line 5) but not added to the solution because it violates the budget constraint. Let j_i, $i=1,2,\ldots,l$, be the indices of the first l iterations of the greedy algorithm in which an edge is added to the solution, $j_i<j_{i+1}$, and let S_i be the solution at the end of iteration j_i. Moreover, let j_{l+1} be the index of the first iteration in which an edge e in S^* is considered but not added to S because it violates the budget constraint and let $S_{l+1}=S_l\cup\{e\}$. For each $i=1,2,\ldots,l+1$, we denote by $\bar{c}_i$ the marginal cost of S_i as computed at line 5, $\bar{c}_i=c(\hat{e})$, where $\hat{e}$ is the edge selected at iteration j_i.

The next lemmas are the core of our analysis, note that the statements are similar to lemmas in [19].

Lemma 7. *For each* $i=1,2,\ldots,l+1$, $\sigma(A,S_i)-\sigma(A,S_{i-1})\geq\frac{\bar{c}_i}{k}(\sigma(A,S^*)-\sigma(A,S_{i-1}))$.

Proof. First, we define δ_i to be the expected number of nodes affected by solution S_i and not affected by solution S_{i-1}, $\delta_i=\delta(S_i,S_{i-1})$.

It is easy to see that the following inequality holds

$$\sigma(A,S^*)-\sigma(A,S_{i-1})\leq\sum_{e\in S^*\setminus S_{i-1}}\delta(S_{i-1}\cup\{e\},S_{i-1}),\tag{6}$$

i.e. the value $\sigma(A,S^*)-\sigma(A,S_{i-1})$ is at most the sum, for each edge in S^* and not in S_{i-1}, of the expected number of nodes affected by such edge and not affected by solution (A,S_{i-1}).

Since the greedy algorithm selects at each step the element that maximizes the ratio between δ_i and $\bar{c}_i$, for each $e\in S^*\setminus S_{i-1}$ the following holds,

$$\frac{\delta(S_{i-1}\cup\{e\},S_{i-1})}{c(e)}\leq\frac{\delta_i}{\bar{c}_i}.$$

Therefore,

$$\sum_{e\in S^*\setminus S_{i-1}} \delta(S_{i-1}\cup\{e\}, S_{i-1}) \le \sum_{e\in S^*\setminus S_{i-1}} \frac{\delta_i}{\bar{c}_i} c(e) = \frac{\delta_i}{\bar{c}_i}\left(\sum_{e\in S^*\setminus S_{i-1}} c(e)\right) \le k\frac{\delta_i}{\bar{c}_i}.$$

To conclude the proof, we need to show that $\delta_i = \sigma(A, S_i) - \sigma(A, S_{i-1})$. Indeed, if $S_i \setminus S_{i-1} = \{e\}$,

$$\begin{aligned}\delta_i &= \sum_{X\in\chi(S_i)} \mathbb{P}[X]\cdot\left(|R(A,X)| - |R(A,X^e)|\right)\\ &= \sigma(A,S_i) - \sum_{X\in\chi(S_{i-1})} \mathbb{P}[X]\,(p_e|R(A,X)| + (1-p_e)|R(A,X)|)\\ &= \sigma(A,S_i) - \sigma(A,S_{i-1}). \quad \square\end{aligned}$$

Armed with Lemma 7, we prove the next lemma by induction on iterations j_i.

Lemma 8. *For each $i = 1, 2, \ldots, l+1$,*

$$\sigma(A,S_i) \ge \left[1 - \prod_{\ell=1}^{i}\left(1 - \frac{\bar{c}_\ell}{k}\right)\right]\sigma(A,S^*).$$

Proof. For $i = 1$, by Lemma 7, $\sigma(A,S_1) \ge \frac{\bar{c}_1}{k}\sigma(A,S^*) = \left[1 - \left(1 - \frac{\bar{c}_1}{k}\right)\right]\sigma(A,S^*)$. Let us assume that the statement holds for iterations $j_1, j_2, \ldots, j_{i-1}$, then

$$\begin{aligned}\sigma(A,S_i) &= \sigma(A,S_{i-1}) + [\sigma(A,S_i) - \sigma(A,S_{i-1})]\\ &\ge \sigma(A,S_{i-1}) + \frac{\bar{c}_i}{k}\left[\sigma(A,S^*) - \sigma(A,S_{i-1})\right]\\ &= \sigma(A,S_{i-1})\left(1 - \frac{\bar{c}_i}{k}\right) + \frac{\bar{c}_i}{k}\sigma(A,S^*)\end{aligned}$$

where the inequalities follows from Lemma 7.

