



An efficient and effective GRASP algorithm for the Budget Influence Maximization Problem

Isaac Lozano-Osorio¹ · Jesús Sánchez-Oro¹ · Abraham Duarte¹

Received: 1 February 2022 / Accepted: 21 August 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Social networks are in continuous evolution, and its spreading has attracted the interest of both practitioners and the scientific community. In the last decades, several new interesting problems have aroused in the context of social networks, mainly due to an overabundance of information, usually named as infodemic. This problem emerges in several areas, such as viral marketing, disease prediction and prevention, and misinformation, among others. Then, it is interesting to identify the most influential users in a network to analyze the information transmitted, resulting in Social Influence Maximization (SIM) problems. In this research, the Budget Influence Maximization Problem (BIMP) is tackled. BIMP proposes a realistic scenario where the cost of selecting each node is different. This is modeled by having a budget that can be spent to select the users of a network, where each user has an associated cost. Since BIMP is a hard optimization problem, a metaheuristic algorithm based on Greedy Randomized Adaptive Search (GRASP) framework is proposed.

Keywords Information systems · Social networks · Budget Influence maximization · Viral marketing · GRASP

1 Introduction

The continuous growth of social networks is increasing the data generated by active users exponentially in such a way that problems related to social networks are becoming a challenging task for traditional algorithms. A Social Network (SN) is defined as a set of social interactions among users with the aim of transmitting ideas, propagation of diseases, misinformation detection, or viral marketing, among others (see Reza et al. 2014; Barabási and Pòsfai 2016; Bello-Orgaz et al. 2017; Chen et al. 2020; Tretiakov et al. 2022).

The massive information available nowadays hinders the task of differentiating real from false information. Most of the research related to detecting fake news and misinformation are based on the analysis of the publication content and context-oriented methods, mainly tackled from the Natural

Language Processing area. This research is focused on identifying the most influential users in a Social Network, which may help the algorithms to identify if the source of a piece of information has credibility or not (Noguera-Vivo et al. 2023).

Traditionally, an SN is represented by a graph $G = (V, E)$, where the users are modeled as the set of nodes V and the relation between two users $u, v \in V$ is modeled as an edge $(u, v) \in E$. If there is a relation between two users, then information can be transmitted between them, following one of the Influence Diffusion Models (IDM) which will be described in Sect. 3. Since information can be transmitted in several ways depending on the social network analyzed, or the nature of the relations, Kempe et al. (2003) proposed two different models of information spreading, which have led to a wide variety of new models in the last years.

This paper is intended to deal with a problem in the family of Social Influence Maximization (SIM). It is assumed that if in an SN there exists a relation between two users, then the information can be transmitted from one to another. Without loss of generality, the objective of each variant of SIM is to find a set S of users to start the diffusion of information with the aim of maximizing the scope of the information in terms of the number of users influenced. In the context of infodemics, identifying these users will allow other

✉ Jesús Sánchez-Oro
jesus.sanchezoro@urjc.es

Isaac Lozano-Osorio
isaac.lozano@urjc.es

Abraham Duarte
abraham.duarte@urjc.es

¹ Universidad Rey Juan Carlos, Móstoles, Spain

algorithms to elucidate if it is relevant to analyze the veracity of the information due to the capacity to spread of the user.

The most common variant is named as Social Network Influence Maximization Problem (SNIMP). The objective in this problem is to select a set of k nodes, with $k < n$, in such a way that the number of nodes in the network which are influenced is maximum. This problem has been widely studied in the literature (see Gong et al. 2016; Lozano-Osorio et al. 2021).

However, this variant is not even close to the real SN behavior. In particular, if a company is trying to spread information of their product through the network, the cost of selecting one or another user is not uniform, i.e., some users, which are usually known as influencers, will require a larger budget to be selected than any other anonymous user. The rationale behind this is that the influencer guarantees a larger spreading of the information than the anonymous user.

