IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 8, 2019, accepted May 5, 2019, date of publication May 15, 2019, date of current version June 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917123

Influence Maximization Based Global Structural Properties: A Multi-Armed Bandit Approach

MOHAMMED ALSHAHRANI[1,2], (Student Member, IEEE), ZHU FUXI[1,3], AHMED SAMEH[4], (Member, IEEE), SOUFIANA MEKOUAR[5], (Member, IEEE), AND SICHAO LIU[6]

[1]Computer School, Wuhan University, Wuhan 430072, China
[2]College of Computer Science and IT, Al Baha University, Al Baha 65527, Saudi Arabia
[3]Information Engineering College, Wuhan College, Wuhan 430050, China
[4]College of Computer, Prince Sultan University, Riyadh 12435, Saudi Arabia
[5]Faculty of Sciences, Mohammed V University, Rabat 8007, Morocco
[6]College of informatics, Huazhong Agricultural University, Wuhan 430070, China

Corresponding author: Zhu Fuxi (fxzhu@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61272277 and Grant 91746206.

ABSTRACT The influence maximization problem is defined by identifying the seed set that has the most influence on other users in the network, which when selected, the cascading process reaches a large number of users. We use a greedy algorithm and an epsilon-greedy algorithm from the MAB models in this work, unlike prior works that used the MAB models to quantify the unknown propagation probability in the diffusion models. In this paper, we did not also make any assumption regarding the diffusion models and tries to learn to identify the most influential users based on designed reward function ''hybrid edge strength-similarity'' using global centrality measures and by trying to find a tradeoff between exploitation and exploration strategies. The new proposed reward function initializes the MAB algorithms using global characteristics that quantify the strength of each arm (edge). The proposed reward will feed algorithms from MAB models uses hybridization of edge betweenness centrality and Jaccard similarity measures with some level of participation of each measure. Then, three algorithms are proposed for the extraction of relevant influencers, namely: SRI-CGSS FEXPL-GREEDY) algorithm which almost exploiting the best arm; the SRI_CGSS FEXPR-GREEDY which is almost exploring; and the SRI-MAB ϵ-GREEDY algorithm that alternate between exploring and exploiting the best arms. We conduct extensive experiments on a large-scale graph in terms of influence spread, efficiency performance in terms of running time and space complexity, and how the reward parameters impact cumulative regret.

INDEX TERMS Influence maximization, relevant influencers, multi-armed bandits, semi-uniform strategies, local metric, global measure.

I. INTRODUCTION

The social network entities and relationships that bound those entities via different kinds of links include personal and professional exchanges that are an integral process in the daily life of each individual across the globe. The social network is a set of connected individuals distributed across different places in the world that was designed to facilitate the exchange of information among these individuals for various purposes and goals. Therefore, the process by which an information is propagated and adopted by individuals has become a popular research topic in recent years and addresses the problem of influence maximization using various techniques and methods [1]–[4]. In ordinary daily life, each individual is supposed to be connected in online platforms to their closest friends, family members, colleagues, and even strangers. Each individual within the social network determines whether to adopt new information (product) depending on their social contacts and to what extent the individual is frequently in touch with their social neighborhood.

To design an efficient viral marketing campaign, it is fundamental for companies to identify ''relevant influencers'' or the most influential users to select as initial spreaders with the aim of influencing them to promote information (product) and influence coverage across the network. Accordingly, the problem of influence maximization can be defined

The associate editor coordinating the review of this manuscript and approving it for publication was Jun Hu.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

formally by identifying a set of interconnected individuals through different kinds of relationships represented by means of a graph and thus determining the relevant influencers. These influencers form the seed set that, when selected, the influence propagation (cascading process) that follows a given probabilistic diffusion model will yield a large-spread coverage.

Noticeable and extensive efforts have been made recently to improve the efficiency, scalability, and performance of the influence maximization problem. Domingos and Richardson [5], were the first to model the problem of influence maximization as an arbitrary Markov random field. Then, the influence maximization was formulated as an algorithmic problem that provided the first approximation guarantees. In addition, due to the good performance achieved by the greedy algorithm [6], various scalable approaches based on a greedy algorithm, including CELF [7] and CELF++ [8], have been tried to improve the running time while providing a good influence spread for the selection of relevant influencers. These mentioned approaches concentrated mainly on running useless and redundant Monte Carlo simulations for selecting key influencers. However, despite the success in identifying numerous individuals, they are still inefficient for performing effectively in large-scale graphs without considering the considerable amount of time needed to simulate the spreading process on large graphs.

In addition, there was a remarkable effort to deal with the time-consuming issue of approaches based on greedy algorithms. This motivated scientists to find approaches based on heuristics that rely on local structural properties including degree discount heuristics [9], RND d-hops, CPRND d-hops [10], DERND D-hops and UERND D-hops [11], while other approaches focused on the use of global structural properties such as PrKatz [12], PageRank [13] and others [14]–[16], to extract the most influential users or ''relevant influencers''. Unlike previously discussed approaches based on greedy algorithms and heuristics, which assume that the graph and influence probability are available as input, other approaches seem to take another direction different to what was discussed. These approaches learn influence probabilities that cause the cascade process and assume that these probabilities are unknown and that the individuals should learn probabilities through interaction with the environment, which is modeled through different multi-armed bandits' algorithms [17]–[20]. Therefore, unlike what was done previously, our contribution falls between the use of global and local structural heuristics and feeds the score values on multi-armed bandit algorithms. We are seeking, more precisely, in this work to answer some of the challenging issues encountered when dealing with maximizing the influence spread inside the online community. Some of the classic and most interesting challenges are which individuals should the marketer target to maximize the profit. In the context of online social communities, users are connected via links and the aim is to identify which relevant influencers to select to maximize the spread of influence, and thus, increase the revenue of the marketer. Thus, all individuals interact in an online environment in which each of them is the most powerful to convince a large number of users to adopt the promoted information (products). Therefore, how can the marketer efficiently select such relevant influencers? What is the strategy that should be followed to attain an increase in profit?

Thus, we assume that some global and local structural properties regarding the network should be known and exploited, whereas we have no knowledge of what will be the final output, in other words, the number of influenced individuals who adopted the information. Our supposition for initial selection is to characterize each edge (arm) by the strength in terms of being between many individuals for the flow of information which is quantified by a normalized edge betweenness centrality with the preselected percentage sample of the individuals in the network to reduce the computation time. In addition, we need to know to what extent two individuals are similar in adopting the promoted information, which is measured by the local edge similarity which is introduced in section 3.

Presumably, a proper solution to the approached problem requires two main steps. The first step characterizes the edge score value that determines the strength and usefulness of each link compared to other links. Then, the matrix value is fed into a variant of an epsilon-greedy multi-armed bandit algorithm to select the most performant edges that seem to spread the information effectively. Second, the relevant influencers that are supposed to increase the adoption of promoted behaviors that maximize a certain objective function are identified from the selected edges. We assume that in our multi-armed bandit framework each individual chooses arms (edges) that maximize a certain objective functions. Generally, the multi-armed bandit problem encompasses a necessary conflict set in all individual decisions. The selection between all available actions, ''which arm to select'', provides an immediate reward of the chosen arms. Many multi-armed bandit algorithms have been used in the context of influence maximization, which focuses mainly on learning the propagation probability in the diffusion models [19]–[21], while our approaches use epsilon-greedy algorithms and feed our algorithms with extra knowledge about the network structure without any feedback about the influence achieved. The aim of our proposed approaches is to extract relevant influencers by using full exploitation and full exploration with the selection and use the classic epsilon-greedy algorithms.

In the proposed approach we investigate the identification of key influencers while using some global and local structural properties over edges by using a variant of the well-known epsilon-greedy algorithm. The main contributions of the present work can be outlined as follows.

- Design a new function that characterizes the arm's ability to spread the promoted behavior and high ability to adopt the information by using a hybridization of edge betweenness and local edge similarity.
- Introduce online algorithms based on a variant of the epsilon-greedy algorithm from a multi-armed bandit

framework, which seeks to achieve a tradeoff between exploration and exploitation.
- Compare the proposed approaches against state-of-the-art influence maximization algorithms on large-scale graphs.

The rest of the paper is organized as follows. Section II presents previous work related to our proposed approach, followed by Section III & section IV, which introduce the system model and a measure to quantify the arm strength and how many arm members are similar to each other. Section V presents algorithms for the identification of relevant influencers and performance analysis of the proposed algorithms. Then, Section VI outlines extensive experiment under the IC and LT models to test the effectiveness and efficiency of the proposed algorithms. Section VII discusses our results with some analysis, and Section VIII concludes the paper with the main findings and possible future research directions.

II. RELATED WORK

This section presents the closest related work to the proposed approach, especially recent research on the influence maximization that makes the use of the MAB models and some approaches based on centrality measures.

Carpentier et al. [21] proposed a bandit strategy called BARE. They considered local influence, where a node can influence only its immediate neighbors. The proposed strategy did not consider any information regarding the graph structure and that the information is gained in a sequential manner. During the influence process, each selected node receives feedback for influencing the other nodes. They demonstrated that a regret guarantee scales with the detectable dimension, a problem dependent quantity that is often much smaller than the number of nodes. They considered two important cases, knowing the immediate neighbors by their identification and the second case of knowing only the number of immediate neighbors. However, local influence with one source with a cascading process was not presented.

Vaswani et al. [22] proposed a diffusion-independent learning algorithm DILinUCB, for semi-bandit influence maximization based on the maximum-reachability approximation. It is a pairwise-influence semi-bandit feedback model that demonstrated its effectiveness in terms of the regret bound against existing work. They proposed a parametrization that did not consider any knowledge about the environment for the underlying diffusion model. They demonstrated that their objective function is a good approximation of the original influence maximization problem and provides their corresponding monotone and submodular function.

Vaswani et al. [20] studied the problem of influence maximization when no cascade model is available with the aim of estimating the influence probabilities as each seed set is selected sequentially. They made use of a combinatorial multi-armed bandit approach and used several algorithms from the MAB framework to decrease the regret in terms of influence spread caused by the lack of knowledge about the exact spread ability of the nodes. Their approach was evaluated on the real-world dataset.

Wen et al. [23] proposed and analyzed a computationally efficient UCB-based algorithm IMLinUCB. They specifically addressed two challenging problems: the combinatorial action space in which a seed set increases exponentially with respect to the maximum number of the most influential users considered and the limited feedback restricted to the portion of the influence network. They dealt with these two issues under the IC model with knowledge of edge semi-bandit feedback that provides a linear generalization which is appropriate for large-scale world problems. Then, they proposed a maximum observed relevance based on the network topology and a non-decreasing activation probabilities function. Their regret bounds were polynomial in all quantities of interest and had near-optimal dependence on the number of interactions. They showed through empirical results that their algorithm has low regret in real-world online influence maximization problems.

Lei et al. [19] attempted to maximize the influence spread when only the social graph is known with a fixed budget. The influence probability is unknown, and there is only an online tradeoff between exploitation and exploration through an iterative process over a number of rounds. The problem deals with influence maximization with incomplete knowledge, called online influence maximization since the agent learns the influence probabilities when the cascade process occurs. The approach proceeds through multiple rounds where some seed sets are chosen based on existing influence information. The cascading process starts by these selected seed sets, and the user feedback updates their knowledge about the influence propagation. They designed an efficient incremental algorithm that decreases the overload of the users' feedback and provides a more effective solution for influencing the maximization problem under partial information.

Lagrée et al. [24] proposed a new formulation for the problem of influence maximization called online influence maximization with persistence "OIMP" that tries to target a subpopulation from the graph without making any prior assumption on the used diffusion model. Then, they proposed an estimator method that can be reached from a given influential node based on real Twitter data. Afterward, they suggested a new algorithm GT-UCB based on upper confidence bounds and demonstrated its effectiveness in terms of influence spread in the simulated and real dataset.

Du et al. [25] suggested learning an influence function from observation without any knowledge of the diffusion model. They assumed a weighted average influence function which should be a coverage functions. Thus, they introduced a novel parameterization of such functions using a convex combination of random basis functions. Their method needs a strong technical condition for an accurate approximation to the reachability distribution.

Kandhway et al. [26] modeled the influence maximization problem as a susceptible-infected epidemic process and formulated the problem of selecting the seed set to maximize the influence coverage for a period of time during a campaign on a social network, where users are grouped based on their centrality measure and each group of users is influenced by an optimal control function. The purpose was to boost the influence spread by maximizing a designed reward function which is a combination of the overall infected users at the predefined deadline and the cost of applying the advertising over a fixed budget. The linear reward function was formulated by an optimal control approach using Pontryagin's maximum principle and then solved using the forward-backward sweep method. The formulation of the reward function includes local and global centrality metrics to maximize the spread of promoted information through an optimal control framework. They found that a simple degree provides good results on some social network graphs and that the central node is targeted when the companies have insufficient resources (budget) and non-central nodes should be targeted when the companies have a large number of resources.

Riquelme et al. [27] presented an interesting survey by collecting centrality metrics used recently on Twitter. They were more interested in identifying the most critical nodes or so-called most influential nodes in the network. They mainly presented some centrality metrics including degree and closeness and showed how the influence could be processed based on those traditional metrics from social network analysis.

Estevez et al. [28] addressed the problem of the choice of seed sets with the overlapping neighborhood by the use of a set covering a greedy algorithm by using the degree centrality and discounting the shared neighborhood during the cascading process.

Cataldi et al. [29] proposed a method based on page rank to identify authority on Twitter graph data and analyzed its effectiveness to determine important and recent topics in real time. In the same direction, Bollen et al. [30] combined the ISI impact factor used to rank journals and weighted page rank to obtain a more effective metric. The weight relied on the number of the citation with the aim to rank journal status.

Lü et al. [31] proposed LeaderRank, a new variant of page rank, which strongly connects the network strongly due to its methodology of adding a ground node and link to other nodes in both directions. They showed its performance and robustness in opposition to manipulations.

Serin et al. [32] introduced a social network sensitivity method that permits locating the most important relationships between a user and all other users in the network. The proposed method used local and global centrality metrics including degree, betweenness, and closeness to set up an entropy measure centrality. Generally, the method assigned a value to each user based on their impact in terms of the quantity of change that occurred in the system entropy due to their elimination. Then, three metrics were normalized, and thus, their products were computed to achieve an aggregate sensitivity for each user within the studied social network.

Mochalova and Nanopoulos [33] started with the assumption that most central users can reach a large number of users with the studied network. Their contribution was primarily focused only on structural properties. Then, they included other information about seed size and members' attitude to identify the most suited seed set for each case. From this, they provided a comprehensive overview of centrality used to quantify most important users within the network and performed extensive experiments on real-world social network data. Their results provided insight on how the studied centrality metrics can affect and impact the attitude of users on whether to adopt the promoted ideas, messages, and products. However, the authors did not compare their approaches with similar work that treated influence maximization with global and local centrality metrics and other well-known algorithms in the literature such as TIM, BCT, CELF++.

Bakshy et al.. [34] investigated the reposting of Twitter URLs to measure the influence of Twitter reposting behavior of users. They reported that Twitter URLs were reposted in a way that the original user was not included. Then, they designed a model for the total diffusion tree considered of a given event with the aim of predicting user influence within the Twitter data.

Weng et al. [35] introduced a new algorithm TwitterRank, which is an extended version of PageRank, a measure that quantifies the topic-sensitive influence of users on Twitter. By suggesting this method, other factors were considered such as similarity and homophily. Their study showed that the presence of "reciprocity" can be explained by the phenomenon of homophily. Their experiments demonstrated that their algorithm performs better than other measures. Their approach was based on latent Dirichlet allocation, which makes their formulation questionable since known data are sparse and the produced Twitter topics relies on terms instead of concepts.

Namtirtha et al. [36] proposed a new algorithm for weighted kshell decomposition that combined degree and kshell with the aim of measuring the influenceability of users within the network. The proposed algorithm demonstrated its performance through experiments on real network data and used the SIR model as a diffusion epidemic model in which a user becomes infected if it is adjacent to a user holding the information through a certain probability of infection. They found that the introduced combination surpassed some traditional centrality metrics including kshell decomposition and degree. They argued that the proposed algorithm is only dependent on the number of network edges, and thus, is cost-effective on large-scale graphs; however, their algorithm was tested on only ten thousand nodes, and the proposed method also had a higher correlation with the SIR model than some existing indexing methods. Additionally, the extraction of such key users should be tested on other well-known diffusion models such as the IC and LT models, which will be interesting to present here to show more of the effectiveness of such indexing metrics.

Al-garadi et al. [37] introduced a new method weighted k-core that improves the results of k-core and strengthens its weakness in terms that k-core considers links equally when computing the influence ability of users and that interaction between users can efficiently quantify how likely the users are to be willing to be influenced by each other. For this purpose, tracking the dynamics of interaction in real-world social networks can witness its performance in determining most important users when compared with other centrality based metrics such as degree, page rank and original k-core.

Ding et al. [38] presented the SpreadRand method that relies on the PageRank metric. This method builds a network that consists of user-retweets, where the edges have a weight assigned as a unique retweet. The duration time of the retweet is considered important in the design of the proposed method. The computation of the edge weights is obtained by the ratio of the number of retweets to tweets. Thus, they argued that the faster that others retweet, the more the spread coverage will be effective.

Unlike existing works on this subject that uses either MAB models to infer the unknown propagation probabilities or other related works that used some heuristics and algorithms from global and local perspective properties. Our approaches try to propose some efficient algorithms and that we try to learn which set of nodes as potential influencers by using some global and local properties without knowing the influence that will produce as final output.

III. PROBLEM FORMULATION

The problem of influence maximization (IM) is the process by which a promoted behavior is adopted among a large number of users. The purpose of IM is to find the most influential users or the "relevant influencers", known as the seed set, so that when selected the influence will propagate as much as possible in the network. This problem can be modeled as a multi-armed bandit problem where users interact in the network (environment), and after each choice, users receive a reward as feedback. This problem can be formalized in our setting by a tuple $IM_{G_p}^{MB} = (G, S, DM)$, where $G = (V, E)$ is a graph in which a set of users (individuals) $n = |V|$ interact with each other in the network through a set of relationships $m = |E|$. We consider that each link is quantified by a certain strength extracted from the global network structure such as edge betweenness. Additionally, the users that are members of such links are measured by a similarity metric to test whether each pair of users is willing to share the same ideas, opinions, information, and orientation. Let S refer to the set of most influential users (i.e., seed set), which is the main purpose of this work. Thus, to maximize the profit, a number of users must have adopted the promoted information. In addition, let the diffusion model DM measure the influence spread of such seed set S on graph G.

Basically, the IM can be defined in a multi-armed bandit paradigm as a set of users $|V|$ interacting with adjacent users linked via different type of relationship $|E|$ inside an environment G. The process begins by selecting $K < n$ seed sets, which are chosen based on either a basic centrality measure or efficient heuristics that initiate the diffusion process according to some diffusion models. In the MAB framework, the marketer chooses a user as the seed set to initiate the cascade process depending on its budget K according to certain structural criteria, and then a reward is received as feedback of the selected node. The reward received in our setting is a combination of local and global structural properties. Then, each activated node will attempt to activate inactivated nodes and the process proceeds according to the used diffusion model. The main purpose of IM is to find the most accurate and influential seed set S that maximizes the number of users touched by the promoted behavior $\sigma_{opt}(G, S, DM)$. where

$$\sigma = \sigma_{\text{opt}}\,(\text{G, S,DM})'\,\text{S} = \{\text{s}_\text{i}, \text{i} \in |\text{S}|\} = [\text{s}_1\text{s}_2, \ldots, \text{s}_{|\text{S}|} \quad (1)$$

where s_i is an element of seed set S.

$$\begin{aligned} S &= argmax\; U^{MAB}\,(G_s, L_{str}) \\ &= \arg\max_{S_i} \sigma(S) \end{aligned} \quad (2)$$

U^{MAB}U is a reward function that relies on global

G_s and local structural properties

L_{str} and a node is selected if it satisfies some constraint. We propose three algorithms based on MAB that are explained in detail in section V.

σ_{opt} is an unknown function, monotone and submodular.

IV. SYSTEM MODEL

In our system model, we consider a social network that consists of a set of users (i.e., nodes) and a set of relationships (i.e., edges) that links those users through distinct types of interactions and activities. In this system, users concerned with the exchange share as much information as possible and make the information largely available to a large number of users.

Each user in the network is conscious of its neighborhood, so he knows only what his immediate neighbors are sharing. Thus, each user knows and gathers certain information from his neighbors, and thus, this user can use collected information to his benefit and can decide intelligently within the network. The main purpose of the proposed algorithms is to maximize the spread of influence by selecting a set of influential users while keeping an acceptable running time as presented in the problem formulation section. However, the identification of such influential users is still an open research topic since its first formulation by Richardson and Dominguos [5] and Kemp et al. [6]. In this model, we assume that each pair of users are linked through edges that are quantified with the strength and how the members of each edge are similar to each other. Therefore, the information collected by users should help to find the most pertinent nodes that have the greatest influence on other users in terms of adopting the promoted behavior. Generally, the main issue is how can the marketer efficiently determine which users to select? How should each seed user be selected, knowing that the marketer does not know exactly the reward that he will receive while providing free samples to each seed set?

Therefore, we assume that the marketer has to decide which users to select from the network as a seed set, with the goal of maximizing the influence spread. We assume that each edge in the network can be evaluated by its strength through the edge betweenness centrality metric and that the pair's member of each edge is tested in terms of how they are similar. We assume that if two users are similar, they are more likely to share the same interests and goals. Thus, when identifying the most influential users, we assume that the marketer has some knowledge about local and global structural properties of the graph illustrated in our model as the pair's edge similarity and edge betweenness centrality. However, the marketer does not know exactly how many users will become potential customers.

Figure 1, shows an example of a considered scenario with a network in which the users interact with each other. In the figure, the marketer is responsible for choosing the seed set based on some structural properties and the feedback received as a reward when choosing a certain node as a seed set. We assume that all users in the environment communicate with each other and can know the degree of its neighbors.

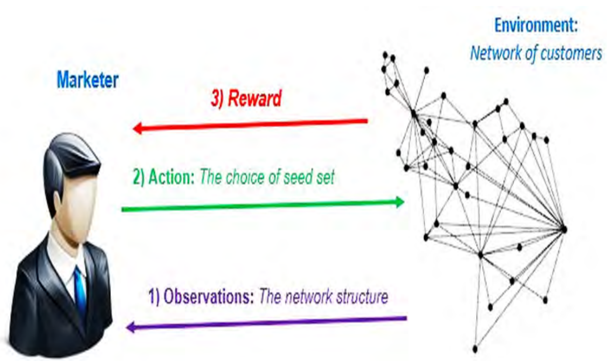

FIGURE 1. General design of seed set selection.

In this paper, we assume that the user can take action in discrete steps $t = 1, \ldots, N$. At every step, the user within the network observes the environment and collects information regarding edge strength and similarity that is presented in the next section. All this information is conserved in the agent database with the aim of helping them to select the seed set.

Therefore, the marketer has to decide at each time step whether to pick a user within the network as a new member in the seed set, keeping in mind that he has only a limited budget. Therefore, the choice of this influential seed set should be selected carefully. In this work, we assume that the marketer has some global knowledge about the graph that shapes the network relationships and that, based on this knowledge, he makes a decision and receive a reward that demonstrates how fortunate the choice is in terms of the structural network properties, but he does not know the influence that output the chosen seed set. Thus, this situation can be modeled perfectly via a multi-armed bandit framework [18], which is a class of machine learning problem that deals with the exploitation versus exploration dilemma. The multi-armed bandit problem can be described as a complex version of A/B testing [39] that involves the choice between a variety of actions, with unknown rewards. The main purpose here is to efficiently determine the best and optimal actions through a series of trials (rounds) that provide the highest reward.

A. MULTI-ARMED BANDIT FRAMEWORK

In this subsection, we recall briefly the multi-armed bandit framework (MAB), which is used in this paper. We start by presenting the basic concept and the state-of-the-art algorithms used throughout this work.

The multi-armed bandit (MAB) is considered a classic kind of reinforcement learning algorithm. It is a clever and complex version of A/B testing. In a multi-armed bandit formalism, each agent has the chance to choose among k arms (actions), and according to the agent, the choice receives a reward. The agent selects a single action over many rounds with the aim of maximizing the reward function, which is assumed in this paper as a hybrid strength-similarity based on a graph topology.

Two principal parameters are investigated in this work, and are the basic components of any algorithm from the MAB framework.

1. The identification of the best arm, which is the main purpose of this work in which the agent possesses a predefined budget and should decide, after a fixed number of rounds, to select the optimal arms that provide the highest reward.
2. The cumulative reward maximization is equivalent to cumulative regret minimization. Accordingly, at each step, the agent decides which arms to select and receives an immediate reward that is initialized with some global structural properties. The agent gathers the cumulative reward with the aim to maximize the sum of the obtained reward, and thus, minimize the cumulative regret.

The main goal of this setting is to reach an optimal tradeoff between the exploration vs exploitation (Er-vs-Et).

- Exploitation: this strategy focuses on the choice of the best arms that provide the highest rewards over several rounds.
- Exploration: this strategy focuses on the choice of random arms rather than selecting the best one that provides the best reward over several rounds, which may reduce the chances to choose other best arms.

Pure greedy: This approach relies on selecting each arm and greedily choosing the arm with the highest reward. This strategy corresponds to full exploitation, which is more likely to miss the selection of the best arm.

Epsilon-Greedy Algorithm: This strategy chooses the best arms mostly and occasionally selects arms at random. This strategy provides a good reward outcome if we balance between the exploration-exploitation strategies.

After a brief introduction of the multi-armed bandit context, we recall some global network properties including edge betweenness centrality and eccentricity from the global centrality metric and the edge similarity based on the local neighborhood and the degree centrality from the local network structure.

Starting from this point, we believe that the edge betweenness centrality constitutes an effective metric to quantify edge strength. The edge betweenness has been used in a variety of research works and has demonstrated its effectiveness in previous work in "identification of communities".

B. THE PROPOSED HYBRID EDGE STRENGTH-SIMILARITY

1) EDGE STRENGTH

The strength of the edge is a critical ingredient for analyzing and characterizing the relationship inside the network and the process that quantifies at which level users are interacting with each other. Therefore, it is of pivotal importance to quantify the tie strength globally inside the network to identify which tie is willing to be a bridge that connects many users and serves by excellence as a communication channel between a large number of users. So, what is the network centrality measure from network science that can effectively measure the tie importance globally inside the network?

The edge betweenness centrality represents an excellent measure that characterizes edge strength within the network globally. E. J. Newman [40] extended Linton C. Freeman's betweenness centrality [41]–[42] from network nodes to network edges, which attempt to identify the edges that fall within other pairs of nodes in the network by introducing the edge betweenness centrality (EBC) as the number of shortest paths between pairs of nodes that go through the edge. This centrality metric demonstrates that edges can occur on numerous shortest paths between other nodes (users) having higher EBC than those that fall within fewer numbers of shortest paths. Therefore, the edges with the highest EBC are more likely to control and manage the total information flow within the network.