To conclude the proof we apply the inductive hypothesis:

$$\begin{aligned}\sigma(A,S_{i-1})\left(1 - \frac{\bar{c}_i}{k}\right) + \frac{\bar{c}_i}{k}\sigma(A,S^*) &\ge \left[1 - \prod_{\ell=1}^{i-1}\left(1 - \frac{\bar{c}_\ell}{k}\right)\right]\left(1 - \frac{\bar{c}_i}{k}\right)\sigma(A,S^*) + \frac{\bar{c}_i}{k}\sigma(A,S^*)\\ &= \left[1 - \prod_{\ell=1}^{i}\left(1 - \frac{\bar{c}_\ell}{k}\right)\right]\sigma(A,S^*). \quad \square\end{aligned}$$

Before proving the main theorem, we prove the following technical lemma.

Lemma 9. *For a sequence of numbers $b_1, b_2, \ldots, b_n$ such that $\sum_{\ell=1}^{n} b_\ell = B$, the fallowing holds: $\prod_{i=1}^{n}\left(1 - \frac{b_i}{B}\right) \le \left(1 - \frac{1}{n}\right)^n$.*

Proof. We show the equivalent statement that function $\prod_{i=1}^{n}\left(1 - \frac{b_i}{B}\right)$ is maximized when $b_i = \frac{B}{n}$, for each $i = 1, 2, \ldots, n$, that is we solve the following optimization problem:

$$\begin{aligned}&\text{maximize} \prod_{i=1}^{n}\left(1 - \frac{b_i}{B}\right)\\ &\text{subject to} \sum_{\ell=1}^{n} b_\ell - B = 0.\end{aligned}$$

We use the method of Lagrangian multiplier. To this aim we define the Lagrangian function

$$L(b_1, b_2, \ldots, b_n, \lambda) = \prod_{i=1}^{n}\left(1 - \frac{b_i}{B}\right) - \lambda\left(\sum_{\ell=1}^{n} b_\ell - B\right),$$

where λ is a Lagrangian multiplier.

We solve $\nabla L = 0$, that is we solve the following $n+1$ equations:

$$\begin{cases} \frac{\partial L}{\partial b_i} = 0 & \forall\, i = 1, 2, \ldots, n \\ \sum_{\ell=1}^{n} b_\ell - B = 0. \end{cases} \tag{7}$$

The first n equations are satisfied if and only if, for each $i = 1, 2, \ldots, n$,

$$-\frac{1}{B} \prod_{j=1, j\neq i}^{n} \left(1 - \frac{b_j}{B}\right) - \lambda = 0,$$

that is $b_i = b_j$, for each $i, j = 1, 2, \ldots, n$. Plugging this into the last equation of (7), we obtain $nb_i = B$, for each $i = 1, 2, \ldots, n$, which is what we wanted to show. □

Theorem 10. *Algorithm 3 achieves an approximation factor of* $\frac{1}{2}\left(1 - \frac{1}{e}\right)$ *for the* CostIMA *problem.*

Proof. We first observe two facts:

1. since (A, S_{l+1}) violates the budget, then $c(S_{l+1}) > k$,
2. since $\sum_{j=1}^{l+1} \bar{c}_j = c(S_{l+1})$, then by Lemma 9 the following holds,

$$\prod_{j=1}^{l+1} \left(1 - \frac{\bar{c}_j}{c(S_{l+1})}\right) \leq \left(1 - \frac{1}{l+1}\right)^{l+1}.$$

Therefore, by applying Lemma 8 for $i = l+1$, we obtain:

$$\begin{aligned} \sigma(A, S_{l+1}) &\geq \left[1 - \prod_{j=1}^{l+1} \left(1 - \frac{\bar{c}_j}{k}\right)\right] \sigma(A, S^*) \\ &\geq \left[1 - \prod_{j=1}^{l+1} \left(1 - \frac{\bar{c}_j}{c(S_{l+1})}\right)\right] \sigma(A, S^*) \\ &\geq \left[1 - \left(1 - \frac{1}{l+1}\right)^{l+1}\right] \sigma(A, S^*) \\ &\geq \left(1 - \frac{1}{e}\right) \sigma(A, S^*). \end{aligned}$$

It follows that:

$$\sigma(A, S_{l+1}) = \sigma(A, S_l) + \delta_{l+1} \geq \left(1 - \frac{1}{e}\right) \sigma(A, S^*). \tag{8}$$