This paper deals with the Budgeted Influence Maximization Problem (BIMP), originally defined in Nguyen and Zheng (2013), which, instead of selecting a fixed number of initial users, allows us to invest a certain budget in users of the SN, considering that the cost of selecting users is not uniform. Notice that this variant is closer to real SN than SNIMP. In BIMP, the traditional model of SN is still considered, defining a network as a graph $G = (V, E)$, where V is the set of users and E the set of relations among them. However, a function $C : V \rightarrow \mathbb{Z}^+$ is introduced, which assign a non-uniform positive integer cost to every user of the network. Additionally, an initial budget B is given, which is the maximum investment that can be used to select nodes. Each selected node u will decrease the available budget in $C(u)$ units. Then, the BIMP consists of selecting a set of seed nodes S^* that maximizes the information diffusion throughout the network without exceeding the given budget B . More formally,

$$S^* \leftarrow \arg \max_{S \in \mathbb{SS}} IDM(G, S) : \sum_{u \in S} C(u) \leq B$$

where \mathbb{SS} represents all possible combinations of seed sets that can be generated, and IDM is one of the Influence Diffusion Models presented in Sect. 3.

The large amount of data and interest in SN have aroused the interest of both the scientific community and companies in considering BIMP for optimizing the spreading process of a certain message, product, or idea to clients. Marketing agencies like BrandWatch (see Hayes et al. 2021) use this approach when their customers need a commercial campaign based on a certain budget to determine the most effective users to initialize the campaign. Last years, a wide variety of works related to infodemics are focused on pandemic prediction and vaccination discussions (Chen et al. 2020; Bello-Orgaz et al. 2017). The results on BIMP will be able

to identify the most influential users in this context, with the aim of validating their credibility when spreading information and their scope.

In the original definition of BIMP (Nguyen and Zheng 2013), an approximation algorithm is presented which guarantees an approximation ratio of $1 - 1/\sqrt{e}$ is presented. Additionally, they proposed a directed acyclic graph-based heuristic for this problem. This problem has been widely studied mainly due to its practical applications. We refer the reader to Sect. 2 for a detailed review of the literature about BIMP. The main contributions of this research are the following:

- A solution framework based on the Greedy Randomized Adaptive Search Procedure methodology.
- A novel efficient and effective heuristic for selecting the seed set in the constructive phase. This heuristic is experimentally compared with the best previous approaches to show its contribution.
- Three influence diffusion models are tested, instead of just one IDM as in the previous research, showing the robustness of the proposal.
- A scalable algorithm for solving BIMP. Since SNs are exponentially growing, it is necessary to provide a highly-scalable algorithm able to deal with eventually large SNs.
- A comparison of the proposed algorithm with the best methods found in the literature using three publicly available social network datasets which were originally considered in previous works.
- A real-life instance directly related to infodemics is generated based on tweets retrieved from the publicly available dataset called Tweetsets in the area of Healthcare.
- A public repository¹ with the developed code to ease further comparisons.

The remainder of the work is structured as follows. Section 2 reviews the related literature, detailing the different approaches followed to deal with different problems derived from SIM. Then, Sect. 3 introduces the most extended IDMs in the literature, which are also used in this research. The proposed approach is described in Sect. 4, where Sect. 4.1 presents the construction method to provide high-quality initial solutions, and Sect. 4.2 describes the proposed local search to find local optima with respect to a given neighborhood structure. Section 5 presents the experimental results considering a public dataset which has been previously used for this task in order to have a fair comparison, divided into preliminary experiments, which are devoted to adjust the search parameters (Sect. 5.1),

¹ <https://grafo.etsii.urjc.es/BIMP>.

and final experiments, with the aim of performing a competitive testing to evaluate the quality of the proposal (Sect. 5.2). An infodemic case study is developed based on Healthcare tweets in Sect. 5.3. Finally, Sect. 6 draws some conclusions derived from this research.

2 Literature review

The problem of selecting target nodes in SNs to spread information was introduced by Richardson et al. (2003) proposing the first problem formulation. Kempe et al. (2003) presented a heuristic approach to solve the SNIMP, and in Kempe et al. (2015) proved that SNIMP is \mathcal{NP} -hard.