We define the normalized edge betweenness centrality $EBC_{ap}(e)$ based on the EBC (e), introduced by [40], [43], as follows:

$$\mathrm{EBC_{ap}(e)} = \frac{1}{|E|} \sum\nolimits_{u \in V} \sum\nolimits_{v \in V} \frac{\sigma_{u,v}(e, sh = \frac{|V|*10}{100})}{\sigma_{u,v}(sh = \frac{|V|*10}{100})} \quad (3)$$

where u and v are two users within network G. Let $\sigma_{u,v}(e, sh = \frac{|V|*10}{100})$ denote the number of all shortest paths between two users u and v that run through 10% of the paths in the graph that crosses edge e. In addition, let $\sigma_{u,v}(sh = \frac{|V|*10}{100})$ denote the number of all shortest paths between two users' u and v that runs through 10% of the paths in the graph.

2) LOCAL EDGE SIMILARITY

The local edge similarity measure is based on a number of common local neighborhood for each pair of users over the degree centrality of those users. So, formally, the local edge similarity can be written as follows:

$$LES\,(u, v) = \frac{neig_{com}(u, v)}{\deg(u) + \deg(v)} \quad (4)$$

where $neig_{com}(u, v)$ represents the number of common neighbors of users u and v. In addition, $deg(u)$ depicts the number of neighbors of user u.

3) HYBRID EDGE STRENGTH-SIMILARITY METRIC

We introduce our proposed metric that relies on a combined global topology-based edge strength and local topology common neighborhood edge similarity measure. The introduced metric is supposed to determine edges that have the highest score value in terms of $EBC_{ap}(e)$ and $LES(u, v)$.

The use of a combination of those two metrics efficiently localizes any edge within the network, and each edge node should be close to most other nodes within the network and nodes of each edge should have high score similarity. We account for both the strength of the edge and the similarity of the node edge that is discounted with two discount parameters α and β, our proposed metric "hybrid edge strength-similarity" for edge $e = (u, v)$ can be written as follows:

$$ESS_{hyb(u,v)} = \alpha * EBC_{ap}\,(e) + \beta * LES\,(u, v) \quad (5)$$

As discussed earlier, we assume that some structural properties are known and that the arms are selected first according to a reward function that combines the individual arms' similarity and how likely it is that the arms are positioned between other arms in the networks. This reward function plays the role of the initialization for the MAB algorithm and assures the choice of arms with the best score value. This score value ranges from 0 to 1 and depends on another two discount parameters namely, α and βthat shows the contribution of each structural property in the reward function. Basically, those parameters are introduced to see their impact on the reward function and how it will improve the choice of seed set S and increase the objective function σ on a graph according to the diffusion models.

4) ECCENTRICITY OF VERTEX

The eccentricity of vertex u in a connected graph G can be defined as the maximum shortest paths from the vertex u to all other vertices in the network. The eccentricity quantifies to what extent the vertex is central in the network. Formally, the eccentricity of vertex $u$denoted $ecc(u)$ can be written as follows:

$$ecc\,(u) = \max_{\forall v \in G} d\,(u, v) \quad (6)$$

V. IDENTIFICATION OF RELEVANT INFLUENCER ALGORITHMS BASED ON MULTI-ARMED BANDITS SEMI-UNIFORM STRATEGIES

In this section, we present three algorithms for the identification of relevant influencers, the following diagram (figure 2) shows the proposed algorithms and the main steps performed required.

So, firstly we propose a new algorithm to identify relevant influencer-based global structural properties and full exploitation with negligible exploration SRI_CGSS fexpl-greedy ($\epsilon = \mathbf{1}$), secondly, a full exploration with negligible

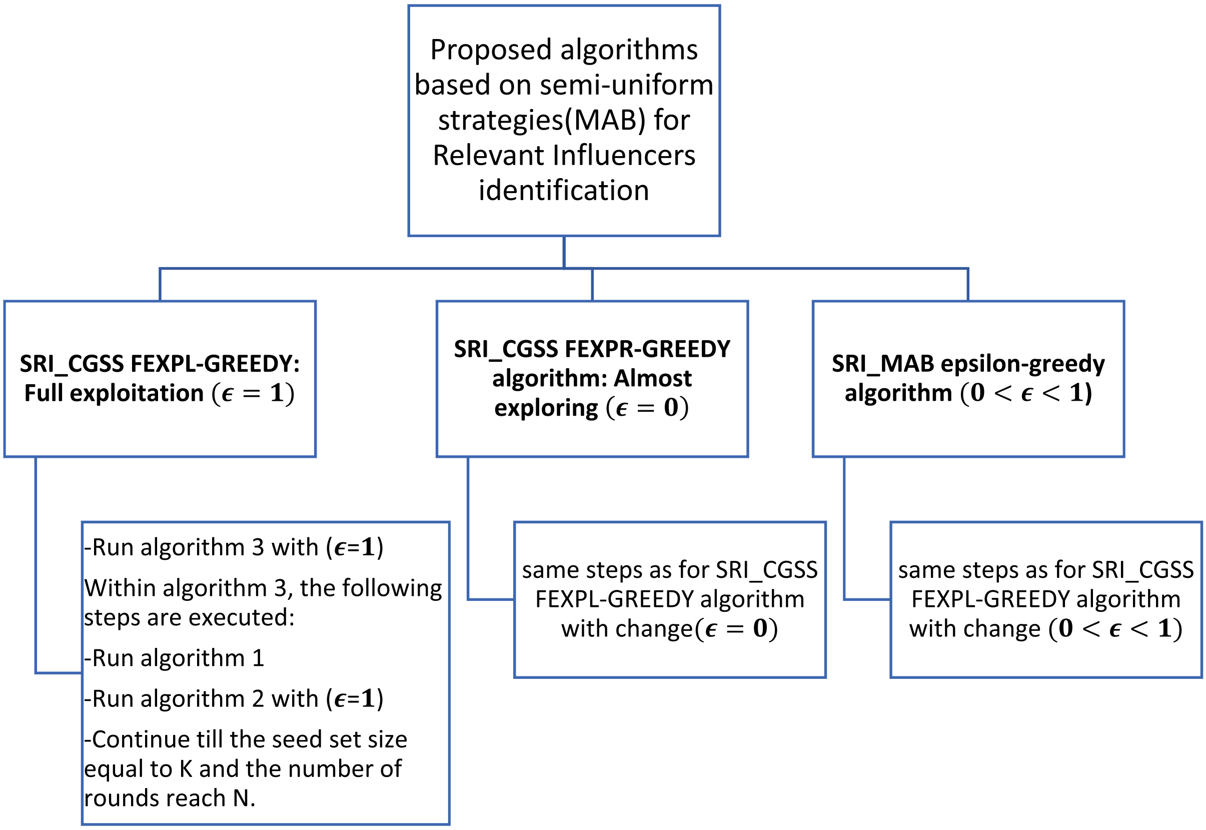

FIGURE 2. **Flowchart of the proposed algorithm for identification of relevant influencers.**

exploitation SRI_CGSS fexpr-greedy ($\epsilon = \mathbf{0}$) and finally an SRI_CGSS epsilon-greedy algorithm ($\mathbf{0} < \epsilon < \mathbf{1}$) that combines the use of exploitation and exploration to provide better results.

We have $|E|$ arms that have been tried at least once and obtained an initial initialization of the reward function. The agent has to select the arms at each step t that varies from 1 *to* N and receive a reward. Each individual attempts to maximize its reward and therefore minimize the regret.

The performance of MAB can generally be indicated by a regret that is defined as the cumulative reward acquired by choosing the best arm with the highest reward rew^t_{e*} minus the cumulative reward obtained with the chosen arms rew^t_e. The regret for the SRI_CGSS algorithms denoted $R_{\mathrm{SRI\_CGSS}}$ can be written as follows:

$$R_{SRI\_CGSS} = \max_{s \in E} rew^t_{e*} - rew^t_e;\ e^* = arg \max_{s \in E} rew\,(e) \tag{7}$$

And

$$\begin{aligned} rew_e &\equiv rew(e) \\ R_{\mathrm{SRI\_CGSS}}\,(N) &= E\left[r^*\,(N)\right] - E\,[r\,(N)] \end{aligned} \tag{8}$$

where $rew\,(e)$ represents cumulative reward to choose arm e and e^* is the optimal selected arm, and $u, v \in V \times V$ $E\left[r^*\,(N)\right]$ is the expected regret when selecting the optimal arm at round N. rnd refers to selection of random value among a set of values, and rw_{rnd} is the selected reward value randomly.

$$\begin{cases} rnd\,(ESS\_hyb\,(u, v)) \\ rw_{rnd} \sim ESS\_hyb\,(u, v) \\ ESS\_hyb \in [0, 1] \end{cases} \tag{9}$$

$P\,(rw_{rnd})$ is the probability of occurrence of a certain random reward value rw_{rnd} and n here presents the number of all possible values that may be selected.

$$P\,(rw_{rnd}) = \left\{P\left(rw^n_{rnd}\right), \forall n \in |rw_{rnd}|\right\} \tag{10}$$

The edges and corresponding nodes are extracted according to ESS_hyb and $P\,(rw_{rnd})$ values.

$$\begin{cases} u, v = \arg\max_{s \in E} ESS\_hyb(e) \\ u_1, v_1 = \arg\max_{s \in E} P(rw_{rnd}) \end{cases} \tag{11}$$

In other terms, the reward of selecting an edge denoted $\theta(t)$, and therefore, a node, can be expressed as follows:

$$\begin{aligned} \theta(t) = (1-\epsilon)\left[\sum\nolimits_{t=1}^{N} \frac{t}{n} \psi_{EES}\,(t) + \sum\nolimits_{t=1}^{N} \frac{t}{t_1} \varphi^{ecc}_{vtx^s}\,(t)\right] \\ +\epsilon\Big[\sum\nolimits_{t=1}^{N} \frac{t}{n+t_1} P(rw_{rnd})(1+\phi^{deg}_{vtx^s}\,(t))\Big] \end{aligned} \tag{12}$$

Let $\psi_{EES} = \max\left(\{ESS\_hyb\,(u, v)\,, u, v \in V \times V\}\right)$ represents the maximal value of the hybrid edge strength-similarity metric.

Similarly, the reward score of selecting a vertex vtx^s via eccentricity denoted $\varphi^{ecc}_{vtx^s}$ and the reward value of selecting a vertex vtx^s via degree measure denoted $\phi^{deg}_{vtx^s}$ can be defined as follows:

$$\varphi^{ecc}_{vtx^s} = \begin{cases} ecc\,(u)\,, & if\ ecc(u) \geq ecc(v) \\ ecc\,(v)\,, & Otherwise, \end{cases} \quad u, v \in \arg\max_{s \in E} ESS\_hyb(e) \tag{13}$$

$$\phi^{deg}_{vtx^s} = \begin{cases} deg\,(u_1)\,, & if\ \deg(u_1) \geq deg(v_1) \\ deg\,(v_1)\,, & Otherwise \end{cases} \quad u_1, v_1 \in \arg\max_{s \in E} P(rw_{rnd}) \tag{14}$$

Algorithm 1, "*ESBJ_Reward*", computes the hybrid edge similarity strength that quantifies each edge according to its position within the network and how the members of the edge are similar to each other. The algorithm takes a graph G, a number of nodes that we use from the entire population of the network, to compute the ECB_{ap}, and two parameters α, β that control the contribution of the edge strength and edge similarity.

Algorithm 1 Edge Strength **ESBJ_Reward**

Input: $GraphG = (VE)$, $nb\_node\_covered\ nb\_c, \alpha, \beta$
Output: *GS_Edge_Reward*
1. *EBC_ap is computed from eq* 3
2. *LES is computed from eq* 4
3. *GS_Edge_Reward* ← 0
4. *Selected_edge* ← ∅
5. ***For*** *e in* $|E|$
6. ***For*** *i in* $|V|$
7. ***For*** *j in* $|V|$
8. ***If*** $i! = j$
9. $GS\_Edge\_Reward(e) = \alpha * EBC_{ap}(enb_c) + \beta * LES(ij)$
10. $Selected\_edge \leftarrow Selected_edge \cup \{(i, j)\}$
11. ***End For***
12. ***End For***
13. ***End For***
14. ***Return*** *GS_Edge_Reward*, *Selected_edge*

Algorithm 2 "MAB_Edge Selection" is designed to select the edges according to multi-armed bandit algorithms. This work makes use of the epsilon-greedy algorithm and its variant. Therefore, considering $0 <\in< 1$, the main purpose of the algorithm is to select edges and their corresponding reward that is received depending on the structural properties. The algorithm starts by initializing the reward and candidate edges to be selected to an empty set (line 1). Then, the reward function and candidate edge are computed and selected according to the *GS_Edge_Reward* value and the value of $(1- \in)$. The algorithm alternate between the use of user eccentricity and the number of user neighbors depends on the value of hybrid edge strength-similarity and $1- \in$ value.

Algorithm 2 MAB_ Edge Selection

Input: $GraphG = (V, E)$, ϵ*Selected_edge*, *GS_Edge_Reward*
Output: *select_candidaterew_candidate*
1. ***Initialization:*** *rew_candidate*←0, *select_candidate*←*[], select_op_vert*← []
2. ***If*** *(max(GS_Edge_Reward)>(1-*ϵ*)*
3. *indx=argmax(GS_Edge_Reward)*
4. *candidate_vert1, candidate_vert2*← *Selected_edge[indx]*
5. *select_candidate*←*select_candidate*∪ *{candidate_vert1, candidate_vert2 }*
6. ***If*** *(is.connected(G))*
7. *Selected_op_vert1*←*EC(G, candidate_vert1)*
8. *Selected_op_vert2*←*EC(G, candidate_vert2)*
9. *select_(op_vert)*←*select_(op_vert)*∪ *{Selected_op_vert1, Selected_op_vert2}*
10. ***END IF***
11. ***Else***
12. *G_subgraph_cv1=connected_shortest_path(G, candidate_vert1)*
13. *G_subgraph_cv2=connected_shortest_path(G, candidate_vert2)*
14. *Selected_op_vert1*←*ECC_ECC (G_subgraph_cv1, candidate_vert1)*
15. *Selected_op_vert2*←*ECC_ECC (G_subgraph_cv1, candidate_vert2)*
16. *select_(op_vert)*←*select_(op_vert)*∪ *{Selected_op_vert1, Selected_op_vert2}*
17. *utility_ select_(op_vert)=max(select_(op_vert))*
18. *indx=argmax(utility_ select_(op_vert))*
19. *seed_sel*←*select_candidate[indx]*
20. ***Return*** *utility_ select_(op_vert), seed_sel*
21. ***Else***
22. *Rnd_rew*←*random(GS_Edge_Reward)*
23. *Indx_rnd*←*gmax(Rnd_rew)*
24. *candidate_vert1, candidate_vert2*← *Selected_edge[Indx_rnd]*
25. *select_candidate*←*select_candidate*∪ *{candidate_vert1, candidate_vert2}*
26. *Selected_op_vert1*←*len(graph[candidate_vert1])*
27. *Selected_op_vert2*←*len(graph[candidate_vert2])*
28. *select_(op_vert)*← *select_(op_vert)*∪ *{ Selected_op_vert1, Selected_op_vert2}*
29. *utility_ select_(op_vert)=max(select_(op_vert))*
30. *indx=argmax(utility_ select_(op_vert))*
31. *seed_sel*←*select_candidate[indx]*
32. ***Return*** *utility_ select_(op_vert), seed_sel*

A. SELECTION OF RELEVANT INFLUENCERS BASED ON THE MAB ϵ-GREEDY ALGORITHM

Algorithm 3, the "SRI_MAB epsilon-greedy" algorithm, adjusts the randomness in the selection of arms via an ϵ value that determines the probability of exploring new arms instead

Algorithm 3 Select_Relevant_Influencer SRI-MAB ϵ-greedy

Input: Graph G, α, β, ϵ, seed set size K, N
Output: Seed set S regret Ψ
1. ***Initialization:*** $S = \emptyset$, $\Theta = []$, $\Psi = []$, $\Phi = 0$, $\theta = 0$
2. ψ_EES, all_tried_arms =**ESBJ_Reward** (graph, $\alpha\beta$
3. n=size(graph_nodes)
4. t_1=size(all_tried_arms)
5. **while** (length(S)<K):
6. **For** t in 1: N
7. idx=argmax($\psi\_EES(t)$)
8. $\varphi_vtx^{secc}(t)$vert_sel=**MAB_ Edge Selection**(G,ϵ,$\psi_EES(t)$all_tried_arms)
9. Candidate_seed=vert_sel
10. **if** Candidate_seed not in S and len(S)<K:
11. S=SU{Candidate_seed }
12.

$$\theta(\mathrm{t}) = \theta(t-1) + (1-\epsilon) \left[\frac{t}{n}\psi\_EES\,(t) + \frac{t}{t\_1}\varphi\_vtx^{secc}\,(t)\right] + \epsilon[\sum\_t = 1^N \frac{t}{n+t\_1}P(rw\_rnd)(1+\phi\_vtx^{sdeg}\,(t))]$$

13. $\Theta(\mathrm{t}) = \Theta(t-1)U\theta(t)\}$
14. $\psi\_EES(t) = \mathrm{remove}\psi\_EES(\psi\_EES(t)[idx])$
15. **If** sizeΨ)<N:
16. Φ (t)= Φ (t-1)+max(Θ(t))$-\theta$(t)
17. $\Psi(\mathrm{t}) = \Psi(\mathrm{t}-1)\mathrm{U}\{\Phi(\mathrm{t})\}$
18. **End if**
19. **Else if** sizeΨ(t))>= and size(S)<K:
20. Candidate_seed = pick_rnd (all_tried_arms)
21. **if** Candidate_seed not in S:
22. S=SU{Candidate_seed}
23. **End if**
24. **End else**
25. **Return** S, Ψ

of exploiting available arms. The epsilon-greedy-based algorithm is a simple kind of multi-armed bandit algorithm that requires no knowledge of the history of exploration. However, it is hard to determine the ϵ value that assures the highest expected reward and the optimal choice of arms, and thus, the optimal key influencers. Therefore, the proposed algorithm takes as input the graph, parameters that control the computation of the strength-similarity edge score, the seed set size and the number of rounds. The SRI-MAB proceeds by initializing necessary variables and initializing the reward function with the output of algorithm 1, then, the size of the seed set is fixed, and the selection of the key influencers takes place for N rounds in which the objective is to maximize the reward and obviously, minimize the regret. After, that the number of steps reaches N rounds, the selection of key influencers is performed randomly, and then, a set of key influencers of size K and the regret obtained over N rounds is output.

B. REGRET ANALYSIS PERFORMANCE

In this section, we analyze the performance of the online algorithms proposed for identifying a set of influencers within the online social network. Thus, we derive an upper bound of regret for the proposed algorithm. Thus, this bound can be adapted to other proposed algorithms depending on the ϵ value that has a certain impact on the exploration vs exploitation tradeoff. Regret is an opportune strategy for determining whether an online algorithm is doing well. Obviously, the regret quantifies how an algorithm is performing on certain graphs. One of the classic methods is applying the algorithm with the best optimal reward parameters on the studied problem compared to the mean reward. More formally, the cumulative optimal arm (seed set) selection reward of the online algorithm is compared with the cumulative mean reward, which can be written as

$$\Phi\,(N) = \theta^* N - \sum\nolimits_{t=1}^{N} \theta(t) \tag{15}$$

The quantity θ^* refers to the reward for selecting the best arm, and thus, the best seed set and $\bar{\theta} = \sum_{t=1}^{N} \theta(t)$ is the cumulative mean reward.

We focus on analyzing the regret of the **SRI-MAB algorithm**

Therefore, by recalling the Hoeffding inequality, we have

Theorem 1 (Hoeffding Inequality): Let $X_1, X_2, \ldots, X_N$ be independent random variables with mean $\bar{\theta}$ and $X_i \in [a, b]\, with i = 1, 2, .., N$, so

$$P\left(|\theta^*\,(N) - \bar{\theta}\,(N)\,| \geq t\right) \leq 2\exp(-\frac{2\,Nt^2}{(\text{b-a})^2})$$

Proof of Theorem 1: can be found in [44]

Theorem 2: Let $0 < \epsilon < 1$, *and* $L^\epsilon \neq N$ *and* $K > 0$, *and the upper regret bound for our SRI-MAB algorithm is at most* $\frac{N}{N-L^\epsilon} + O\left(\sqrt{\frac{N}{2\,KL^\epsilon log\frac{(N-L^\epsilon)}{N(N+L^\epsilon)}}}\right)$

Proof of Theorem 2: Therefore, let $\theta^*_{a^*}$ be the average cumulative reward for selecting the best arm and thus selecting the seed set $a^* \in S \subset n$, $|S| = K$, and $a^* = \arg\max_{a \in S} \theta^*_a$

The main purpose is $|\theta^*(N) - \bar{\theta}(N)| < \delta$, wherein δ should be a small quantity, which shows the level of regret for not always playing the best arm. Therefore, considering an unknown distribution of the reward function of the reward obtained by selecting the best arm, and thus, a seed set that falls within the range [0, 1], in which we suppose that we have N cumulative rewards $\theta_1, \theta_2, \ldots, \theta_N$ and considering an interval $[\bar{\theta} - \delta, \bar{\theta} + \delta]$. Thus, by applying the Hoeffding inequality to show how the cumulative optimal reward is near the expected reward, Therefore, we have:

$$P\left(|\theta^*\,(N) - \bar{\theta}\,(N)\,| \geq \delta\right) \leq \gamma \tag{16}$$

By replacing the Hoeffding inequality in the term (16), we obtain

$$2\exp\left(-2\delta^2 N\right) \leq \gamma \tag{17}$$

$$\log 2 - 2\delta^2 N \leq \log\gamma \tag{18}$$

Therefore,

$$\delta = \sqrt{\frac{1}{2N} log\frac{2}{\gamma}} \tag{19}$$

Therefore, we obtain the confidence radius δ of our reward function. Thus, assuming that our reward functions of θ moves toward 1 as $t \rightarrow \infty$ as all reward functions including $\psi_{EES}, \varphi_{vtx^s}^{ecc}, and \phi_{vtx^s}^{deg}$ are normalized, so their maximum value will be 1.

Therefore, we estimate the value of γ that represents that the upper bound of the optimum of the cumulative reward near the true expected value.

Therefore, let N be the time horizon in which the marketer (user) determines which arm to select and let L^ϵ represent the number of times that the selection of arms follows the $(1-\epsilon)$ strategy and that l^ϵ refers to the number of times that the algorithm follows the ϵ strategy. Therefore, the first term requires $\gamma = \frac{nN}{L^\epsilon K} + \frac{Nm}{L^\epsilon K} + \frac{2\,N(n+m)}{(N-L^\epsilon)K} = \frac{N(m+n)(N+L^\epsilon)}{L^\epsilon K(N-L^\epsilon)}$,

Therefore, by substituting γ into δ, we obtain the following formula

$$\delta = \sqrt{\frac{1}{2N} log\frac{2L^\epsilon K(N-L^\epsilon)}{N\,(m+n)\,(N+L^\epsilon)}} \tag{20}$$

Thus, at the exploration phase, each arm is selected at least once, contributing to the regret with $\frac{N}{N-L^\epsilon}$ for all tested arms. In addition, the exploitation phase requires $O\left(\frac{N}{L^\epsilon K}\sqrt{\frac{1}{2\,N} log\frac{2\,L^\epsilon K(N-L^\epsilon)}{N(m+n)(N+L^\epsilon)}}\right)$

Therefore, our regret bound consists of two parts, $m \leq KL^\epsilon$ *and* $n \leq KL^\epsilon$, since each arm and each vertex can return at most KL^ϵ regret for choosing an arm, and thus, a seed set.

$$\begin{aligned} w\,(N) &\\ = |\bar{\theta}^*\,(N) - \theta\,(N)\,| &\leq \gamma^* \leq \frac{N}{N-L^\epsilon} \\ &+ O(\frac{N}{L^\epsilon K}\sqrt{\frac{KL^\epsilon}{2N} log\frac{2L^\epsilon K\,(N-L^\epsilon)}{N\,(m+n)\,(N+L^\epsilon)}}) \quad (21)\\ &\leq \frac{N}{N-L^\epsilon} + O(\sqrt{\frac{N}{2KL^\epsilon} log\frac{2L^\epsilon K\,(N-L^\epsilon)}{N\,(m+n)\,(N+L^\epsilon)}}) \quad (22)\\ &\leq \frac{N}{N-L^\epsilon} + O\left(\sqrt{\frac{N}{2KL^\epsilon} log\frac{2L^\epsilon K\,(N-L^\epsilon)}{N2L^\epsilon K\,(N+L^\epsilon)}}\right) \quad (23)\\ &\leq \frac{N}{N-L^\epsilon} + O\left(\sqrt{\frac{N}{2KL^\epsilon} log\frac{(N-L^\epsilon)}{N\,(N+L^\epsilon)}}\right) \quad (24) \end{aligned}$$

C. THE GREEDY ALGORITHM FOR SELECTING RELEVANT INFLUENCERS SRI_CGSS FEXPL-GREEDY: FULL EXPLOITATION ($\epsilon = 1$)

The SRI_CGSS fexpl-greedy algorithm is a version of the ϵ-greedy algorithm with $\epsilon = 1$, that prioritizes the exploitation with a high rate compared to the exploration with negligible value. Therefore, the pure greedy algorithm for the selection of relevant influencers is based on the previous algorithms namely, algorithm 1, which feeds the multi-armed bandit algorithm with the initial reward function and algorithm 2, which determines the optimal arms, and thus, the optimal "relevant influencers" depending on the variation in the obtained reward and the value of ϵ. In this case, we can determine the steps for the selection of relevant influencers SRI_CGSS fexpl-greedy from the pseudocode of algorithm 3 by setting $\epsilon = 1$. The main difference here is that the algorithm is primarily exploiting the best arms and exploring at a low rate. Then, it selects some relevant influencers by counting the local neighborhood, and the remaining steps are performed by a global metric eccentricity to select all remaining arms. Thus, with this strategy, the algorithm mostly uses global structural properties to choose the candidate seed set.

D. THE GREEDY ALGORITHM FOR THE SELECTION OF RELEVANT INFLUENCERS SRI_CGSS FEXPR-GREEDY: ALMOST EXPLORING ($\epsilon = 0$)

The second version of the "SRI-MAB ϵ-greedy" algorithm 4, named "SRI_CGSS fexpr-greedy" is when the algorithm primarily explores where the arms are selected randomly, and the seed set is selected according to the number of neighbors. Thus, the routine for the algorithm follows the same strategy while putting $\epsilon = 0$, and thus, extracts required relevant influencers with the size of the seed set equal to K and regret over N rounds.

Analyzing the algorithm consists of an essential part to prove the efficiency and effectiveness in terms of required tasks given an input and a series of instructions. In the following subsections, we provide the time complexity and space required by the SRI-MAB ϵ-greedy algorithm.