Moreover, since $\delta_{l+1} \leq \sigma(A, \{e_M\})$ as computed in line 3 of Algorithm 3, we get:

$$\sigma(A, S_l) + \sigma(A, \{e_M\}) \geq \left(1 - \frac{1}{e}\right) \sigma(A, S^*).$$

Finally, note that $\max\{\sigma(A, S_l), \sigma(A, \{e_M\})\} \geq \frac{1}{2}\left(1 - \frac{1}{e}\right) \sigma(A, S^*)$. □

As for Algorithm 2, we use repeated sampling to estimate a $(1+\lambda)$-approximation of $\sigma(A, S \cup \{e\})$ with probability $1 - \delta$. Also in this case the complexity of the greedy algorithm is $O(k \cdot n \cdot (n+m+k) \cdot R)$, where R is the number of simulations and the approximation ratio is $\frac{1}{2}\left(1 - \frac{1}{e}\right) - \epsilon$, for any $\epsilon > 0$.

We now propose an algorithm which improves the performance guarantee of Algorithm 3. Let I a fixed integer, we consider all the solutions of cardinality I (i.e. $|S| = I$) which have cost at most k, $c(S) \leq k$, and we complete each solution by using the greedy algorithm.

Theorem 11. *For* $I \geq 3$ *Algorithm 4 achieves an approximation factor of* $\left(1 - \frac{1}{e}\right)$ *for the* CostIMA *problem.*

Proof. We assume that $|S^*| > I$ since otherwise Algorithm 4 finds an optimal solution.

We sort the edges in S^* in decreasing order according to their marginal increment in the objective function.

Algorithm 4: GREEDY IMA algorithm with enumeration technique.

Input : A directed graph $G = (V, E)$, integer $I \in \mathbb{N}$ and an integer $k \in \mathbb{N}$
Output: A set of edges $S \subseteq (A \times V) \setminus E$ such that $c(S) \leq k$

```
1 S_1 := arg max{σ(A, S) : |S| < I, c(S) ≤ k};
2 S_2 := ∅;
3 T := (A × V) \ E;
4 foreach S ⊆ T such that |S| = I, c(S) ≤ k do
5     T := T \ S;
6     Complete S by using lines 4–9 of Algorithm 3 with T as possible edge set;
7     if σ(A, S) > σ(A, S_2) then
8         S_2 := S;
9 return arg max{σ(A, S_1), σ(A, S_2)};
```

Let S_Z be the first I elements in this order. We now consider the iteration of Algorithm 4 in which element S_Z is considered. We define $S_{Z'}$ such that $S = S_Z \cup S_{Z'}$, where S is the solution obtained after applying the greedy algorithm. It follows that:

$$\sigma(A, S) = \sigma(A, S_Z) + \delta(S_Z \cup S_{Z'}, S_Z).$$

The completion of S_Z to S is an application of the greedy heuristic from Algorithm 3 therefore, we can use the result and notation from the previous theorems. Let l be the number of edges added to the solution during the completion of S_Z to S until an edge e in $S^* \setminus S_Z$ is considered but not added to the solution because it violates the budget constraint, let $S_{l+1} = S_l \cup \{e\}$, and let $\delta_{l+1} = \delta(S_{l+1}, S)$. Applying Inequality (8) on the increment given by Z' to the objective function, we get:

$$\delta(S_Z \cup S_{Z'}, S_Z) + \delta_{l+1} \geq \left(1 - \frac{1}{e}\right)\sigma(A, S^* \setminus S_Z) \tag{9}$$

we observe that, since we ordered the elements in S^*, $\delta_{l+1} \leq \frac{\sigma(A, S_Z)}{I}$.

Therefore, applying Inequality (9) and the previous observation:

$$\begin{aligned}
\sigma(A, S) &= \sigma(A, S_Z) + \delta(S_Z \cup S_{Z'}, S_Z) \\
&\geq \sigma(A, S_Z) + \left(1 - \frac{1}{e}\right)\sigma(A, S^* \setminus S_Z) - \delta_{l+1} \\
&\geq \sigma(A, S_Z) + \left(1 - \frac{1}{e}\right)\sigma(A, S^* \setminus S_Z) - \frac{\sigma(A, S_Z)}{I} \\
&\geq \left(1 - \frac{1}{I}\right)\sigma(A, S_Z) + \left(1 - \frac{1}{e}\right)\sigma(A, S^* \setminus S_Z)
\end{aligned}$$