E. TIME COMPLEXITY

The time complexity for an algorithm expresses the total amount of time required to complete the computation of the problem described by the algorithm instructions. Therefore, by analyzing the algorithm in terms of time complexity, we can quantify to what extent the algorithm performs well independent of resources and hardware. The first line 1 initializes five algorithm variables which requires at most $O(5)$, so the first line has a constant time complexity. The second line calls algorithm 2 to output the initialization of the reward matrix of the multi-armed bandit and corresponding pulled arms (selected edges) that requires at most $O(n^2 d + n^2 \log n + mn + m)$ time. The first part of algorithm 3 from line 1 to line 4 requires at most $O(n^2 d + n^2 \log n + mn + m)$ time to initialize the reward matrix of the multi-armed

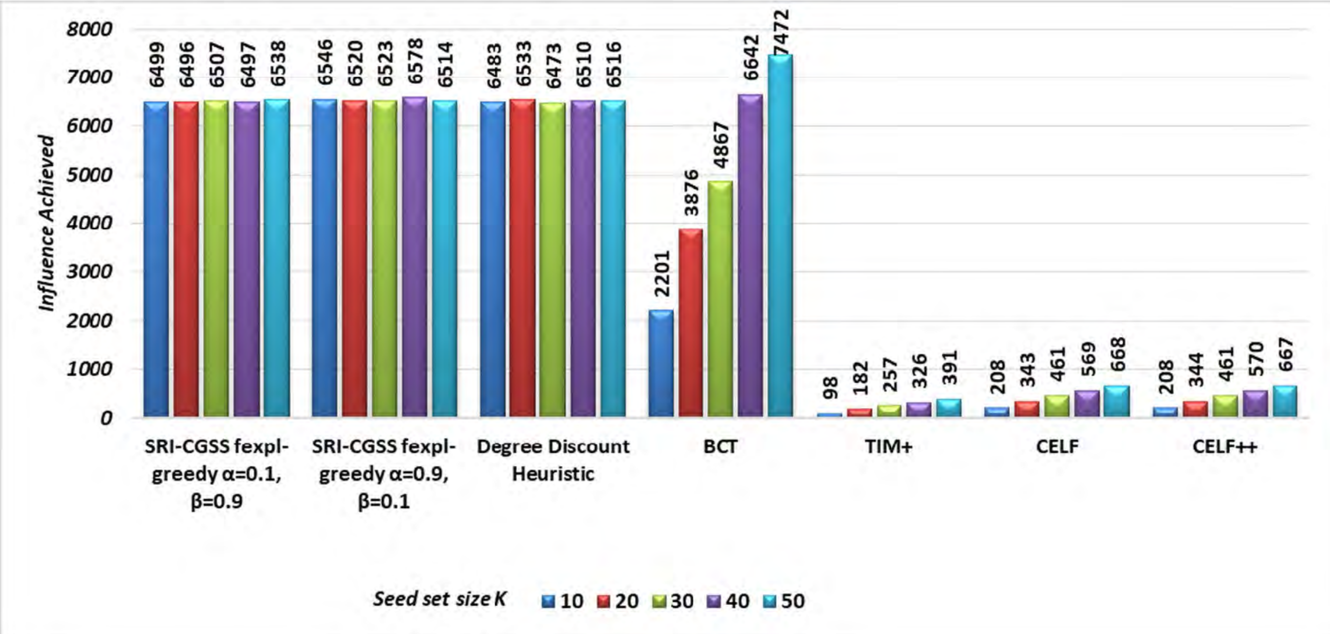

FIGURE 3. Influence spread under the IC model on Nethept data.

bandit algorithm and essential variables. Then, the remaining instruction that follows while and the inner for loops requires at most $O((Nn\,(m\log n + 1)\log K))$ time.

Overall the time complexity required by the SRI-MAB ϵ-greedy algorithm is $O(n^2d + n^2\log n + mn + m + Nn\,(m\log n + 1)\log K)$ to complete the computation of the seed set of size K and regret Ψ of size N.

F. SPACE COMPLEXITY

The space complexity of an algorithm can be seen as the total memory required by the algorithm pseudocode to run. It serves to assess how much pseudocode requires for storage to run promptly. The SRI-MAB algorithm requires one unit of storage for each variable for initialization (Line 1, lines 3-4). The rest of the algorithm requires $O(m + n^2)$ of storage to run. Knowing that most functions used by the algorithm need at most $O(m + n^2)$ space including $\psi_{EES}, \varphi^{ecc}_{vtx^s}$. Therefore, the overall space complexity needed by the algorithm is $O(n^2 + m)$.

VI. EXPERIMENTAL RESULTS

The main purpose of this section is to test the effectiveness and performance of the proposed algorithms and thus evaluate the selection of a seed set of size K and show how regret varies with respect to$N$ rounds. The major goal here is to assess the three proposed algorithms based on global structural properties and three variants of ϵ-greedy algorithms against the state-of-the-art approaches in terms of the influence achieved, running time required and the necessary space complexity needed by each algorithm to complete the selection and running the diffusion models. Additionally, we are interested in showing the cumulative regret of our algorithms and the impact of using the greedy strategy from the MAB algorithms and reward parameters used for initialization. We also assigned different values to ϵ to compare its impact on the influence spread under the IC and LT models.

TABLE 1. Data characteristics.

Datasets	*Nodes*	*edges*	*Type*	*Diameter*
Nethept[1]	15 233	58 891	Undirected	14
Netphy[1]	3715	1965	Undire	12
	4	91	cted	
Email-EuAll[2]	265,214	420,045	Directed	14
munmun_twitter_social(Twitter)[3]	465,017	834,797	Directed	8

Table 1 represents characteristics of the dataset used throughout the experiments section. All experiments are conducted on a Linux server 28 CPU with 112 GB memory.

In this section, we present the experimental results for three proposed algorithms based on the multi-armed bandit including nearly full exploitation with negligible exploration, nearly full exploration with insignificant exploitation and the last algorithm is when the algorithm is exploring and exploiting with a certain value ϵ. We fixed the number of rounds to N $=$ 100, and evaluated the proposed algorithms in terms of the influence achieved compared with existing work in which we fixed $\alpha = 0.9, \beta = 0.1 (\alpha = 0.1, \beta = 0.9)$ for all algorithms and datasets. Thus, we compared its efficiency

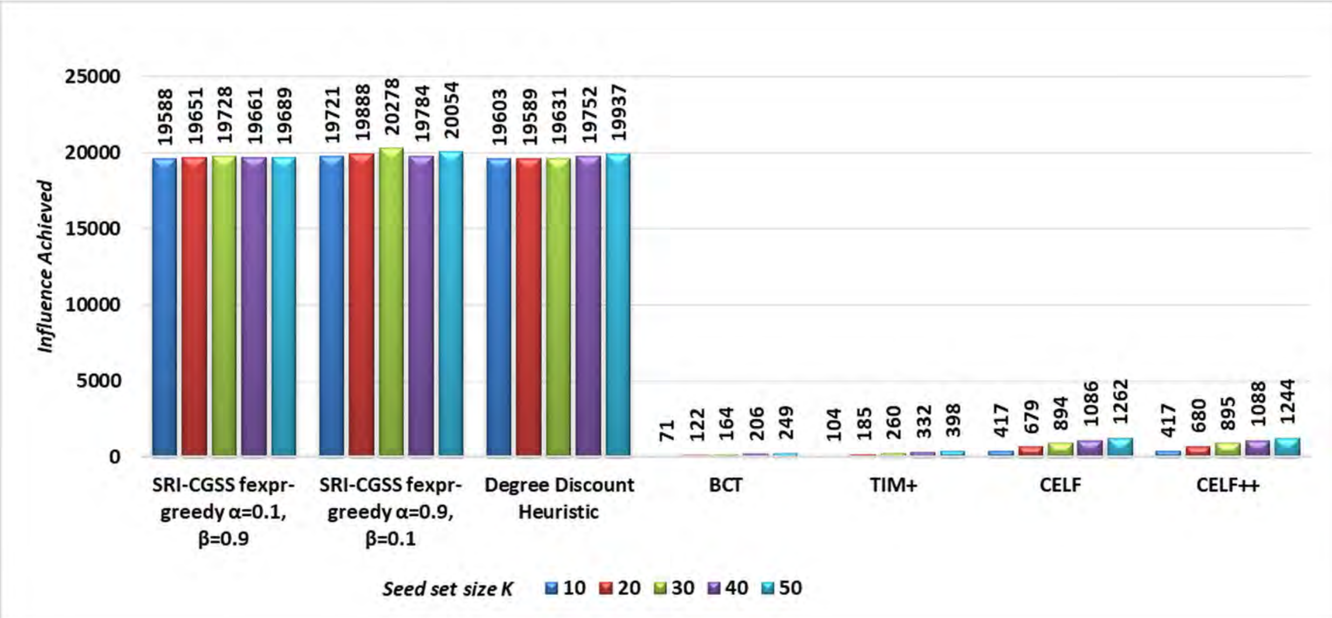

FIGURE 4. Influence spread under the IC model on Nephy data.

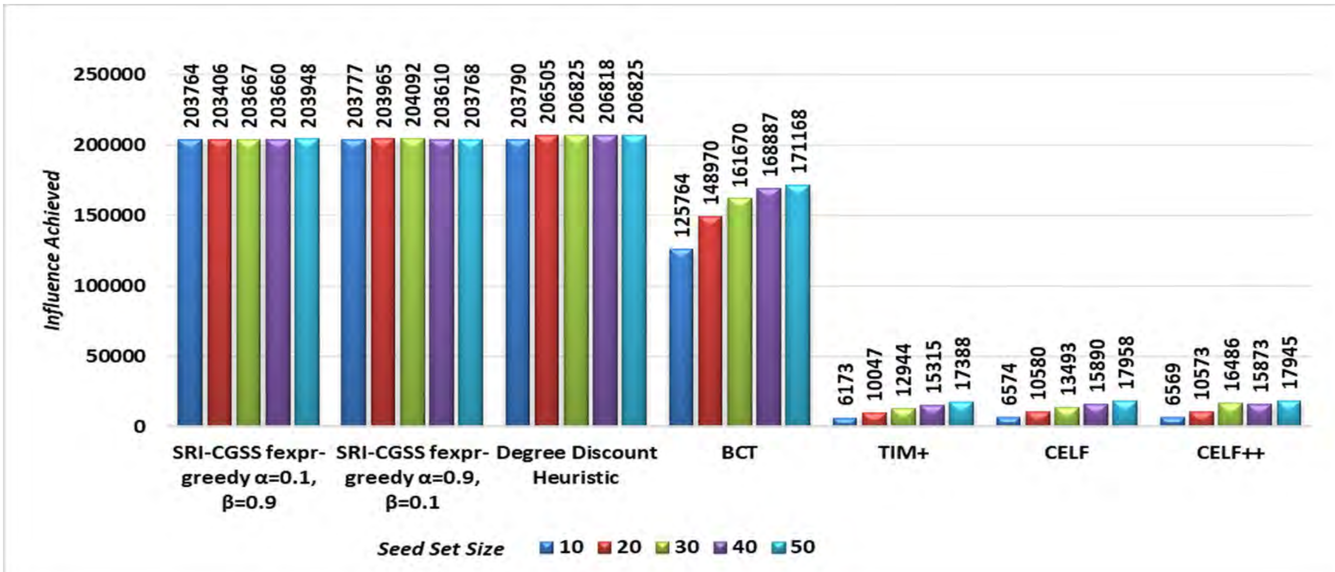

FIGURE 5. Influence spread under the IC model on Eu-Email data.

in terms of time complexity and storage space needed. Thereafter, we conducted experiments for each algorithm in terms of cumulative regret and influence spread under the IC model, where the best parameters that provide the highest spread of influence are selected.

We compare our proposed algorithms against some well-known algorithms including degree discount heuristic "DDH" [9], BCT [45], TIM+ [46], CELF [7], and CELF++ [8].

In the following, for each algorithm, we present the influence achieved, time, and storage space required compared with existing approaches. Next, for each algorithm, we vary the multi-armed bandit initialization reward with different values and see its impact on the influence achieved as well as on the cumulative regret for each initialization reward value.

A. THE GREEDY ALGORITHM (SRI-CGSS FEXPL-GREEDY) ALMOST EXPLOITING ($\epsilon = 1$)

1) INFLUENCE SPREAD UNDER THE IC MODEL ON THE FOUR DATASETS

Figure 3, Figure 4, Figure 5, and Figure 6 represent the influence spread achieved versus the seed set size under the IC model. We can observe clearly in Figure 3, Figure 4, and Figure 6 that the proposed algorithm outperforms all

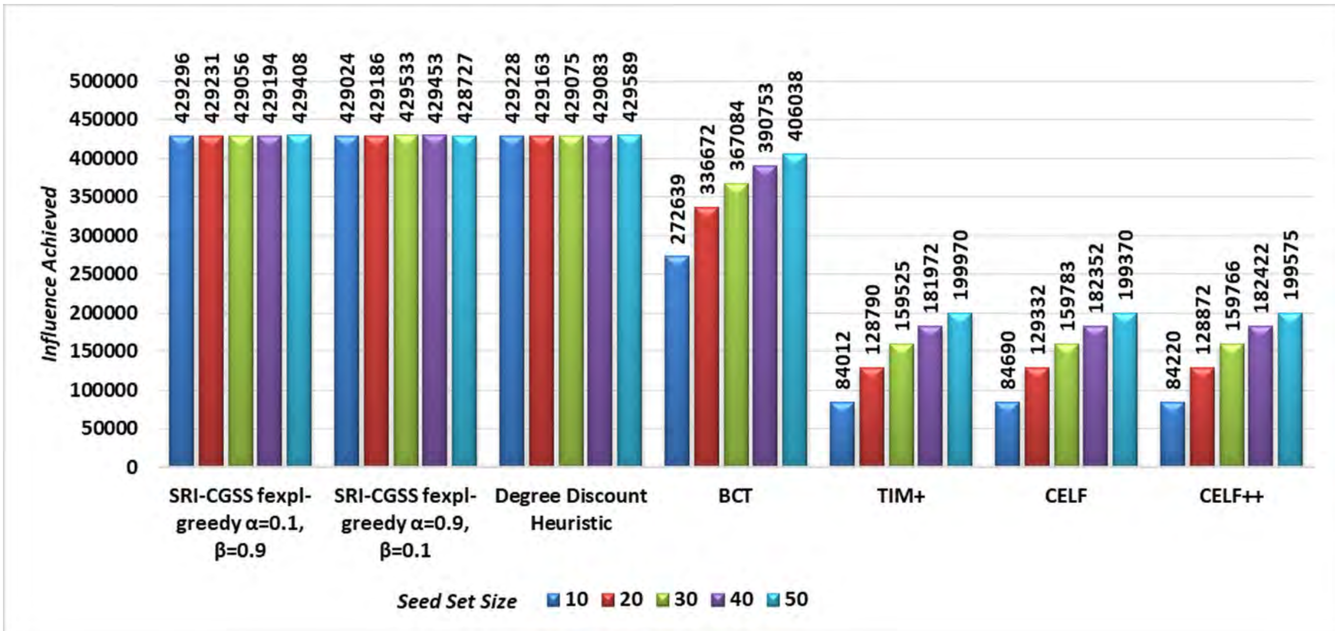

FIGURE 6. Influence spread under the IC model on Twitter data.

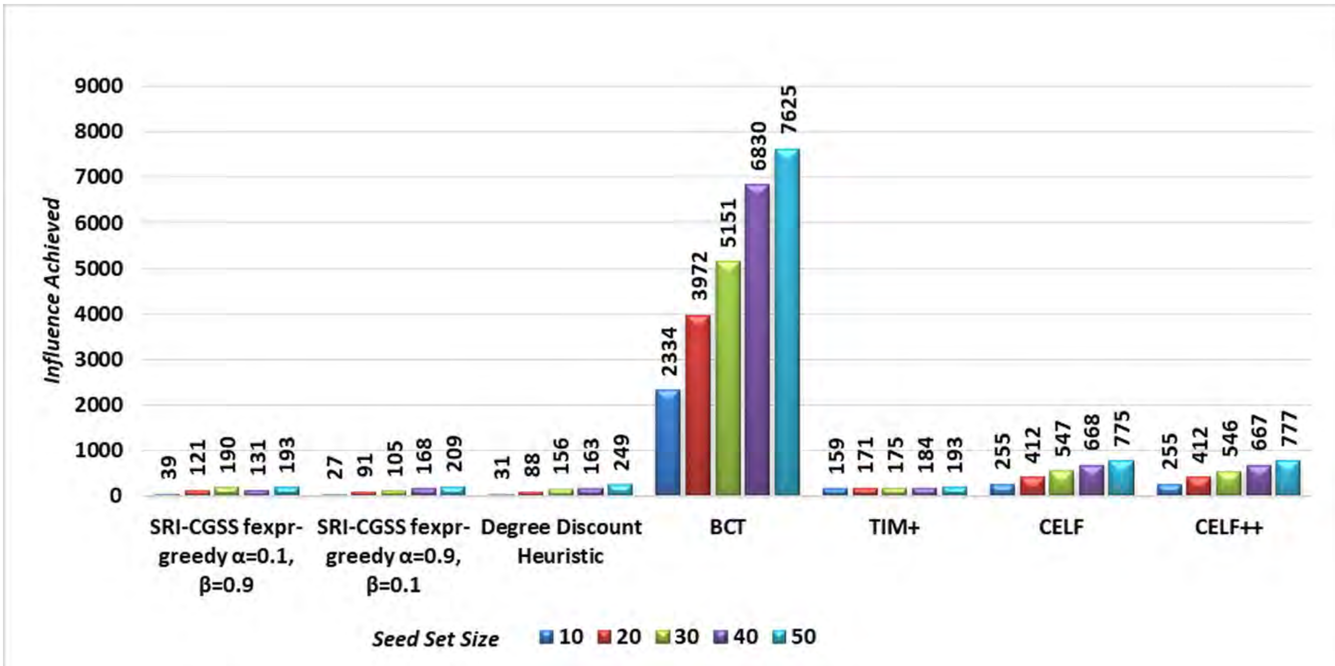

FIGURE 7. Influence spread under the LT model on Nethept data.

algorithms in terms of influence spread and is especially better than the ''DDH'' algorithm by 0.0066% for NetHept data. Likewise, the proposed algorithm for full exploitation (pure greedy with negligible exploration "SRI-CGSS fexpl-greedy" is better than all algorithms, especially when the algorithm relies more on the similarity of the immediate shared neighbors "$\beta = 0.9$"that provide the best results in terms of the number of touched users that surpasses the DDH algorithm with 0.01% for the NetPhy data. Similarly, for the Twitter data, the spread of influence is better overall than all other algorithms with 0.810^{-5}% which is a small difference between our algorithm compared with the "DDH". However, "SRI-CGSS fexpl-greedy" performed better than other algorithms except for the DDH algorithm for the Eu-Email data, which is lower by 0.005%, which seems to be a small amount that can be improved by adjusting the value of the reward

TABLE 2. Time and space complexity under the IC model for seed set size K = 50.

	Algorithms	*SRI-CGSS fexpr-greedy α=0.1, β=0.9*	*SRI-CGSS fexpr-greedy α=0.9, β=0.1*	*Degree Discount Heuristic*	*BCT*	*TIM+*	*CELF*	*CELF++*
NetHept data	*Time (S)*	637.89	725.50	4.21	0.01	5.02	649.80	808.80
	Space complexity(MB)	133.60	135.20	62.40	23.06	848.47	21.37	22.01
NetPhy data	*Time (S)*	4201.88	4812.95	51.96	0.67	14.18	7168.98	7176.60
	Space complexity(MB)	295.20	314.40	162.40	120.28	691.56	32.54	34.21
Eu-Email data	*Time (S)*	3949.06	3745.37	7584.11	0.40	13.70	27382	27659
	Space complexity(MB)	1200	1200	632	79.72	79.78	88.98	101.18
Twitter data	*Time (S)*	8924.45	9024.85	11470.08	0.03	2.63	1169802	1219404
	Space complexity(MB)	1280	1288	840	107.11	192	9826.95	188.074

function which will be discussed later in this section.

$$(\epsilon = 1)$$

Table 2. shows the running time and space complexity under the IC model for four studied datasets. As we can observe in Table 2, our algorithm consumes more time compared with some existing algorithms including DDH, BCT, TIM+, but as seen from previous figures, it achieves a higher influence spread. However, our algorithms require much less time than CELF and CELF++. Also, compared with DDH algorithm on big datasets such as Eu-Email and Twitter, we can notice that the proposed algorithm require less time. This increase in running time can be justified by the fact that our method relies on global centrality measures that we tried to optimize its running time and that the simulation of selecting the seed set requires 100 rounds to complete the identification of the seed set. Comparably we can see from the above Table that the proposed algorithm required more storage space than the existing approaches especially for the Eu-Email and Twitter dataset and that it requires less storage space for NetHept and NetPhy data compared with the TIM+ algorithm.

2) INFLUENCE SPREAD UNDER THE LT MODEL ON THE FOUR DATASET

Figure 7, Figure 8, Figure 9, and Figure 10 report the influence spread achieved for the four datasets under the LT model. We note that the proposed approach is designed primarily for the IC model which operates differently than the LT model. So, from the above figures, we can see that our algorithm performed better than the DDH and BCT algorithms on the NetPhy data. However, our approach needs improvement in terms of tuning parameters to adapt it to the LT model.

Table 3. presents the time and space complexity needed by the proposed algorithm with full exploitation and negligible exploration compared with existing approaches under the LT model. We can deduce from the algorithm results that we have the same time and space requirements for the selection of the seed set over 100 rounds and then the LT is performed. From this, we note the same behavior as the IC model, and thus, the proposed approach consumes less time than CELF and CELF++ and it requires less storage space than TIM+ for the NetHept and NetPhy data. In addition, since the TIM+ algorithm is based mainly on a reverse reachable set that attempts to determine the number of RR sets and then produces the sets that require less time and space, especially for the large graph. In addition, BCT requires less time and space than TIM+ since it already stores cost and benefit graphs, which generate a reasonable sample which makes it more scalable in terms of time and space on a large graph. However, CELF and CELF++ is an improvement in time complexity of the greedy algorithm [6], and despite the improvement in running time, those algorithms cannot run on the large graphs

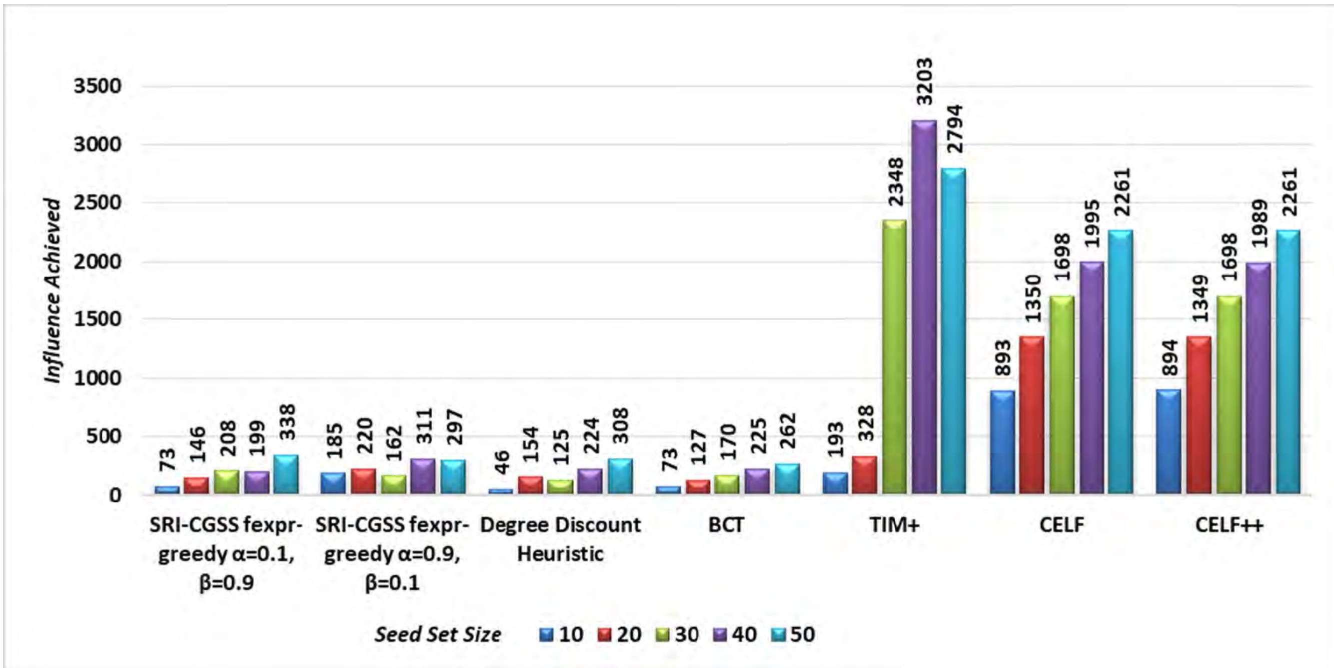

FIGURE 8. Influence spread under the LT model on Netphy data.

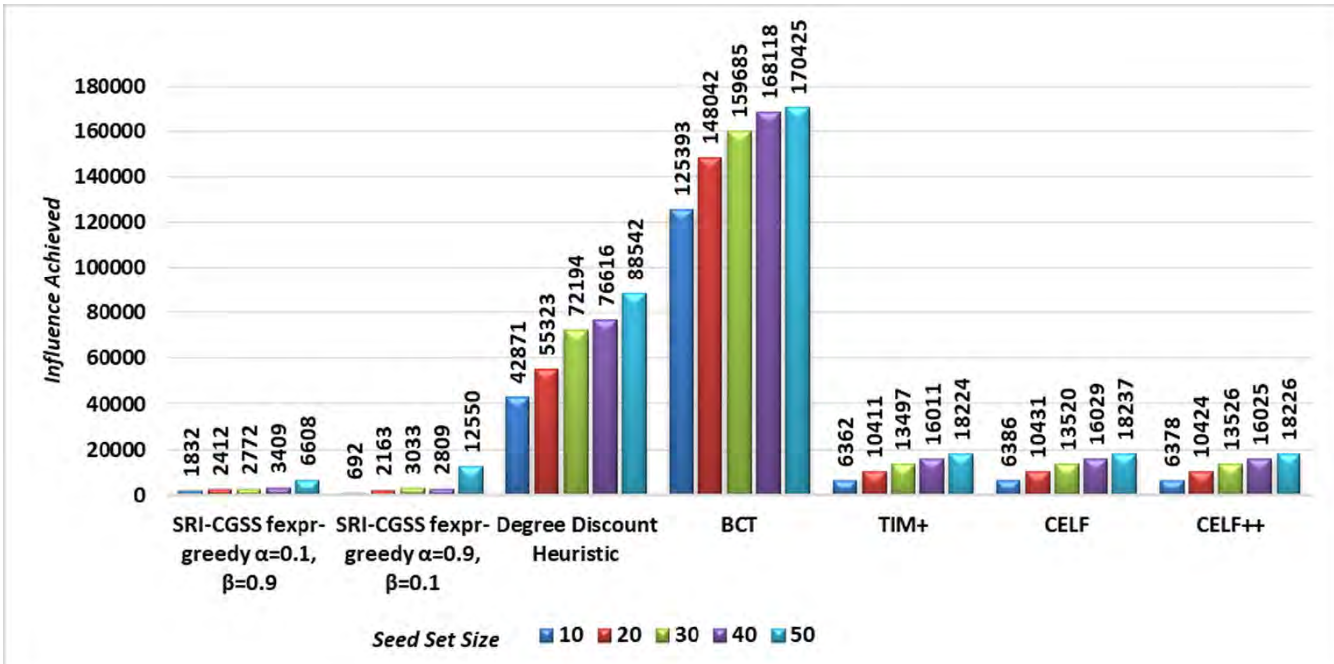

FIGURE 9. Influence spread under the LT model on Eu-Email data.

that are depicted in our experiments and require a large time to complete the computation of the seed set and the influence achieved.

3) CUMULATIVE REGRET ON FOUR STUDIED DATASETS

This subsection concentrates on the study of the cumulative regret when choosing arms over N = 100 rounds by taking the optimal arms (i.e., edges). The main objective of this subsection is to know which values of the initialization reward function will lead to less cumulative regret. More importantly, we are interested to see how the followed exploitation with negligible exploration strategy performed in terms of selecting the best arm, and thus, incurred the minimum cumulative regret. So, specifically, here we present some experiments regarding the proposed algorithm "SRI-CGSS fexpl-greedy" for the selection of relevant influencers with full exploitation and insignificant exploration based on edge strength and edge members similarity which assesses, at the same time, two interesting aspects for the selection of relevant influencers "seed set", which identifies the edges that are in the

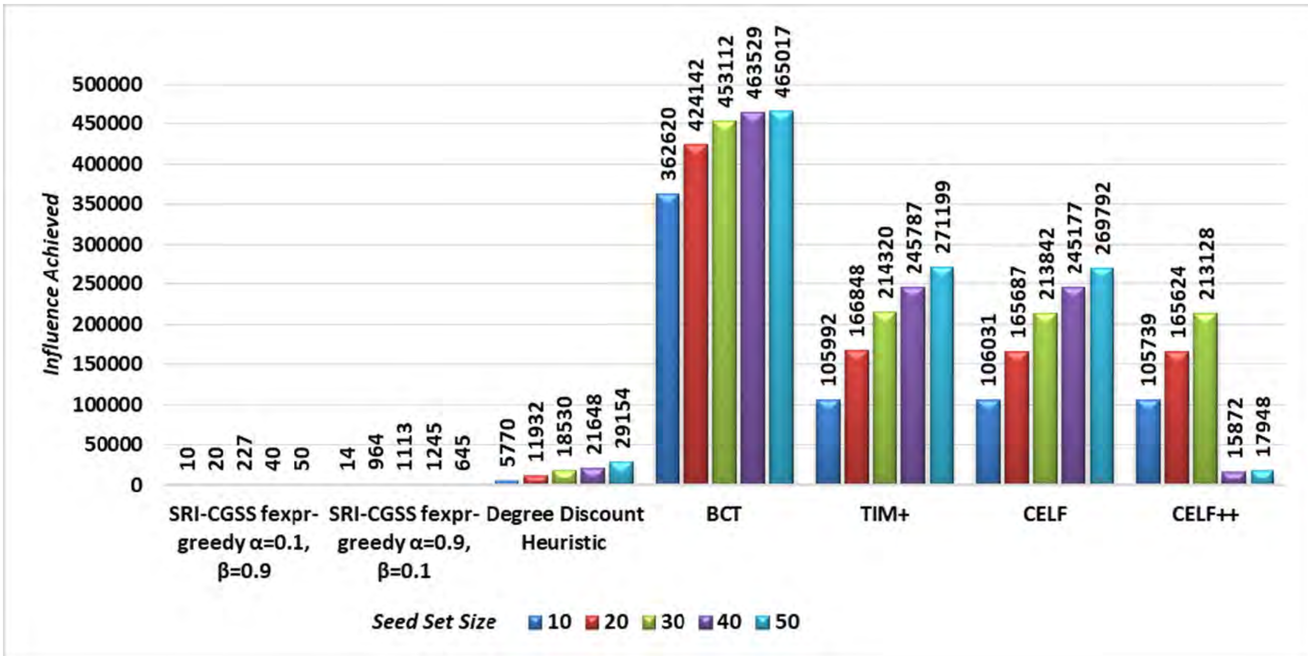

FIGURE 10. Influence spread under the LT model on Twitter data.

TABLE 3. Time and space complexity under the LT model for seed set size K = 50.

	Algorithms	*SRI-CGSS fexpr-greedy α=0.1, β=0.9*	*SRI-CGSS fexpr-greedy α=0.9, β=0.1*	*Degree Discount Heuristic*	*BCT*	*TIM+*	*CELF*	*CELF++*
NetHept data	*Time (S)*	*509.86*	*327.66*	*0.78*	*0.01*	*2,03*	*1111.80*	*1237.80*
	Space complexity(MB)	*135.20*	*135.20*	*72*	*23.06*	*730.66*	*21.40*	*22.12*
NetPhy data	*Time (S)*	7284.19	6418.68	4.67	0,603	2.08	7915.8	8733.6
	Space complexity(MB)	324.80	324.80	162.40	120.44	394.60	32.89	34.53
Eu-Email data	*Time (S)*	*18585.08*	*23667.40*	*11.19*	*0,02*	*5.15*	*36168*	*38497*
	Space complexity(MB)	*1296*	*1296*	*632*	*79.78*	*159.89*	*90.11*	*101,94*
Twitter data	*Time (S)*	*5577.1*	*6763.05*	*54.41*	*0.03*	*2.06*	*1491786*	*30161*
	Space complexity(MB)	*1512*	*1512*	*54.41*	*107.10*	*191.465*	*101.29*	*120.23*

central position and how likely the edge members are similar. As mentioned earlier we conducted various experiments to determine which parameter fits best for the algorithm that outputs a lower cumulative regret over four datasets presented above.

Figure 11, Figure 12, Figure 13, and Figure 14 represent the cumulative regret over N rounds for the SRI-CGSS fexpl-greedy algorithm with distinct parameter values for reward initialization. We can clearly observe from the figures that the values of the reward parameters α and β of the

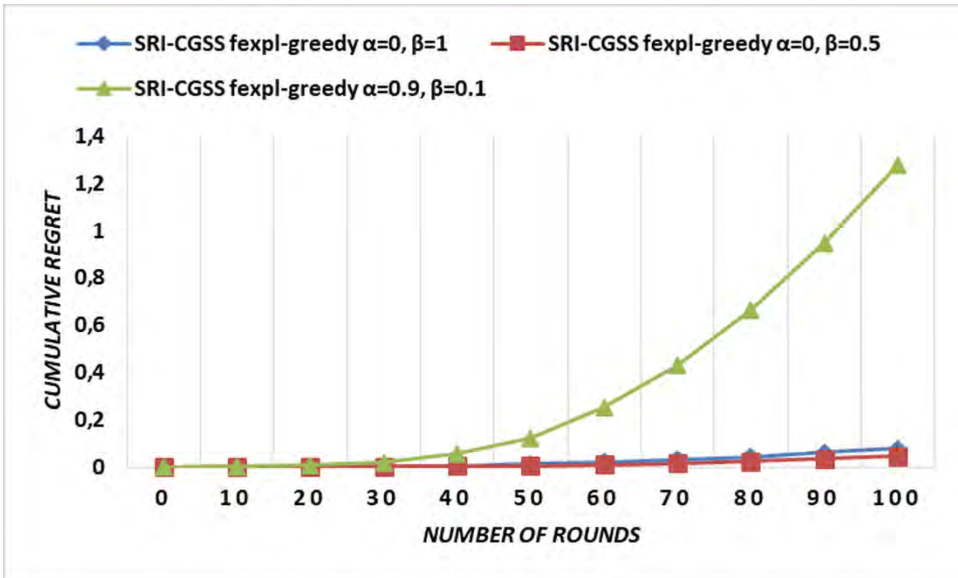

FIGURE 11. Cumulative regret on Nethept data versus the number of rounds.

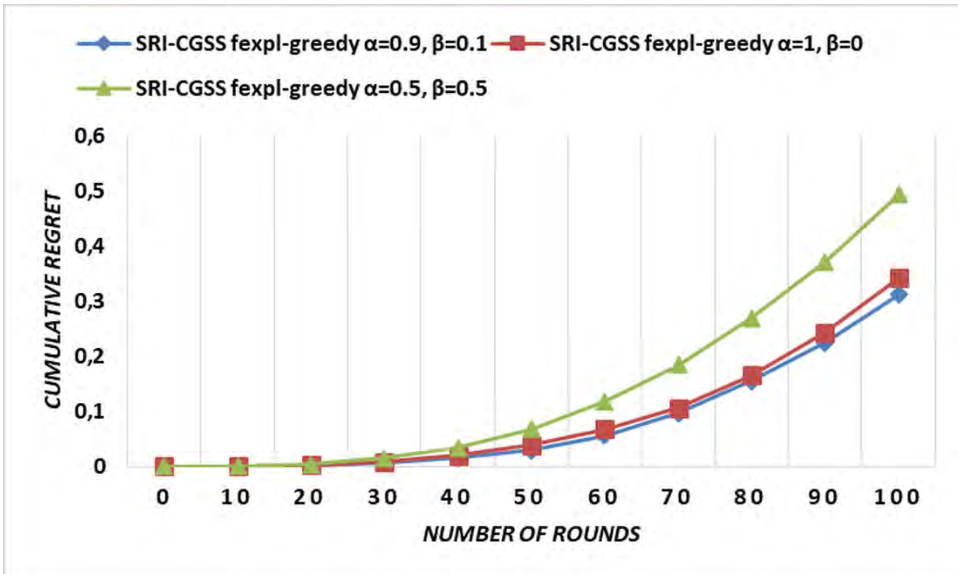

FIGURE 12. Cumulative regret on Netphy data versus the number of rounds.

algorithm that provides a low cumulative regret differ depending on the studied graph. More specifically, SRI-CGSS fexpl-greedy algorithm provides low cumulative regret when edge members have a certain tendency to be similar to each other without addressing the edge position in the network on the NetHept data. However, SRI-CGSS fexpl-greedy algorithm provides satisfactory results manifested with the lowest regret, when the algorithm selects over N round edges based only on the positions of the edges on the Eu-Email. Additionally, we notice that it is important to have a central position for selected edges with a negligible consideration of the members' similarity rate on NetPhy data. Unlikely, on the NetPhy data, the algorithm performs well on the Twitter data when the edges are selected according to edge members' similarity.

4) IMPACT OF REWARD FUNCTION ON INFLUENCE SPREAD

In this subsection, we try distinct values of α and β and observe its impact on the influence spread. More precisely, we are interested in showing how the parameters of the reward function of the multi-armed bandit algorithm impact the decision of arms selection, and thus, nodes that maximize the spread of promoted products and behaviors.

Figure 27, Figure 28, Figure 29, and Figure 30 display the influence achieved under the IC model versus the seed

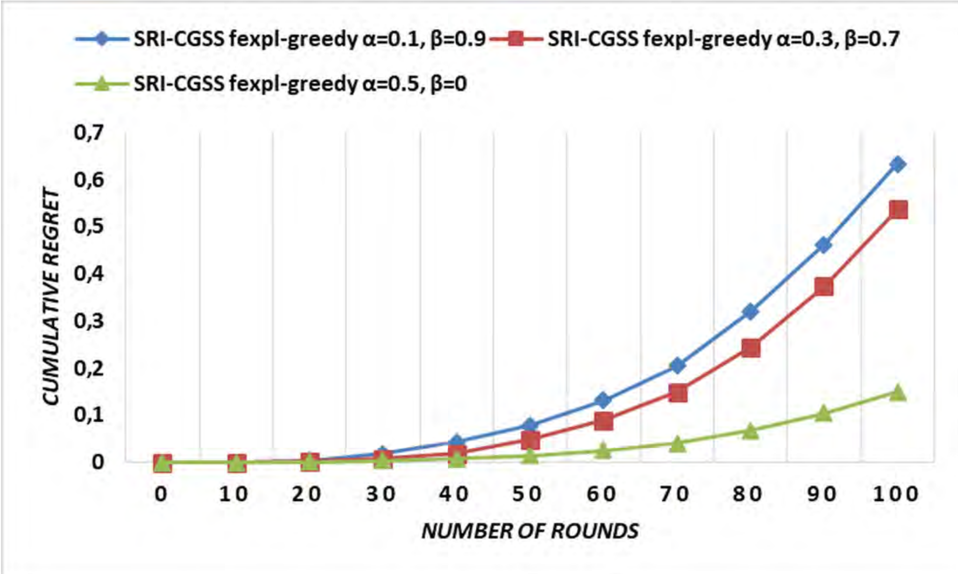

FIGURE 13. Cumulative regret on Eu-Email data versus the number of rounds.

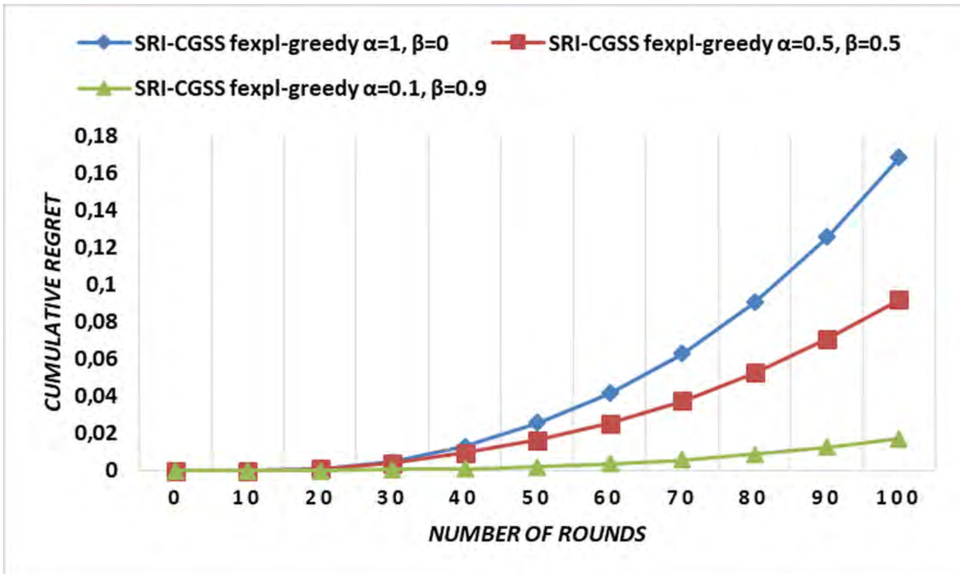

FIGURE 14. Cumulative regret on Twitter data versus the number of rounds.

set size when the selection of arms is performed by mainly exploiting and exploring few times. We notice that the algorithm performed very well on undirected graphs while the directed setting required some adjustment to provide higher influence coverage. We observe that our algorithm performed very well on the NetHept data and increased the number of touched users compared with the previous setting ($\alpha = 0.9$, $\beta = 0.1$), while it seemed promising that if adjacent users were at some extent similar, it presented a high influence spread on the NetHept data which confirmed that performing the choice of seed set using the similarity rate was better and provided less cumulative regret, which shows how we regret the choice of wrong arms and thus, the seed set. Similarly, on the results reported on NetPhy data, it can be seen clearly that the influence was higher when selected edges are central without focusing on how similar the edge members are. In the same sense, the influence coverage is higher on the Eu-Email data, and when choosing the seed set according to the edge position, its cumulative regret is the lowest. Different than undirected data, and Eu-Email, we notice in Figure 30 that it gives good results when the seed set is selected based primarily on the position of the edges, i.e., edges with a central position ''how edges are located among other edges on Twitter data''.

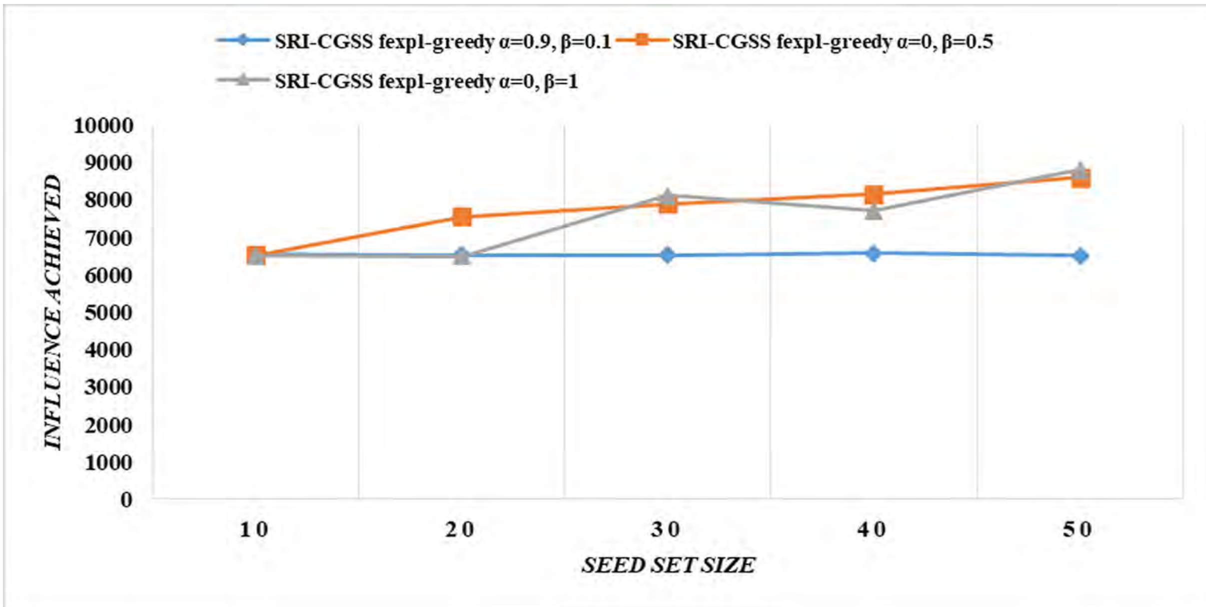

FIGURE 15. Impact of varying α and β on influence spread on Nethept data.

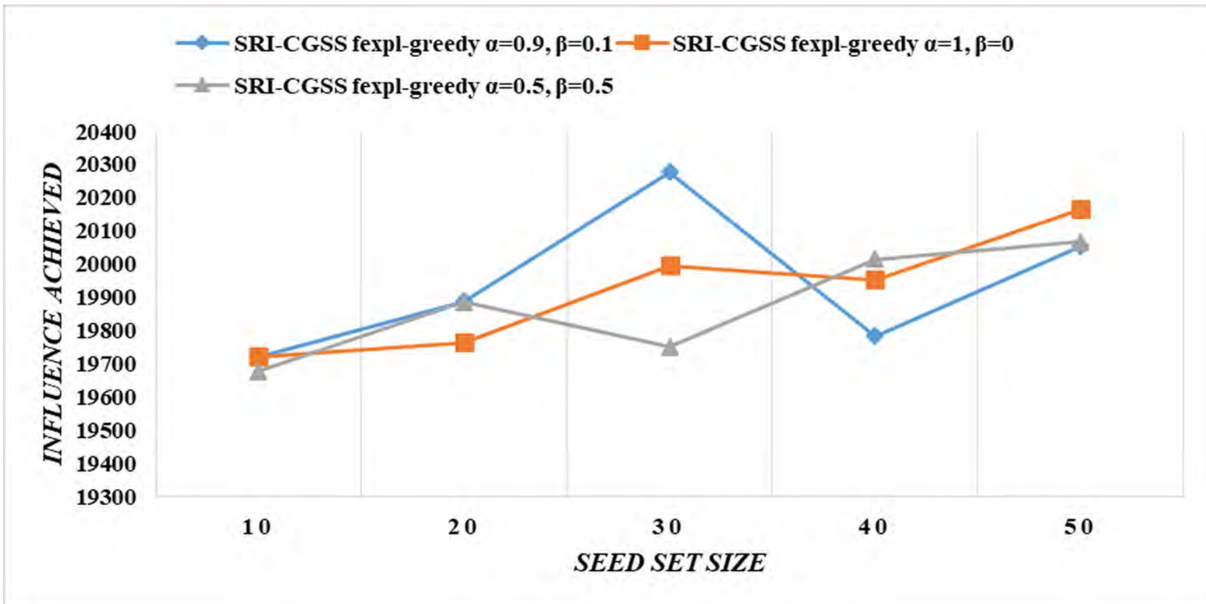

FIGURE 16. Impact of varying α and β on influence spread on Netphy data.

B. GREEDY ALGORITHM (SRI_CGSS FEXPR-GREEDY): ALMOST EXPLORING ($\epsilon = 0$)

We concentrate here on presenting results regarding the second algorithm, SRI_CGSS fexpr-greedy, which focuses mainly on exploring with a selection of arms randomly, while the difference is that the selection of relevant influencers is chosen based on some centrality measures and exploiting rarely the best arms with the highest reward.

Figure 19, Figure 20, Figure 21, and Figure 22 present the influence achieved by our algorithm with the full exploration and negligible exploitation "SRI-CGSS fexpr-greedy "compared with the previous algorithms on four studied datasets under the IC model. From all figures, we observe that "SRI-CGSS fexpr-greedy" algorithm perform better overall in terms of influence spread than all existing algorithms on the NetHept, NetPhy, and Twitter data, whereas the full exploration with few rounds of exploitation has lower performance by 0.005% than the DDH algorithm on Eu-Email data.

Table 4 reports the time and space complexity required on the four datasets under the IC model. We notice that the proposed algorithm requires less time than CELF and

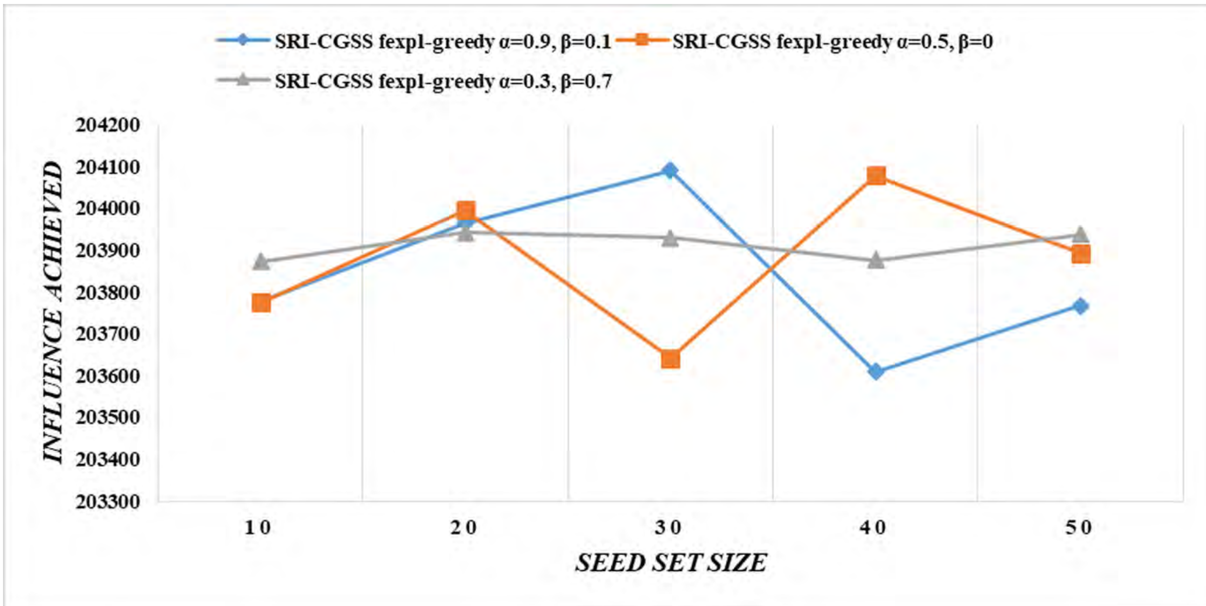

FIGURE 17. Impact of varying α and β on influence spread on Eu-Email data.

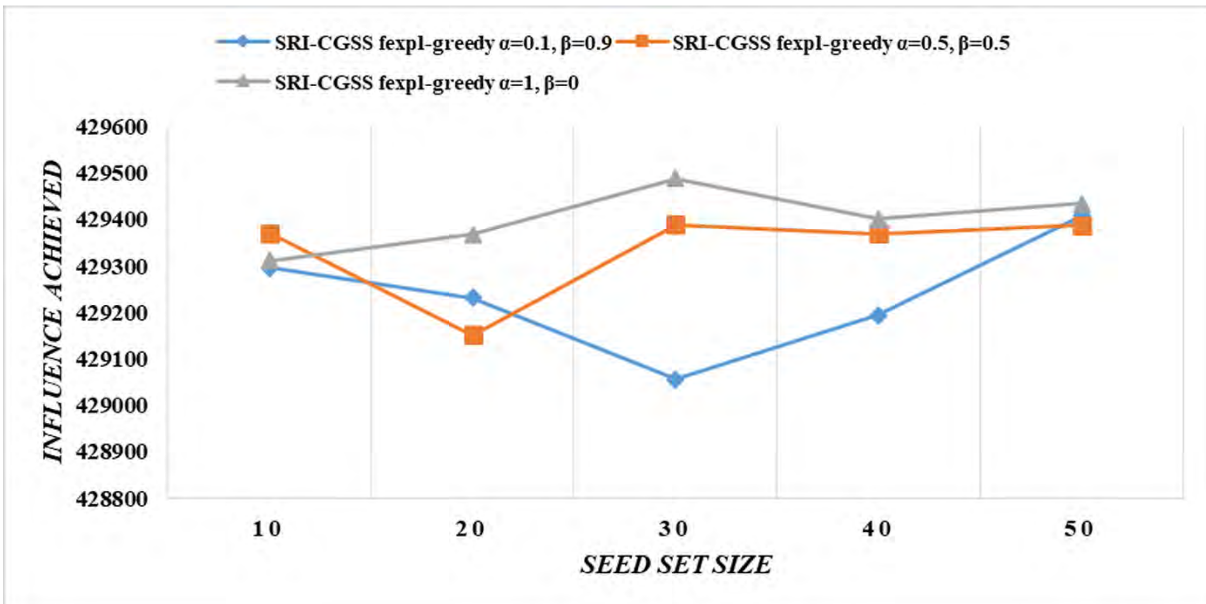

FIGURE 18. Impact of varying α and β on influence spread on Twitter data.

CELF++ on all datasets and less time than DDH on the Twitter and Eu-Email data. In addition, the proposed algorithm demonstrates that it requires more space storage than TIM+ on two undirected graphs while it needs more storage on a large directed dataset including Twitter and Eu-Email. As per justification and as discussed earlier that each algorithm requires certain conditions to be carried on as DDH uses simple degree centrality with discount which makes it a method based local centrality measure and thus requires less computational complexity than our proposed algorithms. However, sometimes DDH has a higher time complexity when conducting experiments on large scale data than our proposed algorithms, this may depend on graph structure and how each algorithm needs to use graph properties. After all, despite that, the DDH is a method based local network topology provides a low spread efficiency on some datasets compared with our proposed algorithms that are based on global network properties. Additionally, our methods perform very well on some graphs compared to all other algorithms such as NetHept data and perform a little bit efficient on all other datasets in term of influence spread and even has a reasonable running time over N rounds.