But, $\sigma(A, S_Z) + \sigma(A, S^* \setminus S_Z) \geq \sigma(A, S^*)$, and we get:

$$\begin{aligned}
\sigma(A, S) &\geq \left(1 - \frac{1}{e}\right)\sigma(A, S^*) + \left(\frac{1}{e} - \frac{1}{I}\right)\sigma(A, S_Z) \\
&\geq \left(1 - \frac{1}{e}\right)\sigma(A, S^*), \quad \text{for } I \geq 3
\end{aligned}$$

proving the theorem. □

As in the previous cases, Theorem 11 can be generalized to the case in which each step of the greedy algorithm uses repeated sampling to estimate a $(1+\lambda)$-approximation of $\sigma(A, S)$ with probability $(1-\delta)$. In this case, the greedy algorithm guarantees a $\left(1 - \frac{1}{e} - \epsilon\right)$-approximation, where $\epsilon \to 0$ as $\lambda \to 0$. We note that the complexity of the enumeration algorithm, which pseudocode is reported in Algorithm 4, is $O(k \cdot n \cdot (n + m + k) \cdot R)$ as for Algorithm 3 since I is a fixed integer.

5. Conclusion and future research

In this paper, we have shown that IMA admits a constant factor approximation algorithm by proving that the expected number of activated nodes is monotonically increasing and submodular with respect to the possible set of edges incident to the seeds. We further provide an upper bound to the approximation factor that is slightly higher than that guaranteed by our algorithm. Moreover, we have shown how to approximate the more general COSTIMA problem: we first provide an algorithm which achieves a $\frac{1}{2}\left(1 - \frac{1}{e}\right)$ approximation factor and then, using the enumeration technique, we improve the performance guarantee within a factor of $1 - \frac{1}{e}$.

As future works, we plan to analyze a minimization version of the CostIMA problem where we allow the deletion of edges incident to seeds. Moreover, our intent is to study the same problem in a generalization of ICM, which is the Decreasing Cascade model. In this model the probability of a node u to influence v is non-increasing as a function of the set of nodes that have previously tried to influence v. Other research directions that deserve further investigation include the study of the CostIMA problem on different information diffusion models such as LTM or the Triggering Model [17]. We are also interested, as a future work, in studying a generalization of the problem of Kempe et al. in which we are allowed to spend part of the budget to select seeds, which are not given, and part of it to create new edges incident to such seeds. Finally, we plan to assess the performance of our greedy algorithm from the experimental point of view and to propose some heuristics with the aim of improving the efficiency of the algorithm.