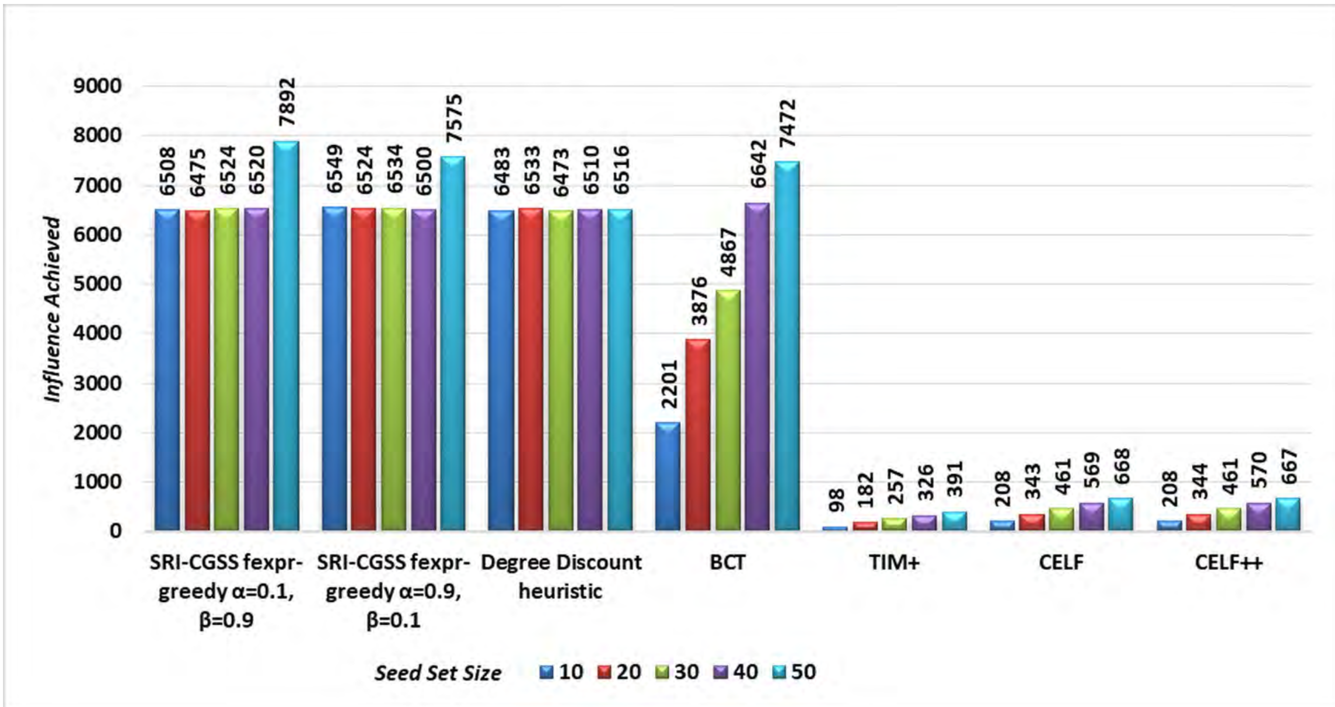

FIGURE 19. Influence spread under the IC model on Nethept data.

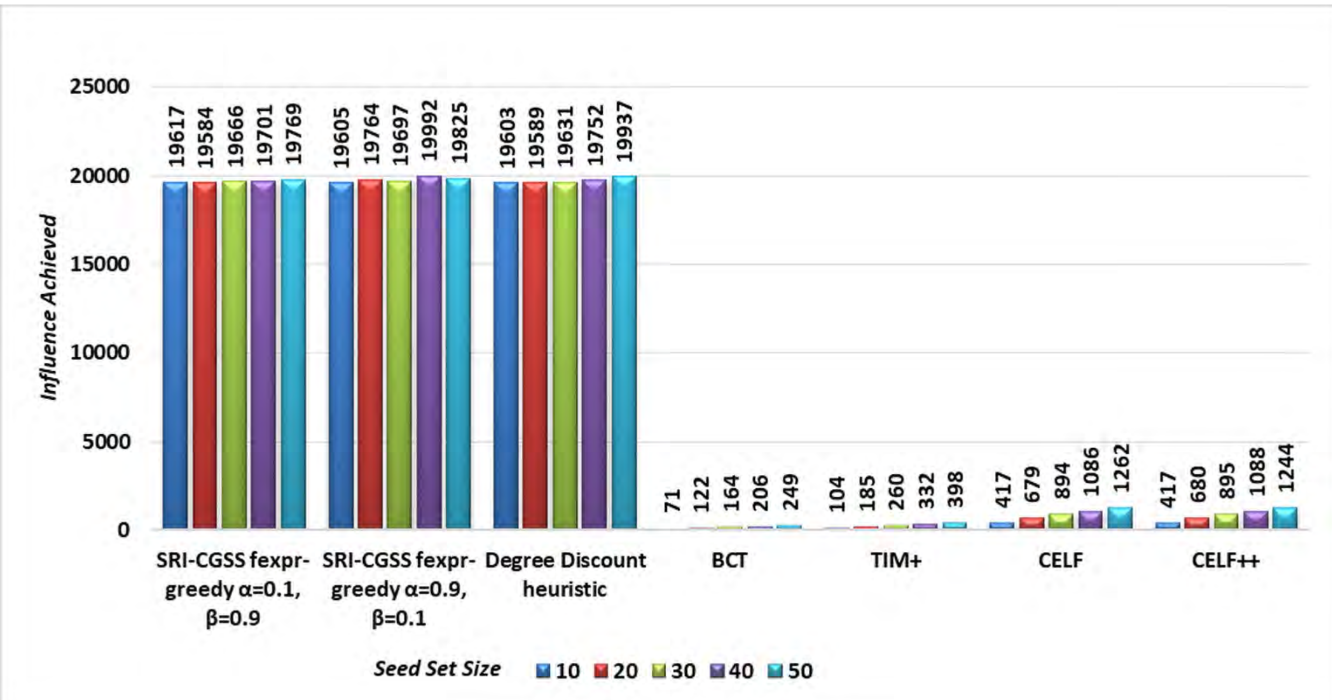

FIGURE 20. Influence spread under the IC model on Netphy data.

1) INFLUENCE SPREAD UNDER THE LT MODEL ON THE FOUR DATASET

Figure 23, Figure 24, Figure 25, and Figure 26 depict the number of influenced users on four studied datasets under the LT model. We can see clearly from the figures that the proposed algorithm performed better than the DDH algorithm for NetHept data while is less in terms of influence achieved than other algorithms. Similarly, our algorithm provided better results in terms of the influence spread on the NetPhy data than the DDH and BCT algorithms while it was lower than

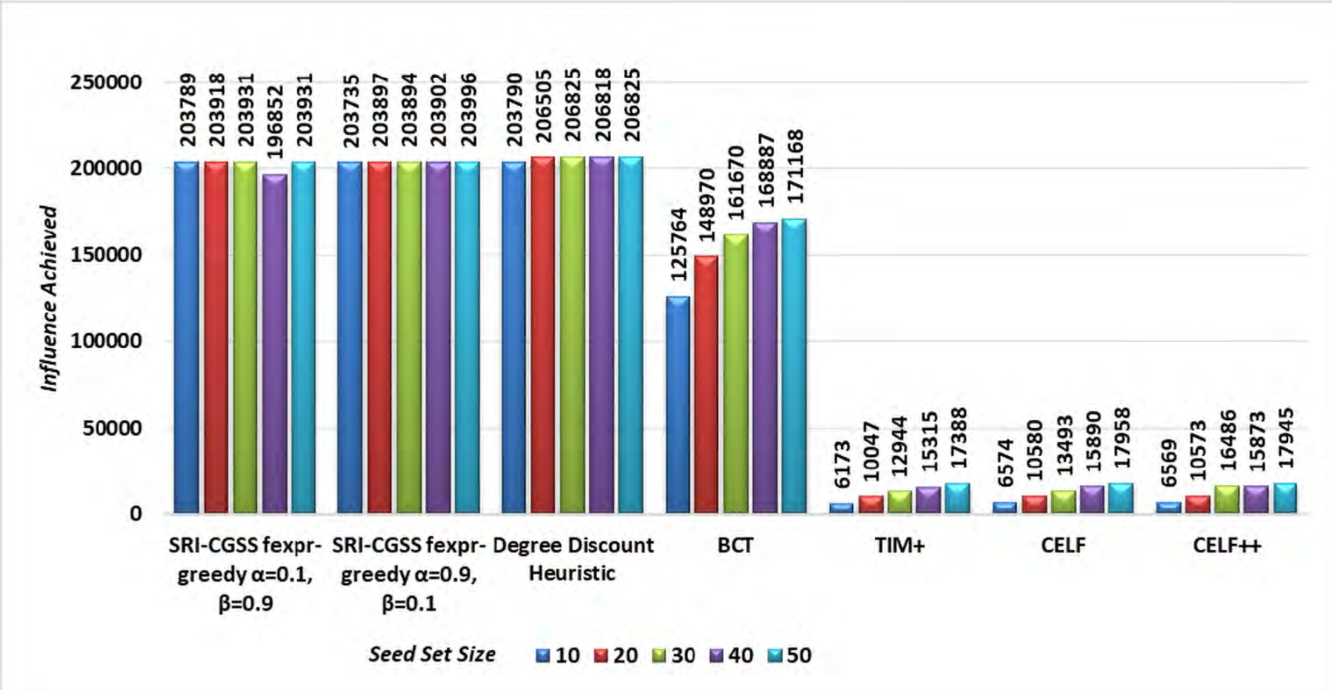

FIGURE 21. Influence spread under the IC model on Eu-Email data.

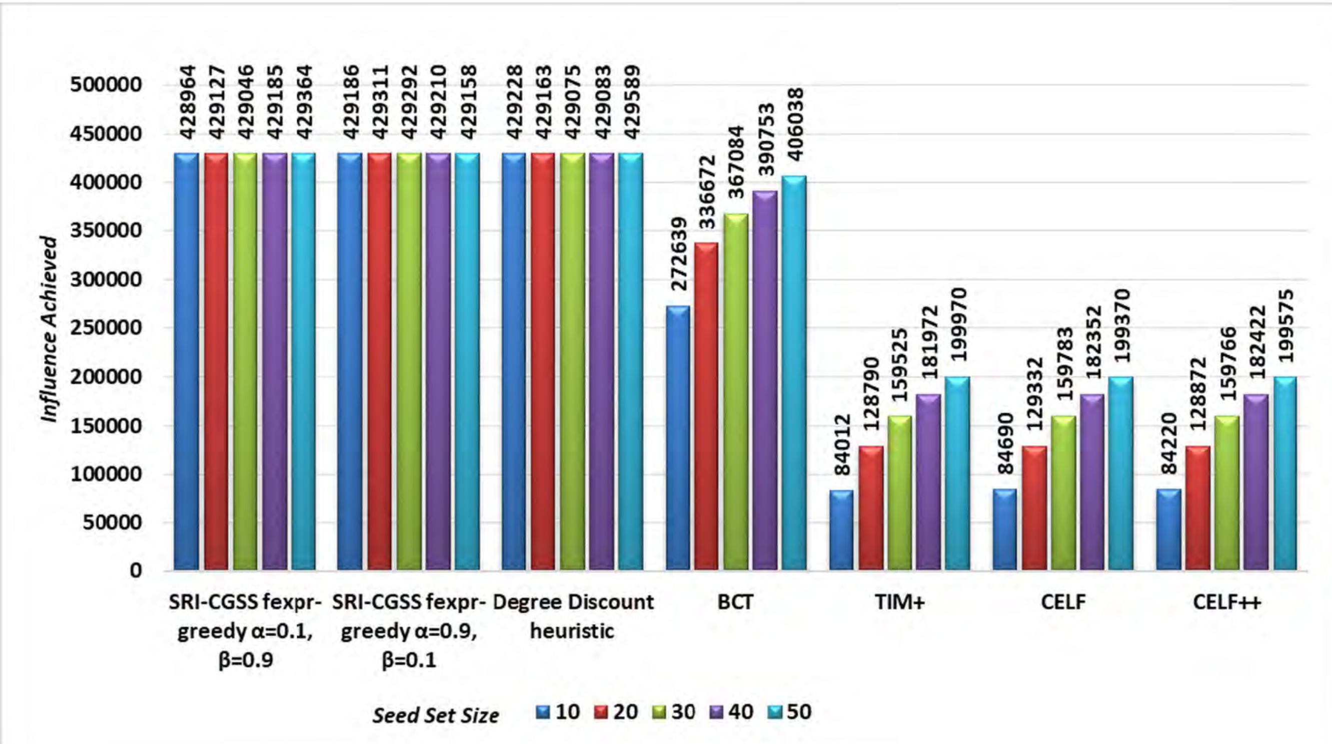

FIGURE 22. Influence spread under the IC model on Twitter data.

the rest of the algorithms. This can be enhanced by setting parameters that mostly fit for the used diffusion models which will be discussed later in this section.

Table 5 reports the time and space required to select the seed set and measure the influence under the LT model on the four studied datasets. It can be seen that the proposed

TABLE 4. Time and space complexity under the IC model for seed set size K = 50.

	Algorithms	*SRI-CGSS fexpr-greedy* α=0.1, β=*0.9*	*SRI-CGSS fexpr-greedy* α=0.9, β=*0.1*	*Degree Discount Heuristic*	*BCT*	*TIM+*	*CELF*	*CELF++*
NetHept data	*Time (S)*	417.93	480.64	4.21	0.01	5.02	649.80	808.80
	Space complexity(MB)	128	128	62.40	23.06	848.47	21.37	22.01
NetPhy data	*Time (S)*	1756.93	3823.11	51.96	0.67	14.18	7168.98	7176.60
	Space complexity(MB)	280	312	162.4	120.28	691.56	32.54	34.21
Eu-Email data	*Time (S)*	43542.88	42296.29	7584.11	0.40	13.70	27382	27659
	Space complexity(MB)	1192	1200	632	79.72	79.78	88.98	101.18
Twitter data	*Time (S)*	5882.86	8709.66	11470.08	0.03	2.63	1169802	1219404
	Space complexity(MB)	1280	1280	840	107.11	192	9826.95	188.07

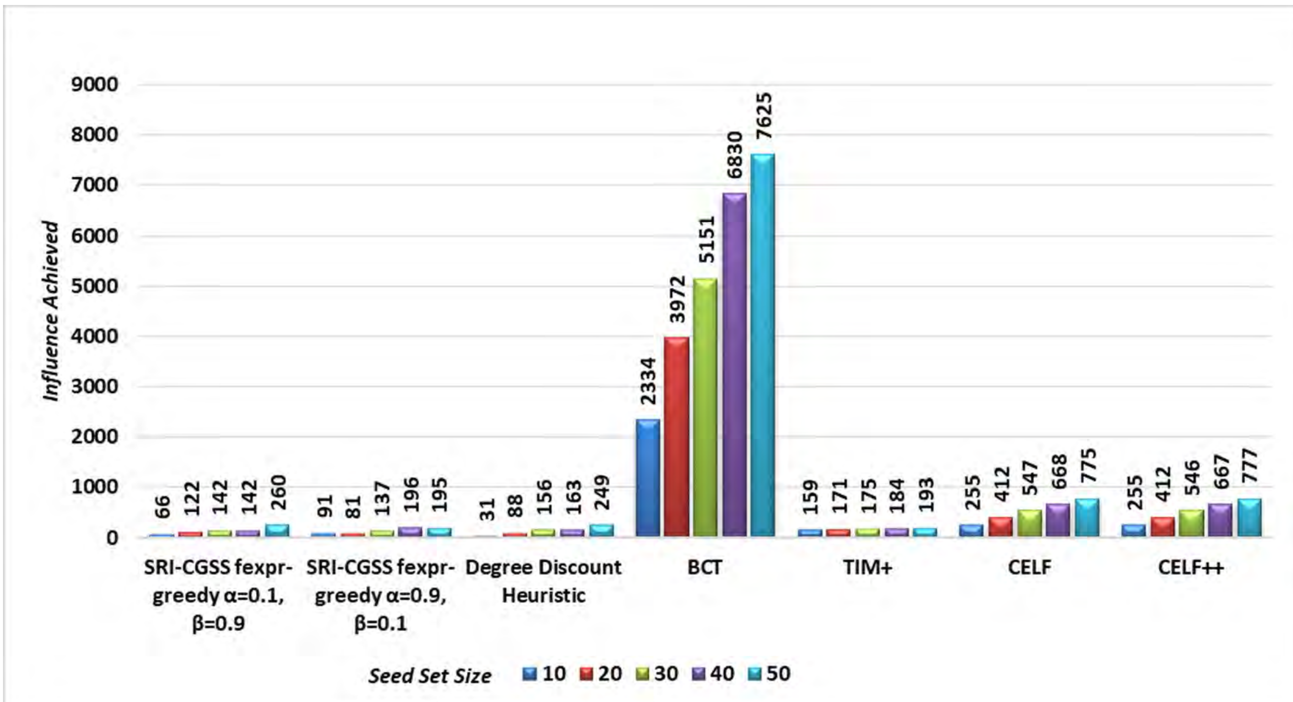

FIGURE 23. Influence spread under the LT model on Nethept data.

algorithm had lower runtime compared with CELF and CELF++ on all the datasets and required less storage than TIM+ on the NetPhy and NetHept datasets.

2) CUMULATIVE REGRET ON FOUR STUDIED DATASETS

We focus on the analysis of cumulative regret obtained over N rounds on the four studied datasets. We are

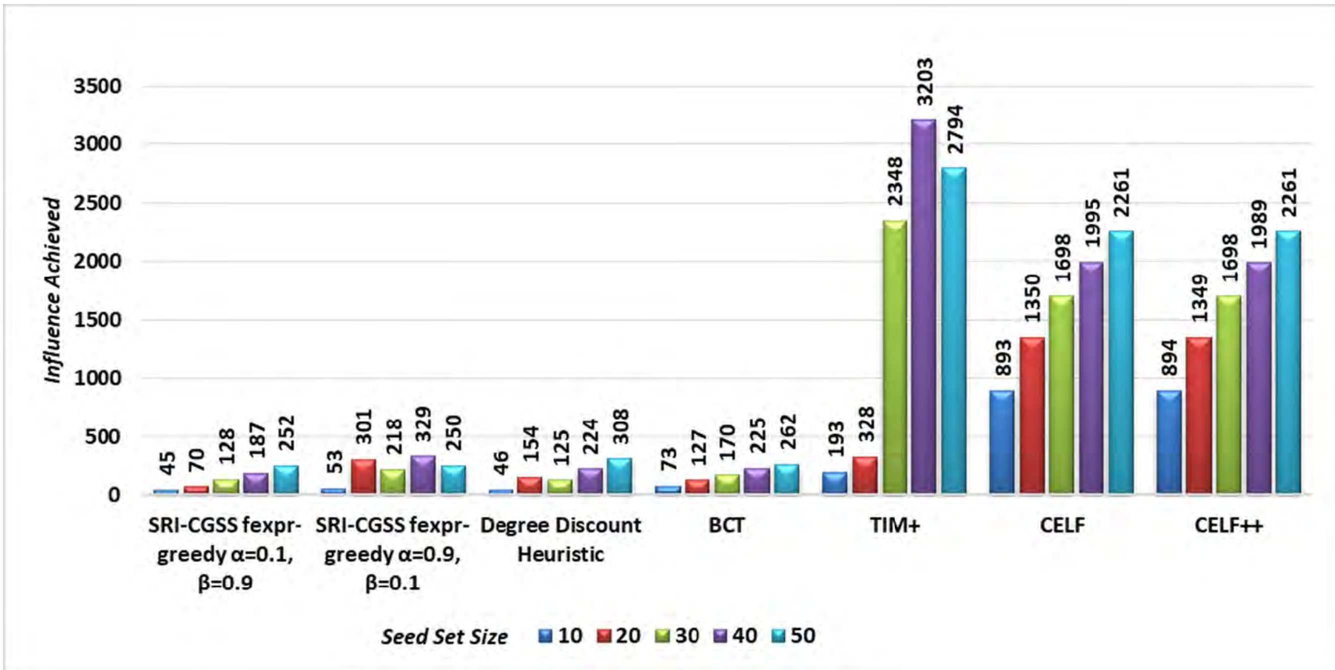

FIGURE 24. Influence spread under the LT model on Netphy data.

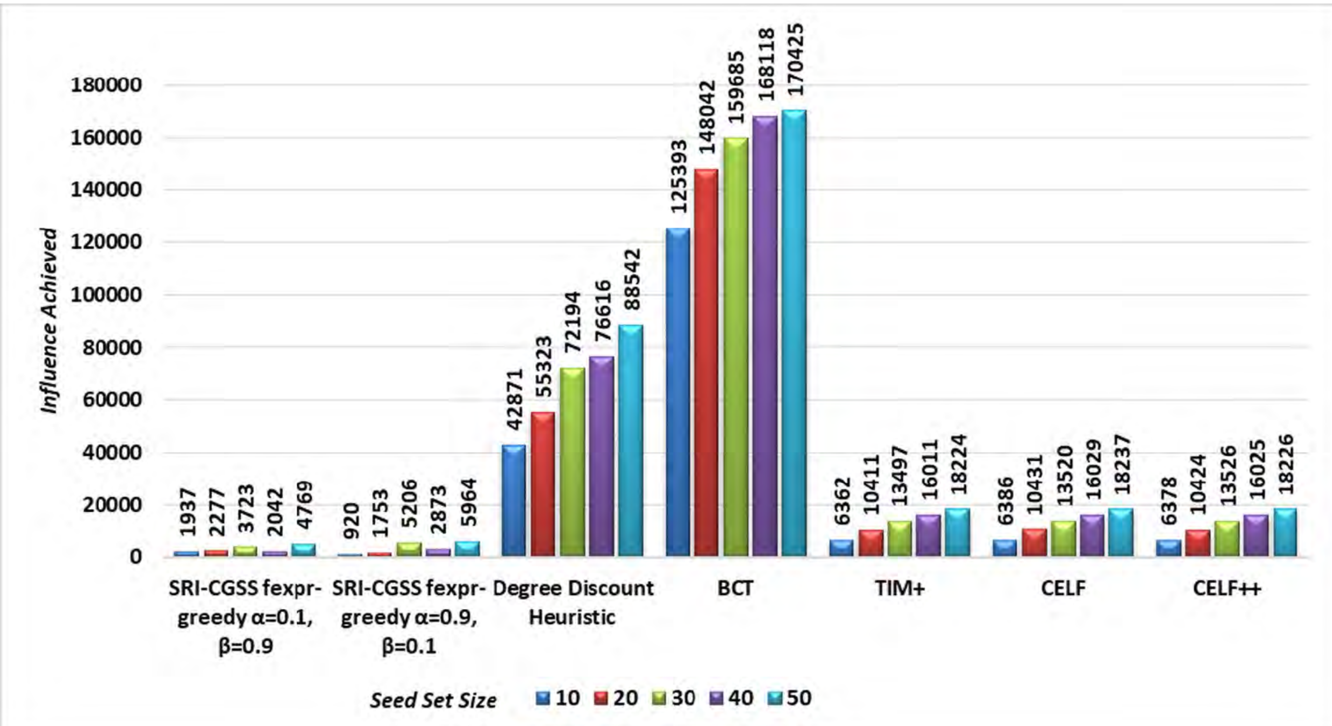

FIGURE 25. Influence spread under the LT model on Eu-Email data.

interested in showing how the reward parameters impact the changes in cumulative regret. This part of the experiment presents cumulative regret when the selection of arms and thus, the seed set, is mostly exploiting and rarely exploring.

Figure 27, Figure 28, Figure 29, and Figure 30 depict cumulative regret on the four studied datasets over N rounds. It can be seen that for the NetHept and Twitter datasets, the selection of relevant influencers should rely mainly on the similarity between adjacent users with a high rate

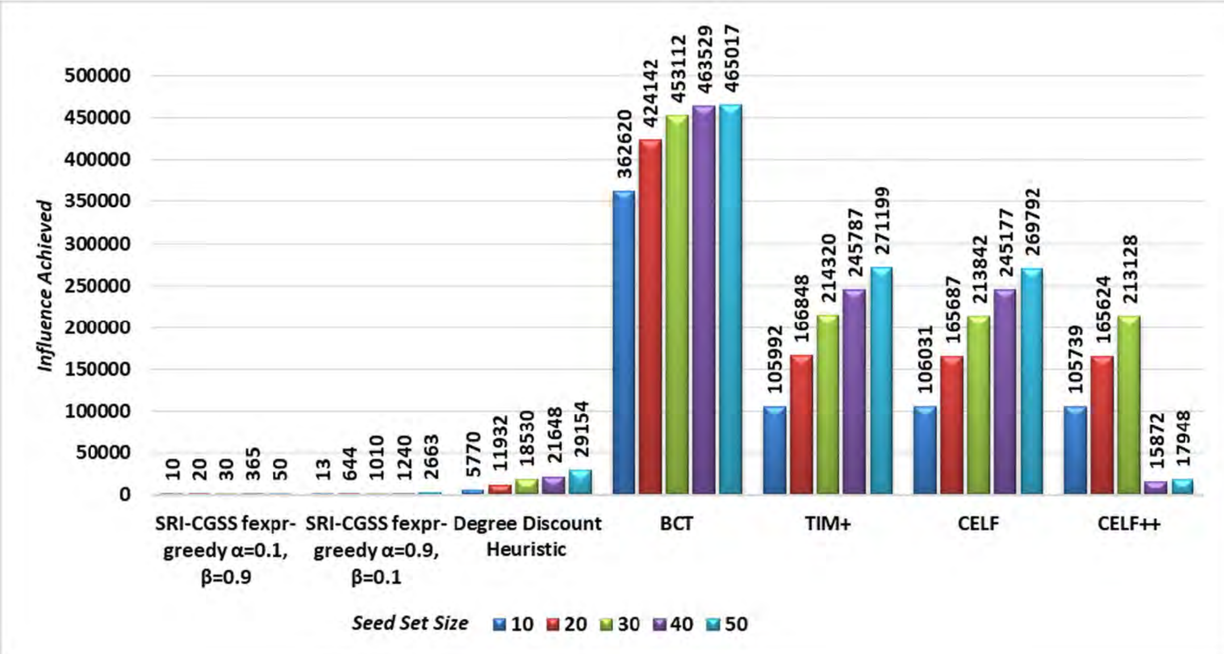

FIGURE 26. Influence spread under the LT model on Twitter data.

TABLE 5. Time and space complexity under the LT model for seed set size K = 50.