References

[1] L.A. Adamic, E. Adar, Friends and Neighbors on the Web, Soc. Netw., vol. 25, 2001, pp. 211–230.
[2] L. Backstrom, J. Leskovec, Supervised random walks: predicting and recommending links in social networks, in: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM, ACM, 2011, pp. 635–644.
[3] B. Bollobás, C. Borgs, J. Chayes, O. Riordan, Directed scale-free graphs, in: Proceedings of the 14th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA, SIAM, 2003, pp. 132–139.
[4] V. Chaoji, S. Ranu, R. Rastogi, R. Bhatt, Recommendations to boost content spread in social networks, in: Proceedings of the 21st World Wide Web Conference 2012, WWW12, ACM, 2012, pp. 529–538.
[5] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD10, 2010.
[6] G. Cordasco, L. Gargano, M. Lafond, L. Narayanan, A.A. Rescigno, U. Vaccaro, K. Wu, Whom to befriend to influence people, CoRR, arXiv:1611.08687, 2016.
[7] P. Crescenzi, G. D'Angelo, L. Severini, Y. Velaj, Greedily improving our own closeness centrality in a network, ACM Trans. Knowl. Discov. Data 11 (1) (2016) 9:1–9:32.
[8] G. D'Angelo, L. Severini, Y. Velaj, Influence maximization in the independent cascade model, in: Proceedings of the 17th Italian Conference on Theoretical Computer Science, ICTCS2016, in: CEUR Workshop Proc., vol. 1720, 2016, pp. 269–274, CEUR-WS.org.
[9] G. D'Angelo, L. Severini, Y. Velaj, Selecting nodes and buying links to maximize the information diffusion in a network, in: 42nd Intl. Symp. on Mathematical Foundations of Computer Science, MFCS 2017, in: LIPIcs, vol. 83, 2017, pp. 75:1–75:14.
[10] U. Feige, A threshold of ln n for approximating set cover, J. ACM 45 (4) (1998).
[11] J. Goldenberg, B. Libai, E. Muller, Talk of the network: a complex systems look at the underlying process of word-of-mouth, Mark. Lett. 12 (3) (2001) 211–223.
[12] J. Goldenberg, B. Libai, E. Muller, Using complex systems analysis to advance marketing theory development: modeling heterogeneity effects on new product growth through stochastic cellular automata, Acad. Mark. Sci. Rev. 2001 (9) (2001) 1.
[13] M. Granovetter, Threshold models of collective behavior, Am. J. Sociol. 83 (6) (1978) 1420–1443.
[14] D.S. Hochbaum, Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishing Co., 1997.
[15] P. Jaccard, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, Bull. Soc. Vaud. Sci. Nat. 37 (1901) 547–579.
[16] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD03, ACM, 2003, pp. 137–146.
[17] D. Kempe, J.M. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, Theory Comput. 11 (2015) 105–147.
[18] E.B. Khalil, B. Dilkina, L. Song, Scalable diffusion-aware optimization of network topology, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD14, ACM, 2014, pp. 1226–1235.
[19] S. Khuller, A. Moss, J. Naor, The budgeted maximum coverage problem, Inform. Process. Lett. 70 (1) (1999) 39–45.
[20] M. Kimura, K. Saito, H. Motoda, Solving the contamination minimization problem on networks for the linear threshold model, in: Proc. of the 10th Pacific Rim International Conference on Artificial Intelligence, PRICA08, Springer Berlin Heidelberg, 2008, pp. 977–984.
[21] M. Kimura, K. Saito, H. Motoda, Blocking links to minimize contamination spread in a social network, ACM Trans. Knowl. Discov. Data 3 (2) (2009) 9:1–9:23.
[22] C.J. Kuhlman, G. Tuli, S. Swarup, M.V. Marathe, S. Ravi, Blocking simple and complex contagion by edge removal, in: IEEE International Conference on Data Mining, ICDM13, IEEE, 2013.
[23] D. Li, Z. Xu, S. Li, X. Sun, A. Gupta, K.P. Sycara, Link recommendation for promoting information diffusion in social networks, in: 22nd International World Wide Web Conference (WWW), Companion Volume, ACM, 2013, pp. 185–186.
[24] Z.L. Li, X. Fang, O.R.L. Sheng, A survey of link recommendation for social networks: methods, theoretical foundations, and future research directions, CoRR, arXiv:1511.01868, 2015.
[25] D. Liben-Nowell, J.M. Kleinberg, The link prediction problem for social networks, in: Proceedings of the 12th ACM International Conference on Information and Knowledge Management, CIKM, ACM, 2003, pp. 556–559.
[26] L. Lü, T. Zhou, Link prediction in complex networks: a survey, Phys. A 390 (6) (2011) 1150–1170.
[27] G. Nemhauser, L. Wolsey, M. Fisher, An analysis of approximations for maximizing submodular set functions–I, Math. Program. 14 (1) (1978) 265–294.
[28] M. Newman, Clustering and preferential attachment in growing networks, Phys. Rev. E 64 (2) (2001) 025102.
[29] M. Papagelis, Refining social graph connectivity via shortcut edge addition, ACM Trans. Knowl. Discov. Data 10 (2) (2015) 12.
[30] N. Parotsidis, E. Pitoura, P. Tsaparas, Centrality-aware link recommendations, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM, ACM, 2016, pp. 503–512.
[31] T.C. Schelling, Micromotives and Macrobehavior, Norton & Company, 2006.
[32] D. Sheldon, B.N. Dilkina, A.N. Elmachtoub, R. Finseth, A. Sabharwal, J. Conrad, C.P. Gomes, D.B. Shmoys, W. Allen, O. Amundsen, W. Vaughan, Maximizing the spread of cascades using network design, CoRR, arXiv:1203.3514, 2012.
[33] D. Williamson, D. Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 2011.
[34] X. Wu, D. Sheldon, S. Zilberstein, Efficient algorithms to optimize diffusion processes under the independent cascade model, in: NIPS Workshop on Networks in the Social and Information Sciences, Montreal, Quebec, Canada, 2015.
[35] Z. Yu, C. Wang, J. Bu, X. Wang, Y. Wu, C. Chen, Friend recommendation with content spread enhancement in social networks, Inform. Sci. 309 (2015) 102–118.
[36] Y. Zhang, A. Adiga, A. Vullikanti, B.A. Prakash, Controlling propagation at group scale on networks, in: IEEE International Conference on Data Mining, ICDM15, 2015, pp. 619–628.