	Algorithms	*SRI-CGSS fexpr-greedy α=0.1, β=0.9*	*SRI-CGSS fexpr-greedy α=0.9, β=0.1*	*Degree Discount Heuristic*	*BCT*	*TIM+*	*CELF*	*CELF++*
NetHept data	*Time (S)*	490.34	583.83	*0.78*	*0.01*	*2.03*	*1111.80*	*1237.80*
	Space complexity(MB)	*136*	120	*72*	*23.06*	*730.66*	*21.40*	*22.12*
NetPhy data	*Time (S)*	1999.15	1687.3	4.67	0.60	2.08	7915.80	8733.60
	Space complexity(MB)	320	336	162.4	120.44	394.60	32.89	34.53
Eu-Email data	*Time (S)*	39855.55	38495.38	*11.19*	*0.02*	*5.15*	*36168*	*38497*
	Space complexity(MB)	1296	856	*632*	*79.78*	*159.89*	*90.11*	*101.94*
Twitter data	*Time (S)*	6965.98	9890.28	*54.41*	*0.034*	*2.06*	*1491786*	*30161*
	Space complexity(MB)	1512	1512	*54.41*	*107.10*	*191.46*	*101.29*	*120.23*

while for NetPhy and Eu-Email it seems that the algorithm performed well in promoting the product among other users in the case in which the selected seed set falls between many other edges. This changes the performance in that the lowest cumulative regret differs from one setting to another depending largely on the structure of the

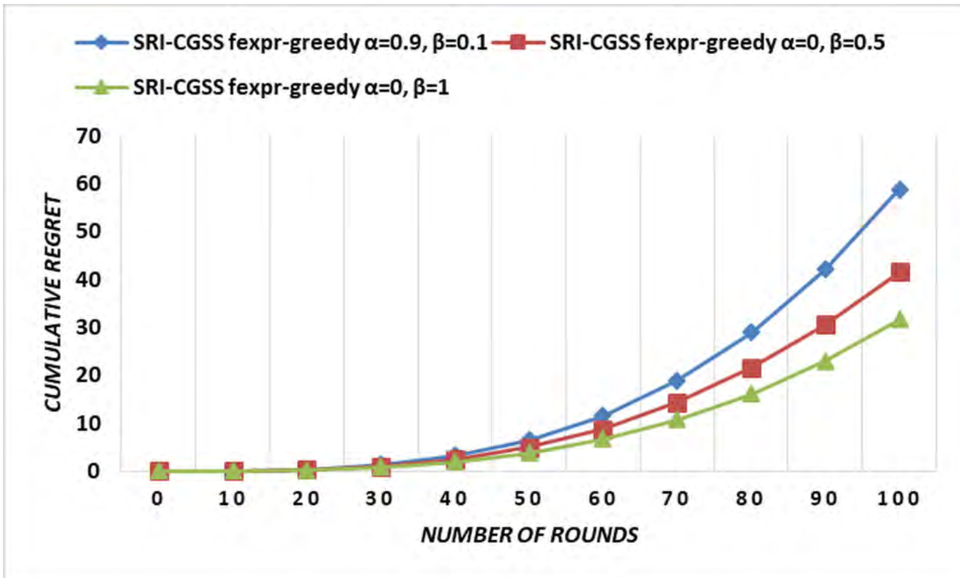

FIGURE 27. Cumulative regret on Nethept data versus the number of rounds.

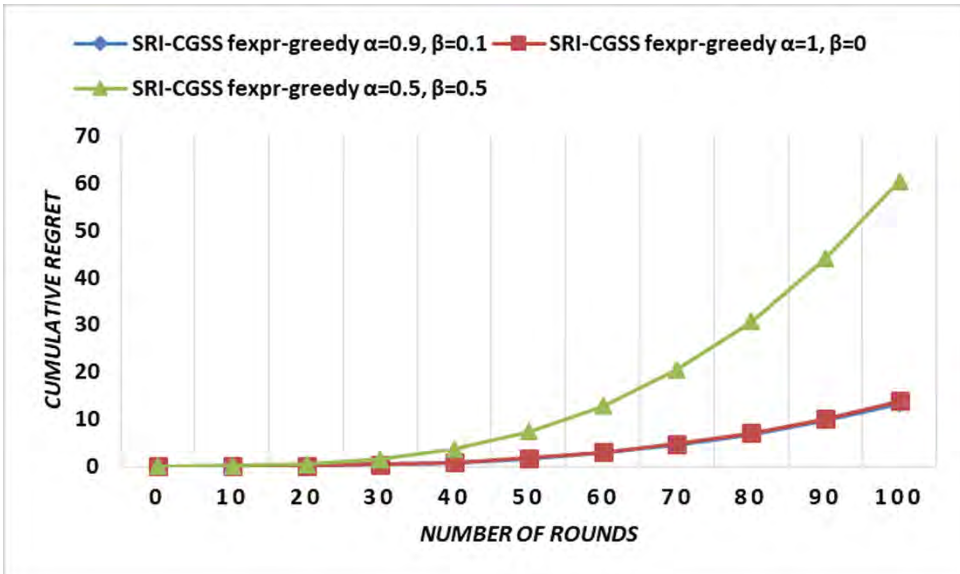

FIGURE 28. Cumulative regret on Nethphy data versus the number of rounds.

studied graph and how the individuals are connected to each other.

3) IMPACT OF REWARD FUNCTION ON INFLUENCE SPREAD

Figure 31, Figure 32, Figure 33, and Figure 34 present the influence achieved versus the seed set size that can be granted to select relevant influencers. We remark that the choice of relevant influencers is affected by the initialization of the multi-armed bandit reward that differs significantly in terms of the number of individuals who adopted the promoted product, and thus, we notice that for NetHept and Eu-Email, the algorithm performed well in terms of influencing many individuals when it rarely relies on the selection of central users while focusing on how individuals are similar to each other. However, for NetPhy data, it seems that the proposed approach achieved better results when most central edges are considered, and thus, seed sets are selected accordingly and that "SRI-CGSS fexpr-greedy" provides promising results when there is a balance between selecting most central edges and most edge members that are similar to each other by sharing the same interest. From this, we conclude that a preliminary study of the network structure of data is of great importance considering the used diffusion models. Another interesting remark is the low regret, and

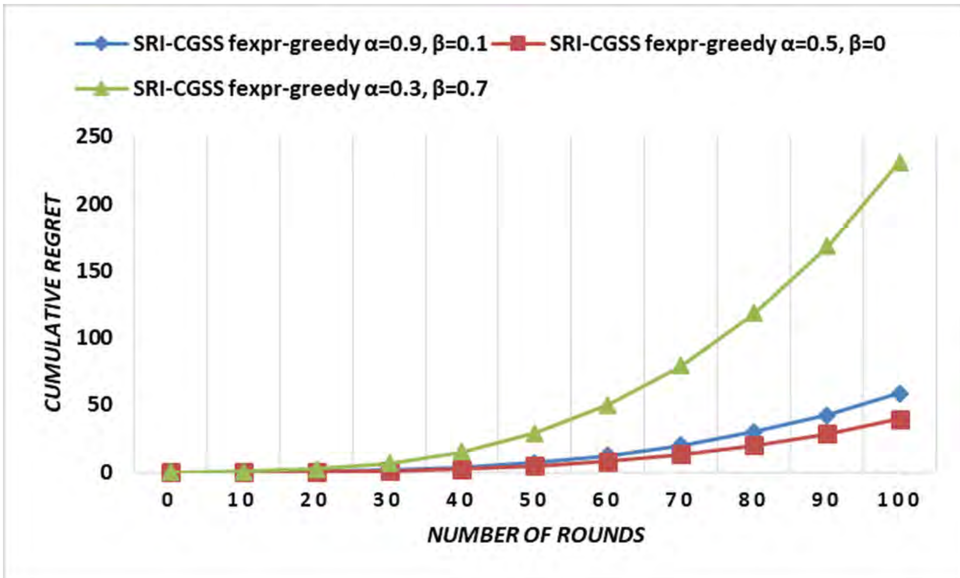

FIGURE 29. Cumulative regret on Eu-Email data versus the number of rounds.

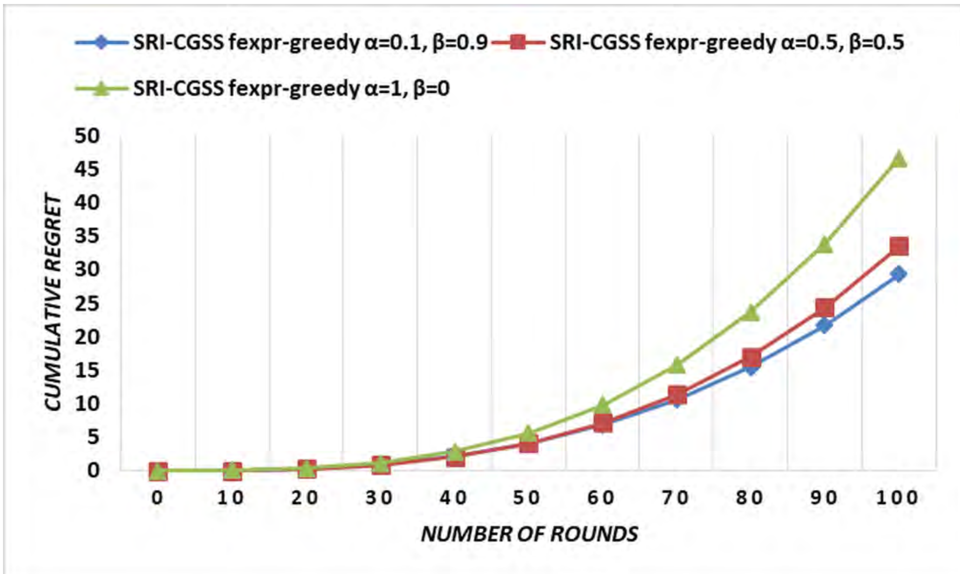

FIGURE 30. Cumulative regret on Twitter data versus the number of rounds.

the best influence can be seen on the NetHept and Net-Phy which confirm that the designed methodology is suited for most for undirected graphs by the 90% rate. However, it should be improved for undirected graphs that match by 40%.

C. SRI-MAB ϵ-GREEDY ALGORITHM

This part of the experiment concentrates on the selection of relevant influencers by using the epsilon-greedy algorithm and by initializing the reward function by some centrality measures that help to select a seed set that is relevant to spread the influence as much as possible. The experiment takes place over N rounds in online sequential decision-making where the selection made at each round is either in selecting the arms at the exploitation phase with immediate reward *rew* should be greater than $1 - \epsilon$ or at the exploration phase with *rew* less than $1 - \epsilon$.

As in the previous section, we are interested in evaluating the "SRI-MAB ϵ-greedy" algorithm regarding its performance in terms of the achieved influence, time complexity and storage needed compared with existing algorithms. Then, we chose two parameterizations for each ϵ to determine its impact on both influence coverage and cumulative regret.

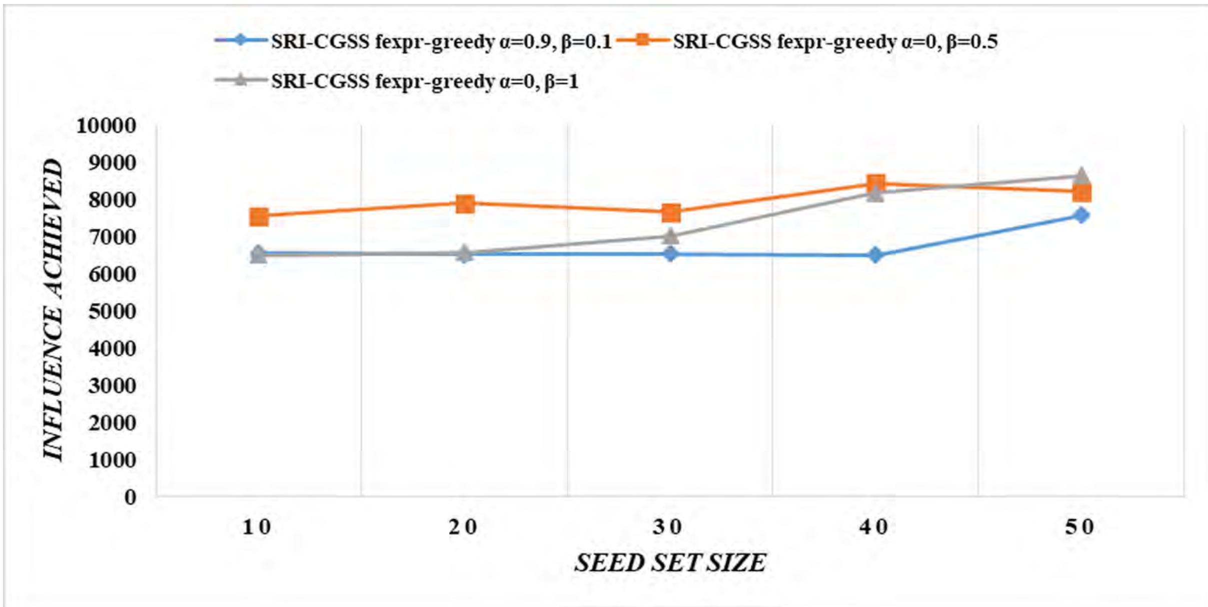

FIGURE 31. Impact of varying α and β on influence spread on Nethept data.

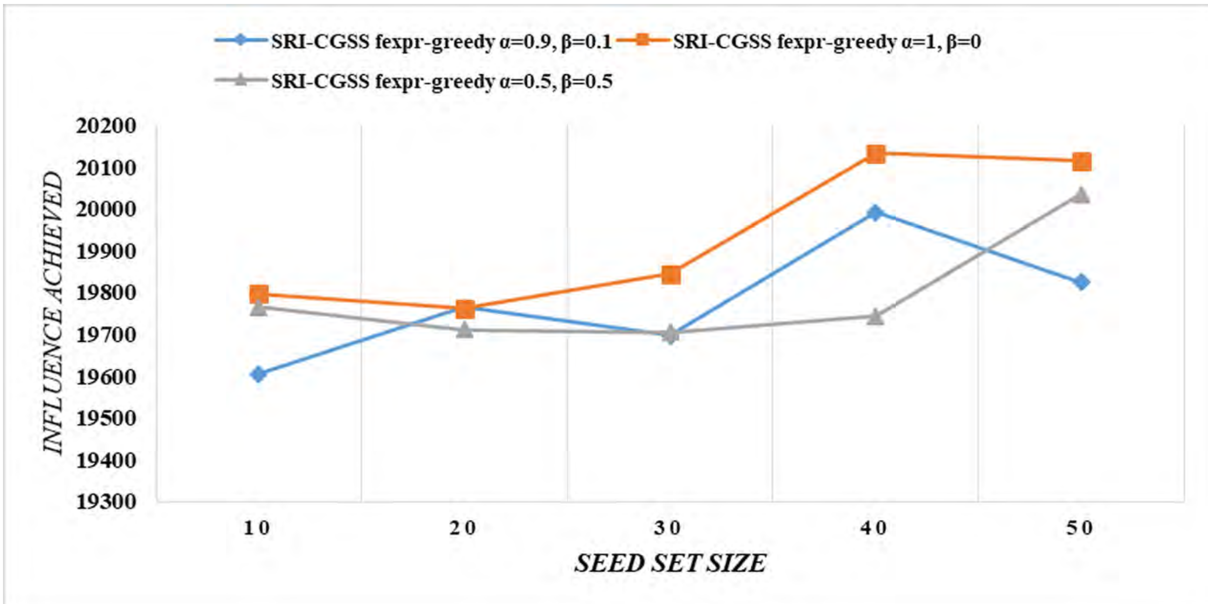

FIGURE 32. Impact of varying α and β on influence spread on Netphy data.

Figure 35, Figure 36, Figure 37, and Figure 38 represent the number of influenced users under the IC model on the four datasets. From the four figures above, we deduce that the "SRI-MAB ϵ -greedy" algorithm performs better by 0.125% than the best existing algorithm "DDH" on NetHept and is better by 0.001% on the NetPhy data and by 0.002% on the Twitter data. However, the proposed algorithm provides worse performance in terms of overall influence spread for Eu-Email compared with the best algorithm, DDH, by 0.0027%. Therefore, the proposed algorithm performs very well compared with existing approaches on three datasets and is slightly lower only for Eu-Email which could be improved significantly by adjusting initial reward function and thus fitting the ϵ and finding an optimal balance between exploration and exploitation.

Table 6, shows the time and space needed to select the seed set and measure the influence spread under the IC model for the worst case, which means that we have to choose the $K = 50$ most relevant influencers. We clearly notice that as with the previous algorithms, the introduced algorithm required less time than CELF and CELF++ on all data and also required less time than DDH on the Twitter data.

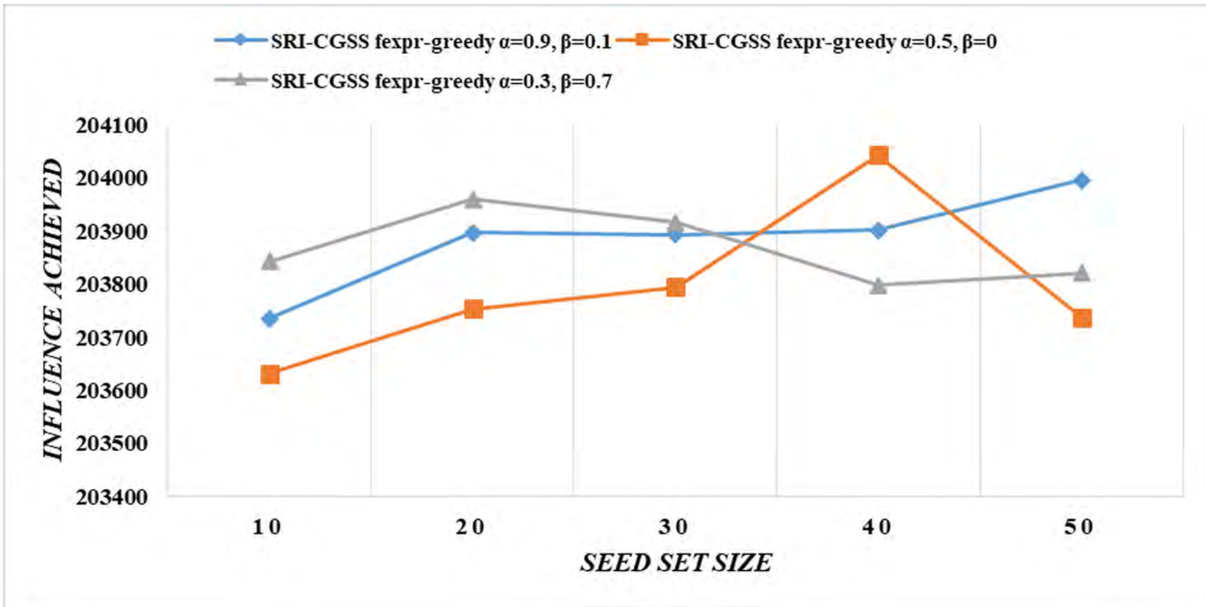

FIGURE 33. Impact of varying α and β on influence spread on Eu-Email data.

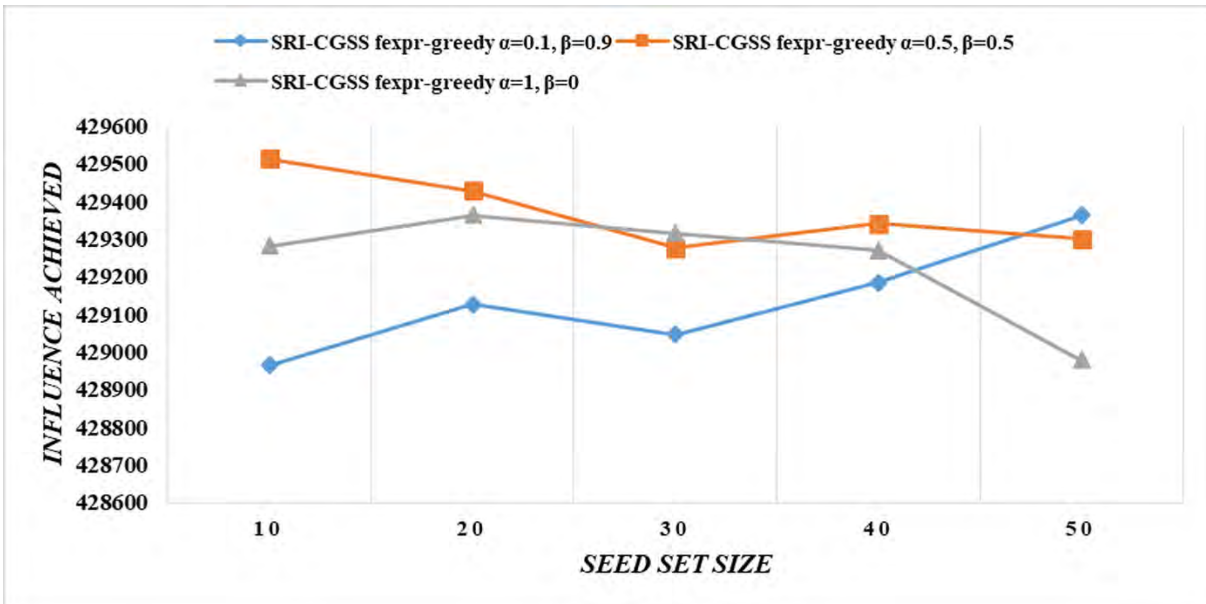

FIGURE 34. Impact of varying α and β on influence spread on Twitter data.

Figure 39, Figure 40, Figure 41, and Figure 42 report the influence spread on the four studied datasets under the LT model. We notice that, as previously, our algorithm did not provide the best influence under the LT model compared with other algorithms including CELF++, TIM+, BCT, and CELF. Additionally, even the existing algorithms did not perform well on all data, since we notice that BCT is the best algorithm on NetHept, Twitter and Eu-Email data, TIM+ and CELF++ is more performant on the NetPhy data.

Table 7 shows the time and space required on four datasets under the LT model for selecting K = 50 relevant influencers. We observe that the proposed algorithm required less time than CELF and CELF++ on all the datasets and required less space on undirected graphs namely, NetHept and NetPhy data compared with TIM+. This indicates that the proposed algorithms compute global structural properties on all the graphs, which require more time and storage space to store the data. Additionally, the algorithm ran over 100 rounds, which increased the time for identification of the seed set.

TABLE 6. Time and space complexity under the IC model for seed set size K = 50.

	Algorithms	SRI-MAB ϵ=0.1α=0.1, β=0.9	SRI-MAB ϵ=0.1,α=0.9, β=0.1	SRI-MAB ϵ=0.05, α=0.1, β=0.9	SRI-MAB ϵ=0.05, α=0.9, β=0.1	Degree Discount Heuristic	BCT	TIM+	CELF	CELF++
NetHept data	Time (S)	473.70	502.97	395.27	385.53	4.21	0.01	5.02	649.80	808.80
	Space complexity(MB)	133.60	135.20	133.60	136.80	62.40	23.06	848.47	21.37	22.01
NetPhy data	Time (S)	4666.20	5002.75	4001	3833.75	51.96	0.67	14.18	7168.98	7176.60
	Space complexity(MB)	288	312	313.60	313.60	162.40	120.28	691.56	32.54	34.21
Eu-Email data	Time (S)	18085.86	17534.40	22016.25	20471.50	7584.11	0.40	13.70	27382	27659
	Space complexity(MB)	1200	1200	1192	1200	632	79.72	79.78	88.98	101.18
Twitter data	Time (S)	7890.53	7562.51	5467.87	8627.65	11470.08	0.03	2.63	1169802	1219404
	Space complexity(MB)	1280	1288	1280	1288	840	107.11	192	9826.95	188.07

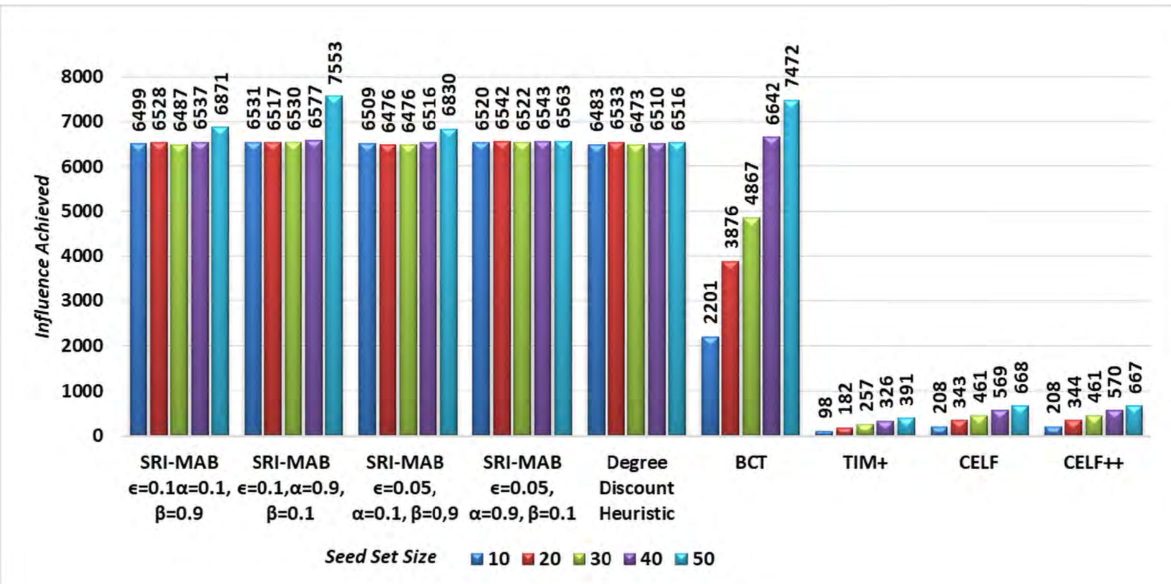

FIGURE 35. Influence spread under the IC model on Nethept data.

1) CUMULATIVE REGRET ON FOUR STUDIED DATASET

Figure 43, Figure 44, Figure 45, and Figure 46 show cumulative regret on the four datasets over N rounds for the "SRI-MAB ϵ-greedy" algorithm. We note that the algorithm has the lowest regret when $\epsilon = 0.1$ and that the selection of the seed set affects a large number of individuals

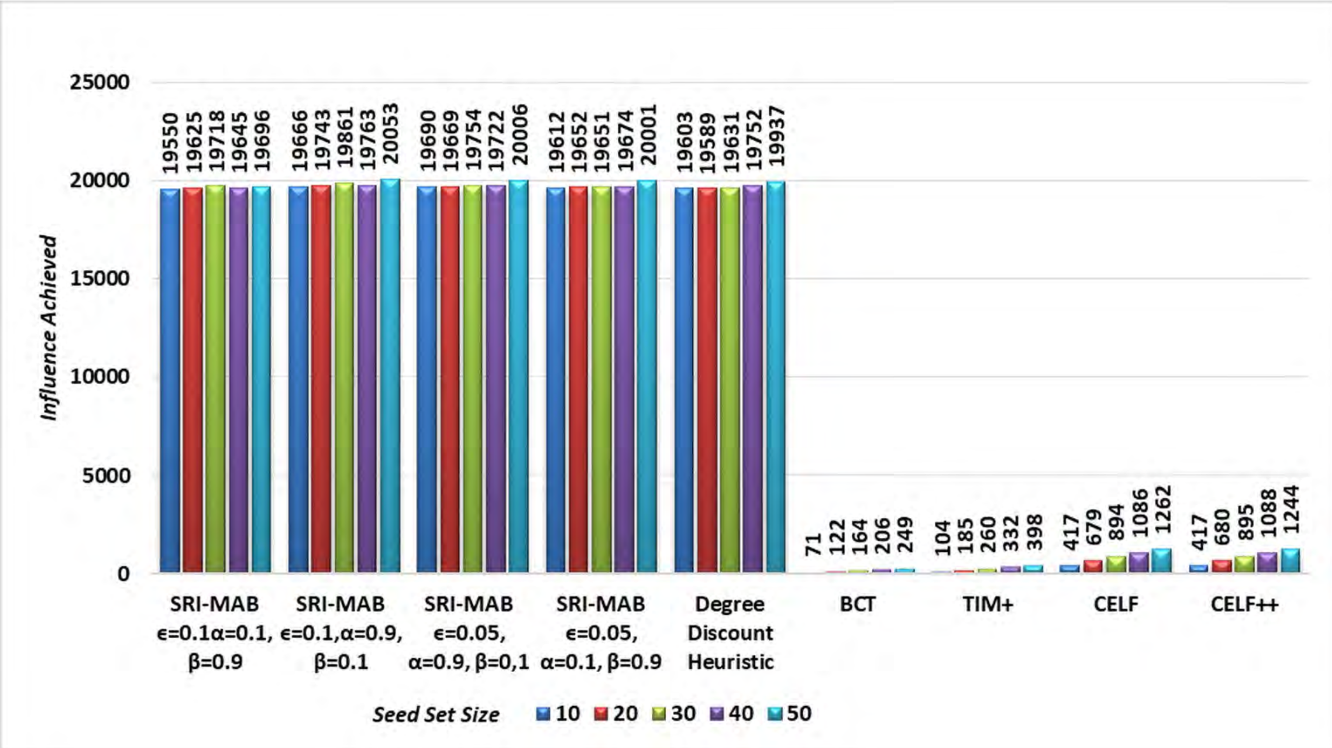

FIGURE 36. Influence spread under the IC model on Netphy data.

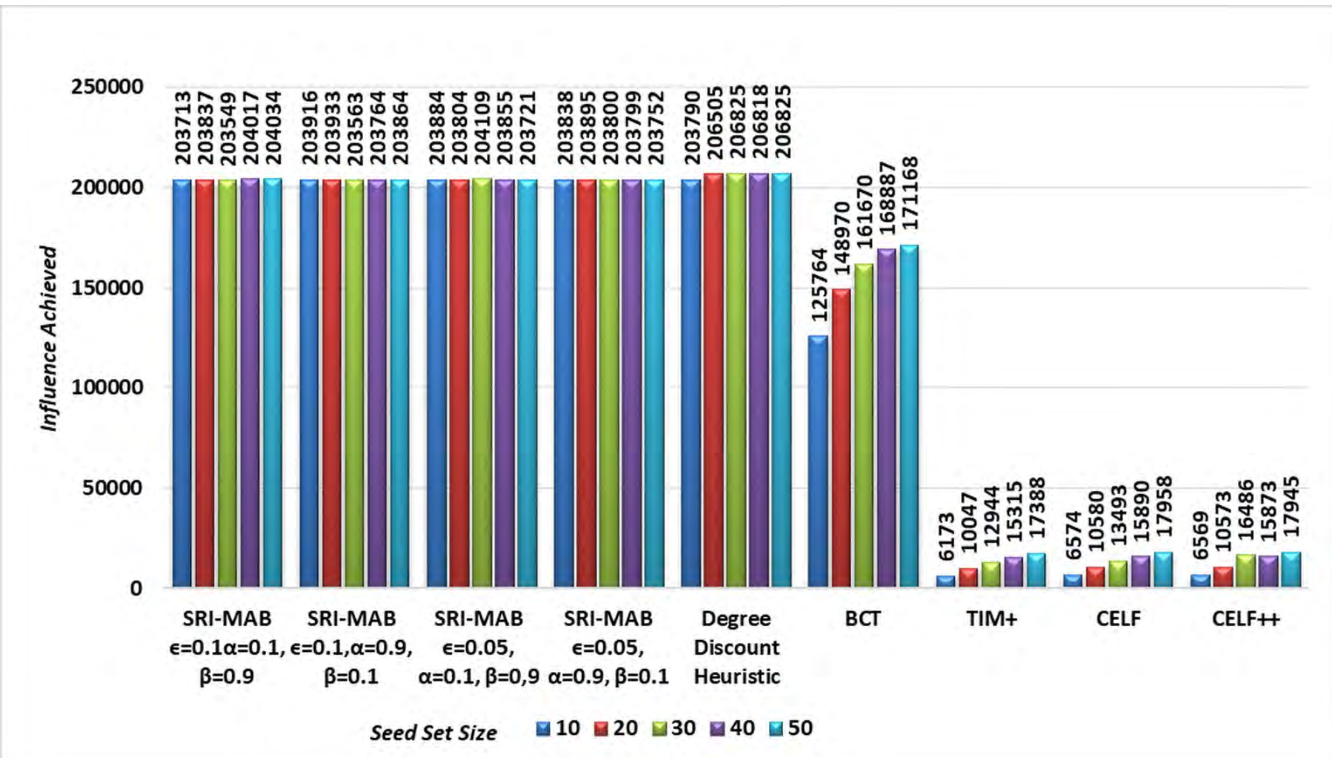

FIGURE 37. Influence spread under the IC model on Eu-Email data.

mostly when those selected seed sets are similar to some extent to each other on the Twitter, Eu-Email and the NetPhy data, and the lowest cumulative regret occurred when the seed set was chosen according to its position within the network among other individuals on the NetHept data.

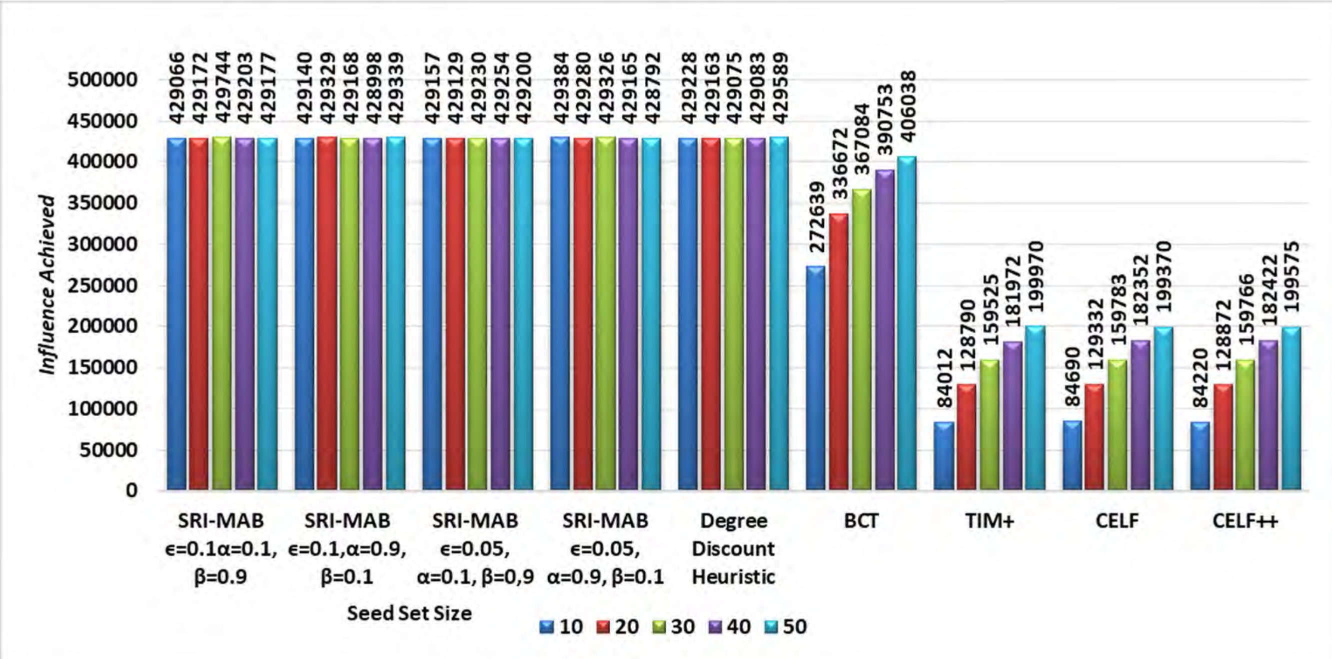

FIGURE 38. Influence spread under the IC model on Twitter data.

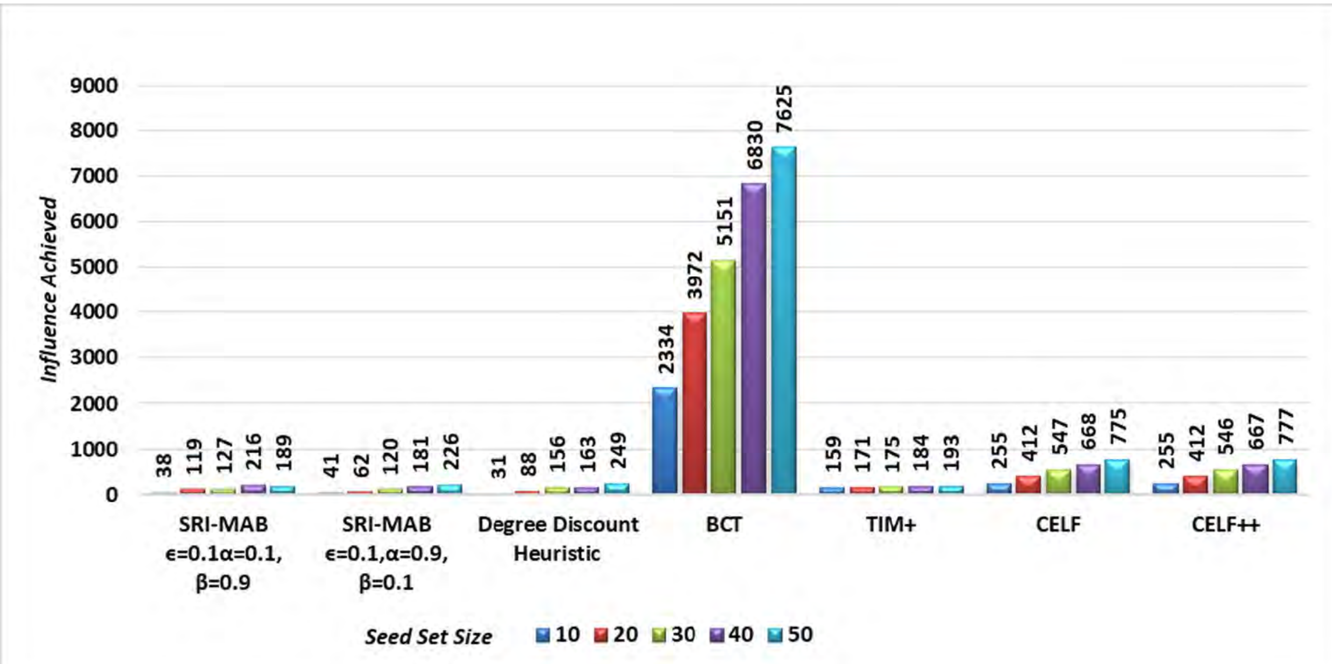

FIGURE 39. Influence spread under the LT model on Nethept data.

2) IMPACT OF REWARD FUNCTION ON INFLUENCE SPREAD

Figure 47, Figure 48, Figure 49, and Figure 50 display the influence spread achieved versus seed set size when we apply the "SRI-MAB ϵ-greedy" algorithm to select relevant influencers over N rounds. We notice that under the IC model, the selected seed set through the "SRI-MAB ϵ-greedy algorithm achieved higher overall influence on all the datasets when $\epsilon = 0.05$ and that it is more efficient, especially on the NetPhy and Twitter data to balance the choice of $\alpha = 0.5$ *and* $\beta = 0.5$ values between selecting edges in a central position and edge members that are similar to each other. In addition, we note that the selection of the seed

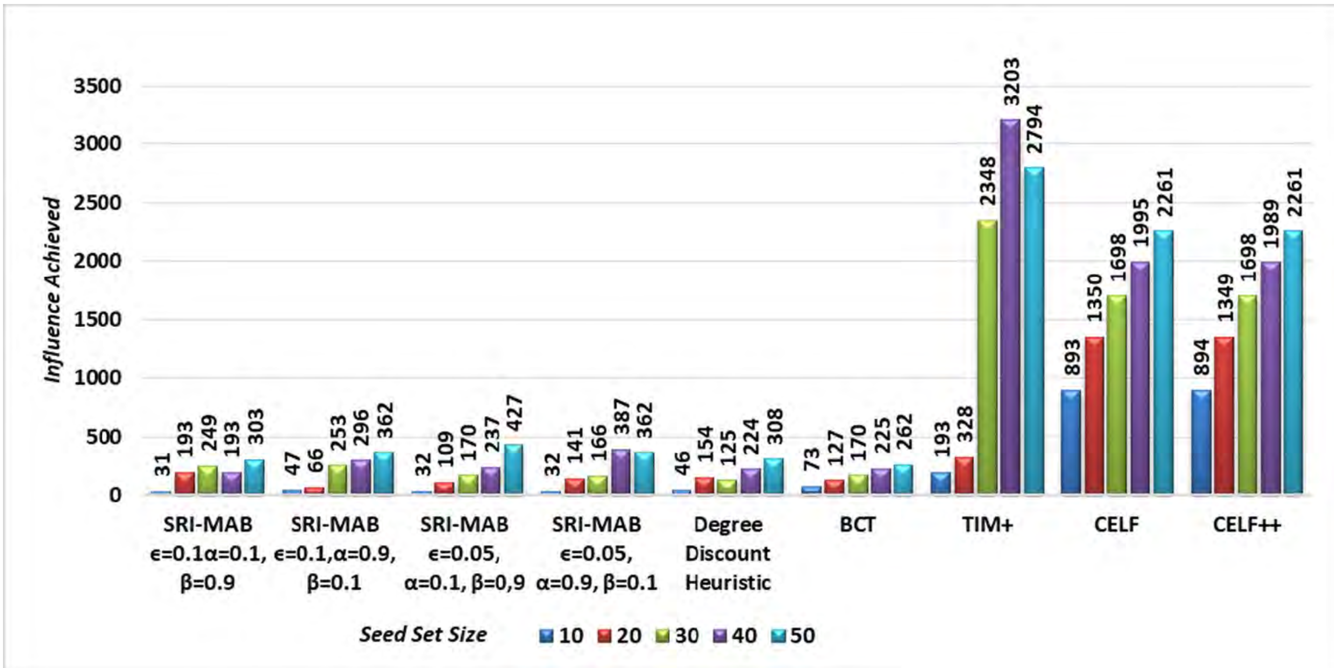

FIGURE 40. Influence spread under the LT model on Netphy data.

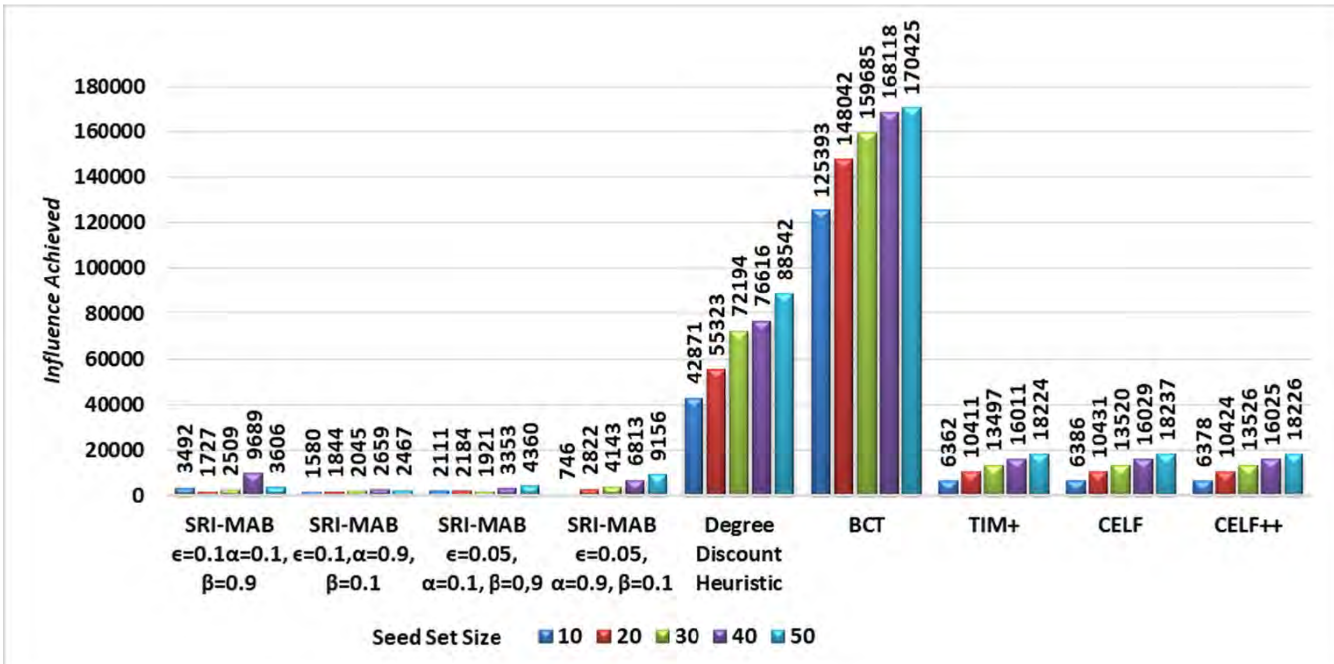

FIGURE 41. Influence spread under the LT model on Eu-Email data.

set is more efficient on the NetHept data with the selected arms, and thus, the seed set are selected by accounting for the edge members' similarity without considering the edge positions among the other edges in the network. However, the Eu-Email edge positions seem important for selecting relevant influencers that maximize the number of infected individuals.

VII. DISCUSSION

This work represents a step toward designing an effective method for selecting the most relevant influencers by adopting the greedy family of multi-armed bandit algorithms and is an attempt to find a tradeoff between exploitation and exploration with a selection of seed set. First, we focus on presenting a seed set selection algorithm that fully exploits and

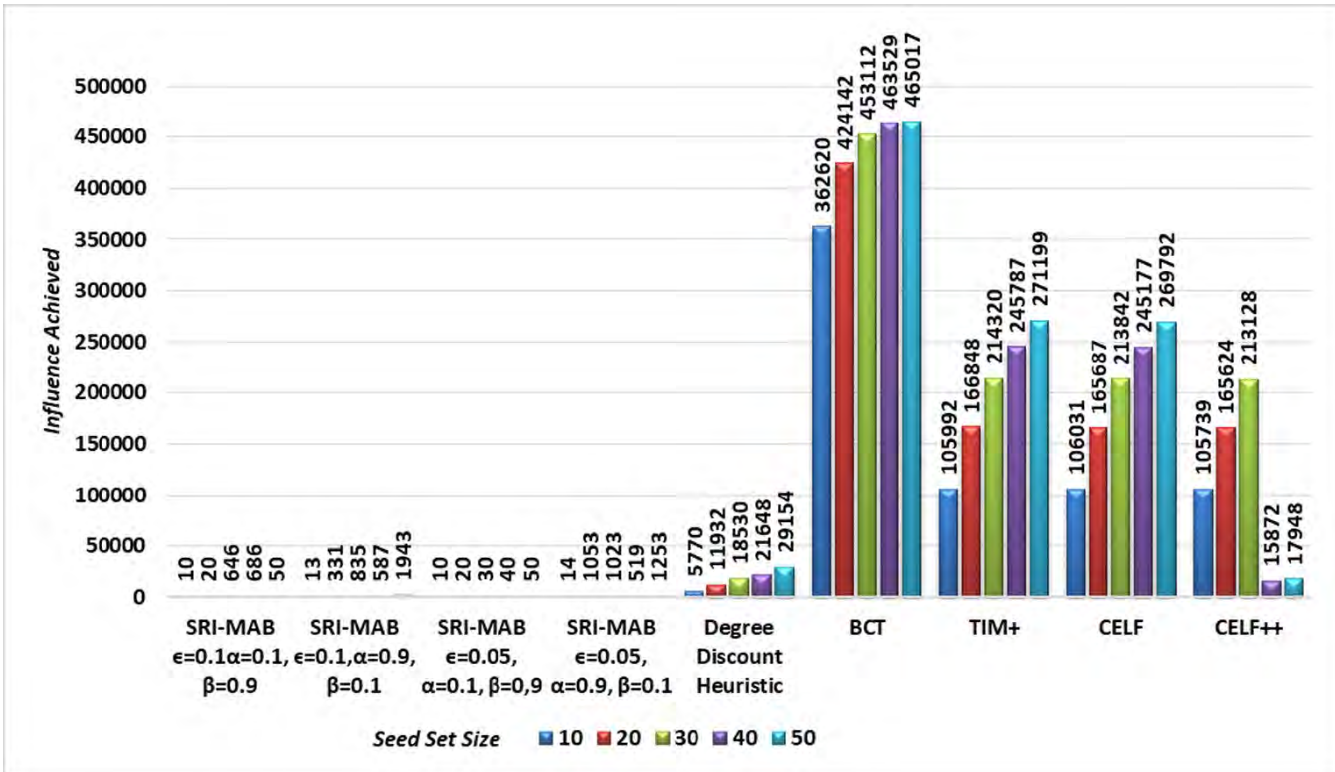

FIGURE 42. Influence spread under the LT model on Twitter data.

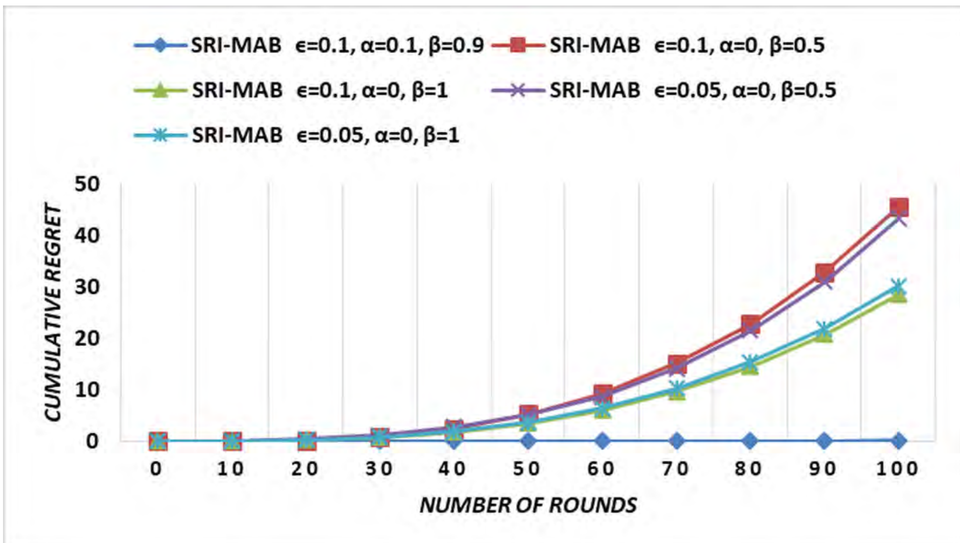

FIGURE 43. Cumulative regret on Nethept data versus the number of rounds.

explores rarely with a selection of relevant influencers based on position and similarity. Second, we concentrate on fully exploring and rarely exploiting with seed set selection. Then, a ϵ SRI-based MAB algorithm that balances exploration and exploitation is introduced. The main challenge is to find the exact value of ϵ that provides the best choice of seed set that maximizes the influence spread. Mainly, we performed extensive experimental results on the three algorithm versions and concluded that the introduced algorithms are better in terms of touching a large number of individuals when the algorithms rely on reward initialization mostly on edge locations among other edges in the network and rarely use the similarity of edge members. The proposed algorithms outperform all existing approaches on three datasets including

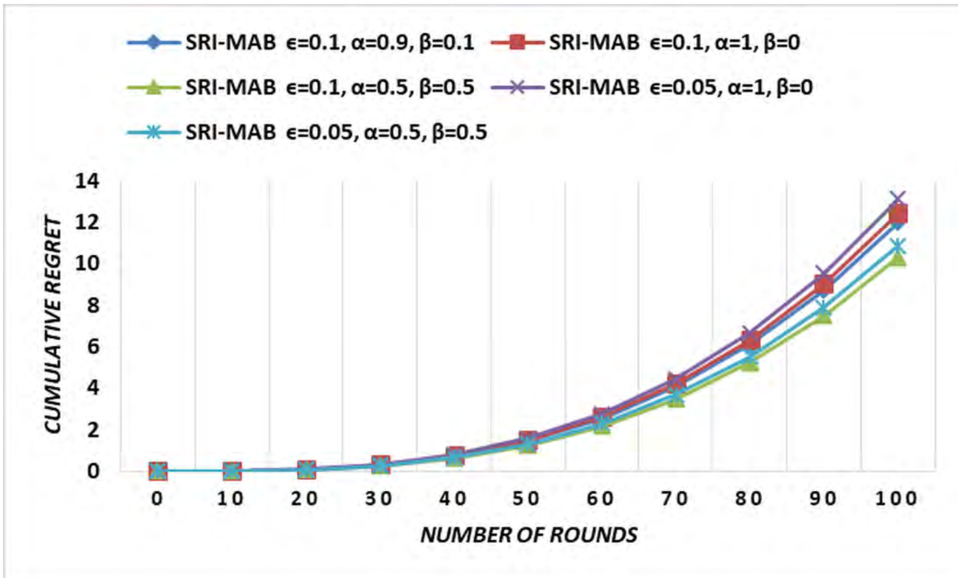

FIGURE 44. Cumulative regret on netphy data versus the number of rounds.

TABLE 7. Time and space complexity under the LT model for seed set size K = 50.

	Algorithms	*SRI-MAB ϵ=0.1α=0.1, β=0.9*	*SRI-MAB ϵ=0.1,α=0.9, β=0.1*	*SRI-MAB ϵ=0.05, α=0.1, β=0.9*	*SRI-MAB ϵ=0.05, α=0.9, β=0.1*	*Degree Discount Heuristic*	*BCT*	*TIM +*	*CELF*	*CELF++*
NetHept data	*Time (S)*	761.85	631.68	740.23	702.56	*0.78*	*0.01*	*2.03*	*1111.80*	*1237.80*
	Space complexity(MB)	116.80	101.60	*124.30*	*124.30*	*72*	*23.06*	*730.66*	*21.4*	*22.12*
NetPhy data	*Time (S)*	2660.16	5262.45	3296.27	5333.18	4.67	0,603	2.08	7915.8	8733.60
	Space complexity(MB)	320	320	325.60	325.60	162.40	120.44	394,60	32.89	34.53
Eu-Email data	*Time (S)*	18254.50	22506.50	19235.2	21282.50	*11.19*	*0,02*	*5.15*	*36168*	*38497*
	Space complexity(MB)	1336	1296	1296	840	*632*	*79.78*	*159.89*	*90.11*	*101.94*
Twitter data	*Time (S)*	8123.79	7191.53	9061.23	9265.40	*54.41*	*0,03*	*2.06*	*1491786*	*30161*
	Space complexity(MB)	1512	1512	1512	1512	*54.41*	*107.10*	*191,46*	*101.29*	*120.23*

NetHept, NetPhy, and Twitter data except for the Eu-Email data that has less influence compared to DDH. Our algorithms provide better results by adjusting values of ϵ, α, *and* β, so a deep study of how these parameter values should be considered for improving the selection of the best arms and thus, seed set. In addition, sometimes random exploration

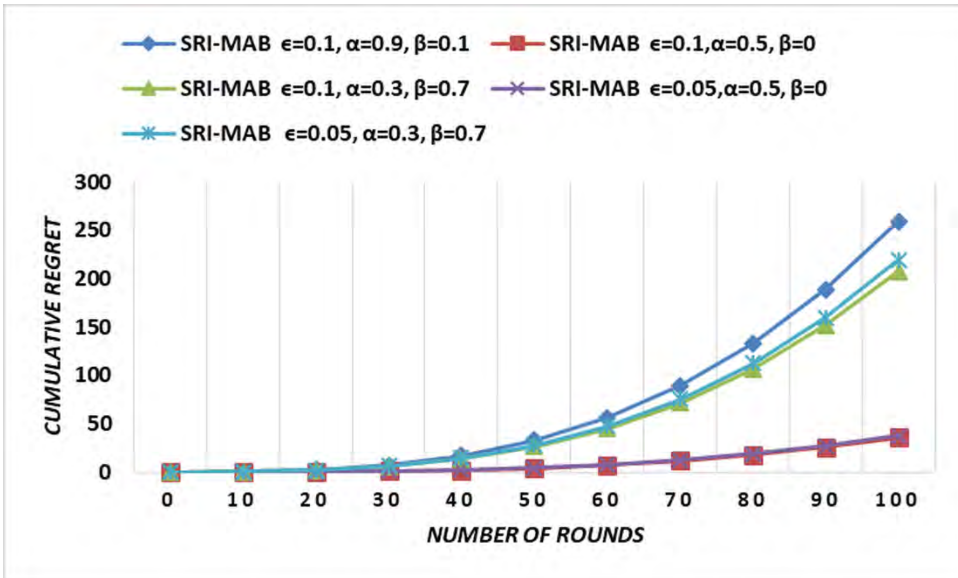

FIGURE 45. Cumulative regret on Eu-Email data versus the number of rounds.

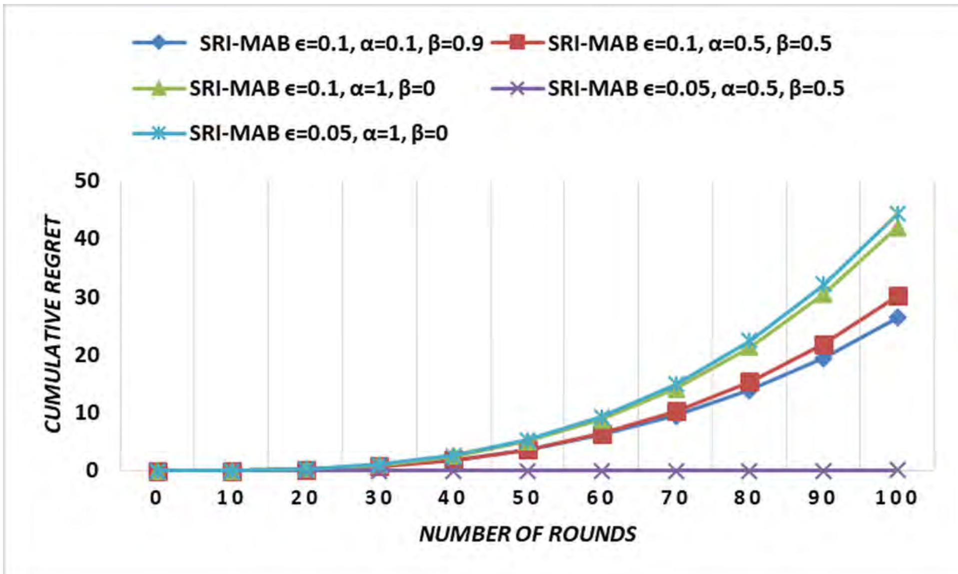

FIGURE 46. Cumulative regret on Twitter data versus the number of rounds.

with some deterministic selection of seed sets leads to the choice of relevant influencers and sometimes performs better than the ϵ exploitation-exploration strategy, that when failing the choice of the correct ϵ value that fits the studied dataset may give lower results in terms of the spread of influence.

Our proposed algorithms consume a little bit more time than some existing algorithms such as BCT, TIM+, and DDH on some datasets. This can be justified that the proposed algorithms need to run over 100 rounds to complete the selection as well as it relies on some global centrality measure that we improved largely its running time and that some of the existing approaches have some marginal spread that could not be improved under their proposal. However, our proposed algorithms can improve the spreading efficiency largely by adjusting the exploitation and exploration strategies and also by adjusting the tuning parameters of the corresponding reward function. Besides, the proposed algorithms run on a reasonable time on a large scale graph and exceed in most the spread efficiency under the IC model of all algorithms we compared with including CELF and CELF++ which are an improvement in time complexity of the original greedy algorithms. In addition, the introduced methodology may fit a large range of datasets which may not be the case for other

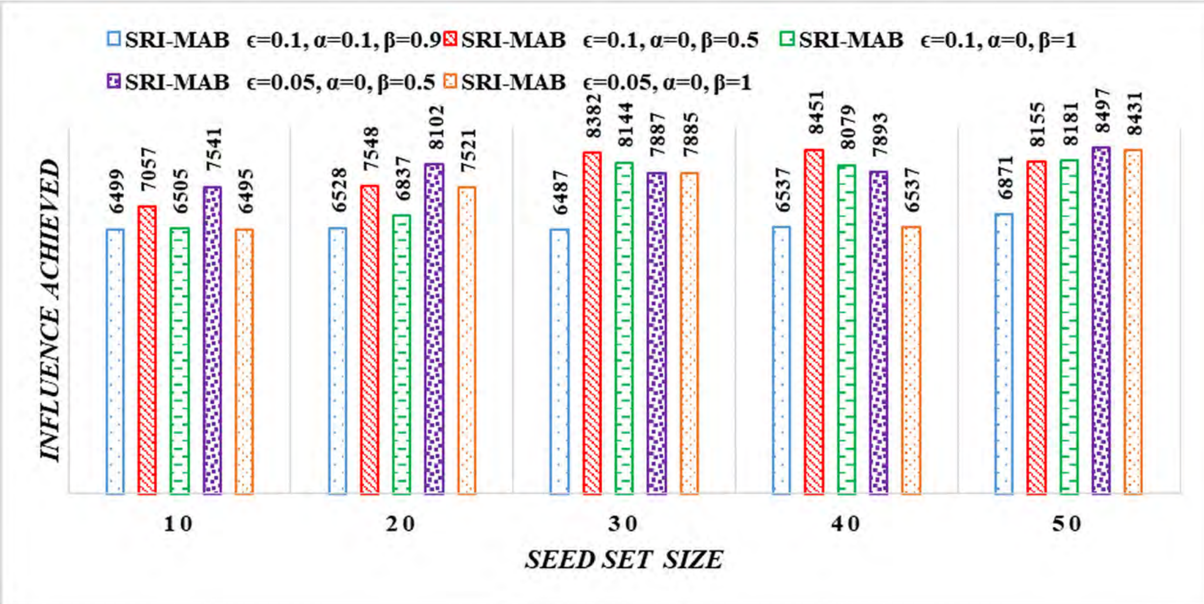

FIGURE 47. Impact of varying α and β on influence spread on Nethept data.

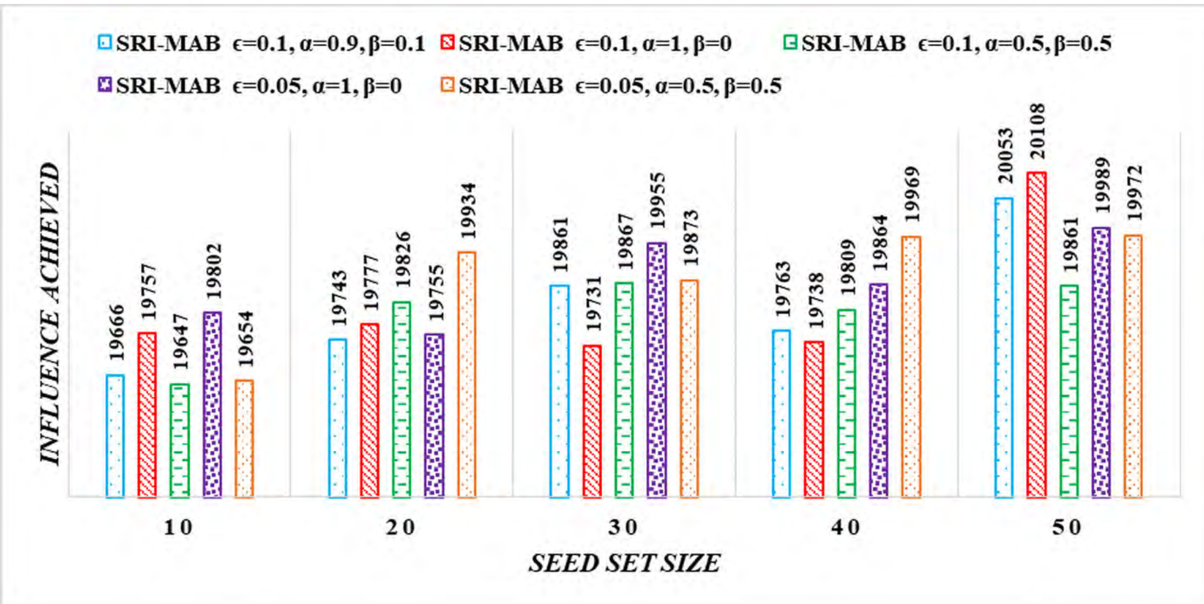

FIGURE 48. Impact of varying α and β on influence spread on Netphy data.

algorithms and is a step forward to explore further how may the reward function impact the results of spread on a MAB framework, and that the time spent to search for best seed set may be optimized later by parallelizing the computation of reward function.

As the next step, we moved on to evaluate the cumulative regret obtained by three proposed algorithms on all studied datasets and we deduced that the cumulative regret is the lowest on NetHept and Twitter data when the initialization of the multi-armed bandit reward algorithms relied more on the similarity of the selected edge members and ignored the central position of the edges. In addition, $\epsilon = 0.1$ in the third algorithm seems to provide good results besides $\epsilon = 0.05$ with the choice of adequate values of reward parameters. Whereas, for NetPhy and Eu-Email data, the cumulative regret is less when the reward of the multi-armed bandit algorithms is based on selected edge members that are more similar to each other. Thus, we note that "SRI-CGSS fexpr-greedy" has the lowest regret compared with other proposed algorithms. In the last experiments,

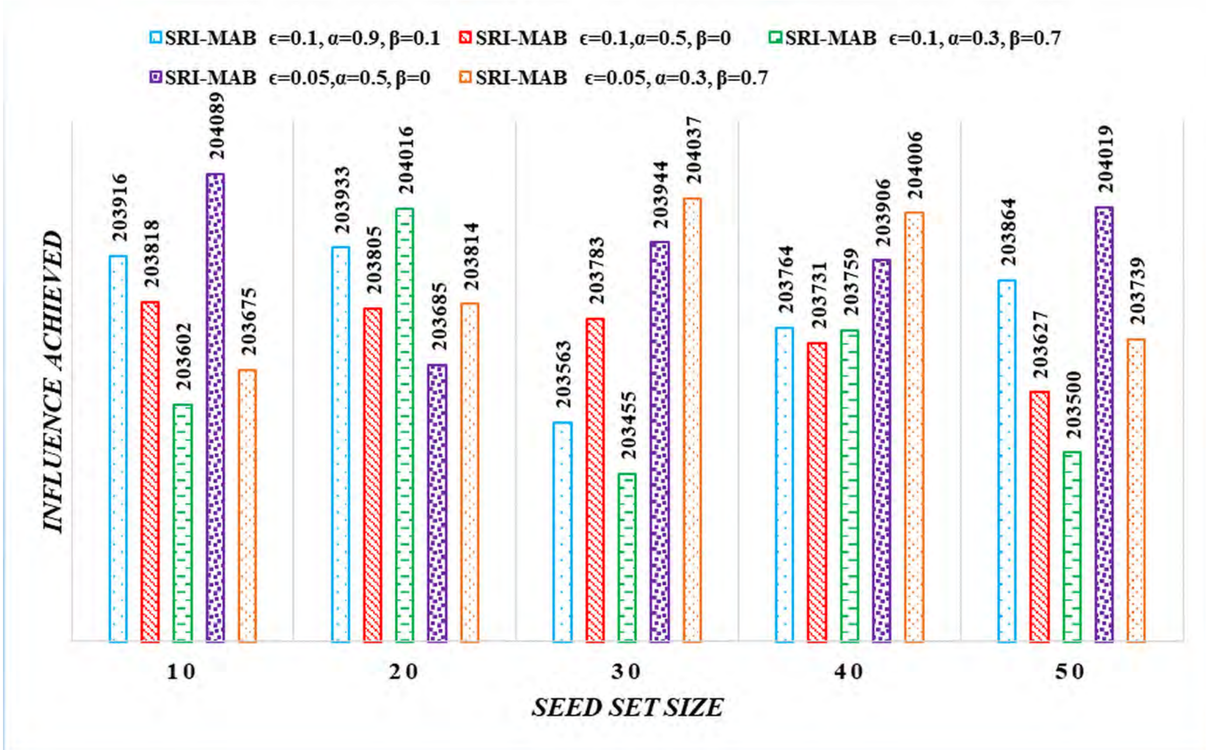

FIGURE 49. Impact of varying α and β on influence spread on Eu-Email data.

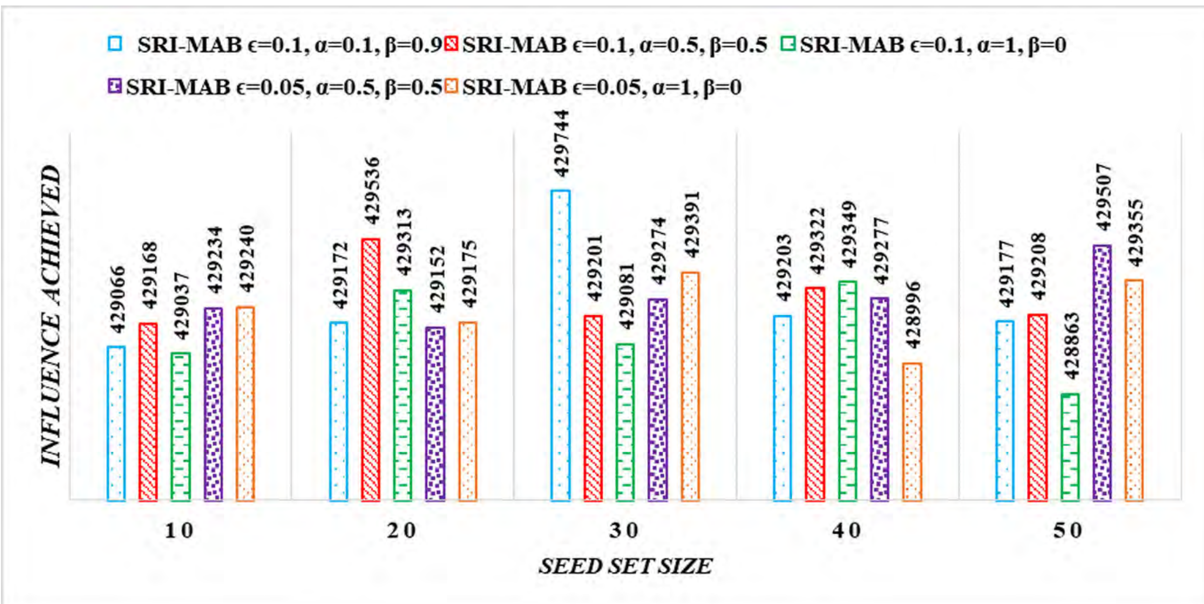

FIGURE 50. Impact of varying α and β on influence spread on Twitter data.

we attempted to test the best values that provide better influence and that efficiently select the seed set that maximizes the spread under the IC model. We notice that the lowest cumulative regret provided better influence spread, especially for the "SRI-CGSS fexpr-greedy" and "SRI-CGSS fexpr-greedy" algorithms that confirm that the lower the cumulative regret, the better the selection of seed set that is made over N rounds on the NetHept, NetPhy and Eu-Email dataset.

These results are based on two assumptions, including that some global structural properties of the graph are known and that in the exploitation and exploration is made with a selection of seed sets with the help of centrality measures according to the designed algorithms. Further investigation should be made regarding the analysis of the designed algorithms' behavior with respect to the ϵ value.

VIII. CONCLUSION

We proposed new methods for identification of relevant influencers based on global and local centrality measures to feed online learning multi-armed bandit algorithms. Additionally, in contrast to existing research works that attempted to learn the probability of diffusion models, and how to infer these propagation probabilities, which is an important direction to understand, analyze and study, our work serves as a ranking method in online sequential decision-making algorithm, where at each round, the individual may select an arm (i.e., edge), and thus, seed sets are selected according to some centrality measures that quantify the best-selected individual. We studied the proposed algorithms from various aspects ranging from the influence spread achieved to the time and space needed to compute the number of influenced individuals. Our methods primarily addressed the selection of the seed set by using exploitation/exploration to select relevant influencers that have a greater tendency to impact their pairs and further the information flow. We showed the impact of the multi-armed bandit reward initialization on the selection process and how the choice of ϵ may rank individuals as potential spreaders differently. Additionally, we tried to adapt proposed algorithms to a different dataset, so that this study can be a benchmark for further study and analysis.

The proposed algorithms can be enhanced and further analyzed to improve the selection of seed set by conducting an in-depth study on multi-armed bandit reward parameters and how it should be chosen to improve results of the selection of arms and thus minimize the cumulative regret over all dataset. Additionally, it is of some interest to consider that the multi-armed bandit reward is unknown or partially unknown where the marketer may have some information about the network structure or no knowledge regarding the network properties that will conform to real-world problems. Thus, as an extension, we will test the approach under different multi-armed bandit algorithms for further investigation.

ACKNOWLEDGMENT

The authors thank Alibaba Cloud Private Co., for their technical support.

REFERENCES

[1] C. Aslay, L. V. Lakshmanan, W. Lu, and X. Xiao, "Influence maximization in online social networks," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 775–776.

[2] Y. Li, J. Fan, Y. Wang, and K. L. Tan, "Influence maximization on social graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1852–1872, Oct. 2018.

[3] D. Bucur, G. Iacca, A. Marcelli, G. Squillero, and A. Tonda, "Improving multi-objective evolutionary influence maximization in social networks," in *Int. Conf. Appl. Evol. Comput.* Cham, Switzerland: Springer Apr. 2018, pp. 117–124.

[4] X. Wang, Y. Zhang, W. Zhang, and X. Lin, "Efficient distance-aware influence maximization in geo-social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 3, pp. 599–612, Mar. 2017.

[5] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2002, pp. 61-70.

[6] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2003, pp. 137–146.

[7] J. Leskovec, A. Krause, C. Guestrin, C.Faloutsos, J. Van Briesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2007, pp. 420–429.

[8] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proc. 20th Int. Conf. Companion World Wide Web*, Mar. 2011, pp. 47–48.

[9] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jun. 2009, pp. 199–208.

[10] M. Alshahrani, F. Zhu, M. Bamiah, S. Mekouar, and S. Huang, "Efficient methods to select top-k propagators based on distance and radius neighbor," in *Proc. Int. Conf. Big Data Comput.*, Apr. 2018, pp. 78–85.

[11] M. Alshahrani, F. Zhu, L. Zheng, S. Mekouar, and S. Huang, "Selection of top-K influential users based on radius-neighborhood degree, multi-hops distance and selection threshold," *J. Big Data*, vol. 5, no. 1, p. 28, 2018.

[12] M. Alshahrani, Z. Fuxi, A. Sameh, S. Mekouar, and S. Huang, "Top-K influential users selection based on combined Katz centrality and propagation probability," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2018, pp. 52–56.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Stanford Digit. Library Technol. Project, Stanford, CA, USA, Tech. Rep., 1999.

[14] D. L. Gibbs and I. Shmulevich, "Solving the influence maximization problem reveals regulatory organization of the yeast cell cycle," *PLoS Comput. Biol.*, vol. 13, no. 6, Jun. 2017, Art. no. e1005591.

[15] J. Sun and J. Tang, "A survey of models and algorithms for social influence analysis," in *Social Network Data Analytics*. Boston, MA. USA: Springer, 2011, pp. 177–214.

[16] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM Sigmod Record*, vol. 42, no. 2, pp. 17–28, Jun. 2013.

[17] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, Oct. 2005, pp. 437–448.

[18] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," 2014, *arXiv: 1402.6028*. [Online]. Available: https://arxiv.org/abs/1402.6028

[19] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, "Online influence maximization," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 645–654.

[20] S. Vaswani, L. Lakshmanan, and M. Schmidt, "Influence maximization with bandits," 2015, *arXiv: 1503.00024*. [Online]. Available: https://arxiv.org/abs/1503.00024

[21] A. Carpentier and M. Valko, "Revealing graph bandits for maximizing local influence," in *Proc. Int. Conf. Artif. Intell. Statist.*, May 2016, pp. 1–11.

[22] S. Vaswani, B. Kveton, Z. Wen, M. Ghavamzadeh, L. V. Lakshmanan, and M. Schmidt, "Model-independent online learning for influence maximization," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 3530–3539.

[23] Z. Wen, B. Kveton, M. Valko, and S. Vaswani, "Online influence maximization under independent cascade model with semi-bandit feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3022–3032.

[24] P. Lagrée, O. Cappé, B. Cautis, and S. Maniu, "Algorithms for Online Influencer Marketing," 2017, *arXiv: 1702.05354*. [Online]. Available: https://arxiv.org/abs/1702.05354

[25] N. Du, Y. Liang, M. Balcan, and L. Song, "Influence function learning in information diffusion networks," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2014, pp. 2016–2024.

[26] K. Kandhway and J. Kuri, "Using node centrality and optimal control to maximize information diffusion in social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1099–1110, Jul. 2017.

[27] F. Riquelme and P. González-Cantergiani, "Measuring user influence on Twitter: A survey," *Inf. Process. Manage.*, vol. 52, no. 5, pp. 949–975, Sep. 2016.

[28] P. A. Estevez, P. Vera, and K. Saito, "Selecting the most influential nodes in social networks," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2007, pp. 2397–2402.

[29] M. Cataldi, C. L. Di, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proc. 10th Int. Workshop Multimedia Data Mining*, Jul. 2010, p. 4.

[30] J. Bollen, M. Rodriquez, and H. Van de Sompel, "Journal status," *Scientometrics*, vol. 69, no. 3, pp. 669–687, Dec. 2006.

[31] L. Lü, Y. C. Zhang, C. H. Yeung, and T. Zhou, ''Leaders in social networks, the *delicious* case,'' *PLoS ONE*, vol. 6, no. 6, 2011, Art. no. e21202.

[32] E. Serin and S. Balcisoy, ''Entropy based sensitivity analysis and visualization of social networks,'' in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2012, pp. 1099–1104.

[33] A. Mochalova and A. Nanopoulos, ''On the role of centrality in information diffusion in social networks,'' in *Proc. ECIS*, 2013, p. 101.

[34] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, ''Everyone's an influencer: quantifying influence on Twitter,'' in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, Feb. 2011, pp. 65–74.

[35] J. Weng, E. P. Lim, J. Jiang, and Q. He, ''Twitterrank: Finding topic-sensitive influential Twitterers,'' in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, Feb. 2010, pp. 261–270.

[36] A. Namtirtha, A. Dutta, and B. Dutta, ''Weighted kshell degree neighborhood method: An approach independent of completeness of global network structure for identifying the influential spreaders,'' in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2018, pp. 81–88.

[37] M. A. Al-garadi, K. D. Varathan, and S. D. Ravana, ''Identification of influential spreaders in online social networks using interaction weighted K-core decomposition method,'' *Phys. A, Stat. Mech. Appl.*, vol. 468, pp. 278–288, Feb. 2017.

[38] Z.-Y. Ding, Y. Jia, B. Zhou, Y. Han, L. He, and J.-F. Zhang, ''Measuring the spreadability of users in microblogs,'' *J. Zhejiang Univ. Sci. C*, vol. 14, no. 9, pp. 701–710, Sep. 2013.

[39] E. M. Azevedo, A. Deng, J. L. M. Olea, J. Rao, and E. G. Weyl, ''The A/B testing problem,'' in *Proc. ACM Conf. Econ. Comput.*, Ithaca, NY, USA, ACM, 2018, pp. 461–462.

[40] M. E. J. Newman, ''A measure of betweenness centrality based on random walks,'' *Social Netw.*, vol. 27, no. 1, pp. 39–54, Jan. 2005.

[41] L. C. Freeman, ''A set of measures of centrality based on betweenness,'' *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[42] L. C. Freeman, S. P. Borgatti, and D. R. White, ''Centrality in valued graphs: A measure of betweenness based on network flow,'' *Social Netw.*, vol. 13, no. 2, pp. 141–154, Jun. 1991.

[43] U. Brandes, ''A faster algorithm for betweenness centrality,'' *J. Math. Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[44] Hoeffding, W, ''Probability inequalities for sums of bounded random variables,'' *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, Mar. 1963.

[45] H. T. Nguyen, T. N. Dinh, and M. T. Thai, ''Cost-aware targeted viral marketing in billion-scale networks,'' in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[46] Y. Tang, X. Xiao, and Y. Shi, ''Influence maximization: Near-optimal time complexity meets practical efficiency,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2014, pp. 75–86.

MOHAMMED ALSHAHRANI received the M.Sc. degree in advance computer science from the University of Liverpool, U.K. He is currently pursuing the Ph.D. degree with Computer School, Wuhan University, China.

He was a Lecturer and the Chairman of the CS Department, Albaha University, Baljurashi Campus, Saudi Arabia. His research interests include graph algorithms, social networks, and sentiment analysis. He is a Student Member of the ACM.

ZHU FUXI was born in 1957. He is currently a Professor and a Ph.D. supervisor with Computer School, Wuhan University, and also a Professor with the Information Engineering College, Wuhan College. His main research interests include artificial intelligence, knowledge mining, and distributed computing.

AHMED SAMEH received the M.Sc. and Ph.D. degrees in computer science from the University of Alberta, Canada, in 1985 and 1989, respectively.

He has held visiting research positions with George Washington University, University of Iowa, and Queen's University. He has been a Professor of computer science and information systems with Prince Sultan University, since 2009. His current research interests include data science, neural networks, and artificial intelligence. He is a member of the ACM, ICS, CIPS, and ISCA. He received the Google Research Award for his life-long contribution to the field of computing throughout his 30 years in the field, in 2005, and chapters, journals, and refereed conferences.

SOUFIANA MEKOUAR received the Ph.D. degree in computer science and telecom from the Faculty of Sciences, Mohammed V Rabat University. She is interested in modeling and analysis of user's behavior and how users react with online resources in online social networks. Her current research studies focus on modeling and simulating social network phenomenon, network science, and data mining.

SICHAO LIU received the Ph.D. degree in computer science from Wuhan University, in 2017. After graduation, he was with Huawei Company for Big Data Analysis. He is currently a Lecturer with Huazhong Agricultural University. His research interests include data mining, social network analysis, and machine learning.