Nguyen and Zheng (2013), formally defined the BIMP inspired by SNIMP, showing its \mathcal{NP} -hardness based on SNIMP. They developed an approximation algorithm which guarantees an approximation ratio of $1 - 1/\sqrt{e}$, being e the base of the natural logarithm. Most of the heuristic proposals for BIMP are inspired by the original algorithms for SNIMP, mainly due to its similarities and the computational effort required to evaluate the IDM. Kempe et al. (2003) presented several greedy heuristics with an approximation of $1 - 1/e - \epsilon$, where ϵ is any positive real number. When the considered greedy function of the heuristic is the degree of the node, the algorithm is called *high-degree heuristic*. Based on the node degree idea, Chen et al. (2010) proposed a new greedy function to optimize the high-degree heuristic, such as the greedy selection function considering the redundancy between likely influenced nodes, but discarding those reached by the already selected seed nodes, in order to provide a better estimation of the total spread.

Han et al. (2014) proposed a set of heuristics for optimizing BIMP by considering influential nodes and cost-effective nodes to increase both accuracy and effectiveness. Later on, Guney et al. (2015) proposed a sample average approximation method for BIMP, which is able to reach almost near optimal solutions.

Banerjee et al. (2020) published the latest survey in SIM, becoming one of the most relevant research works in the area of influence maximization problems. The algorithm named CombBIM proposed by Banerjee et al. (2019) is considered the state of the art for BIMP. CombBIM provides a community-based solution that provides the best results in the literature as far as our knowledge, so it will be considered as the algorithm to benchmark our proposal. Recently,

Lozano-Osorio et al. (2021) proposed a new heuristic method for selecting the seed set in the context of SNIMP. This work adapts this heuristic with the aim of evaluating the performance of the proposal over a different variant of the same family of problems.

3 Influence diffusion model

The evaluation of the influence of a given seed set S over a network G requires the definition of an Influence Diffusion Model (IDM). An IDM is responsible for deciding which nodes are affected or influenced by the information received from their neighbors in the SN. The most extended IDMs in the literature are: Independent Cascade Model (ICM) (see Kempe et al. 2003; Goyal et al. 2011), Weighted Cascade Model (WCM) (see Kempe et al. 2003), and Tri-Valency Model (TV) (see Granovetter 1978). All of them are based on assigning an influence probability to each relational link in the SN, since a relation in a network does not necessarily implies that a user influence another one in a certain period of time.

- ICM, which is one of the most used IDMs, considers that the influence probability is the same for each link.
- WCM considers that the probability of a user v for being influenced by user u is proportional to the in-degree of user v , i.e., the number of users that can eventually influence user v . Therefore, the probability of influencing user v is defined as $1/d_{in}(v)$, where $d_{in}(v)$ is the in-degree of user v .
- TV select the edge probability randomly from the set of probabilities (1%, 0.1%, 0.001%).

Following the recommendations of the literature and, more specifically, the state of the art for BIMP, the three aforementioned IDMs are evaluated in this work. The only IDM that requires for an input parameter is ICM, where the probability values are set to 1% and 2%, as stated in Banerjee et al. (2019).

Due to the probabilistic nature of the IDM, the most extended way of evaluating the spread is by conducting a Monte Carlo simulation (MC). However, even a single iteration of the simulation model is rather time-consuming when considering large graphs derived from SN. Algorithm 1 shows the pseudocode of the Monte Carlo simulation used to evaluate the spread of information through an SN named G given a certain seed set S . Specifically, it

receives four input parameters: the graph which models the SN, G ; a solution, S ; the IDM used to select the diffusion probability of a node to be influenced, IDM ; and the number of repetitions r performed to avoid the impact of randomness.

Algorithm 1 $MC(G = (V, E), S, IDM, r)$

```

1:  $I \leftarrow 0$ 
2: for  $i \in 1 \dots r$  do
3:    $A^* \leftarrow S$ 
4:    $A \leftarrow S$ 
5:   while  $A \neq \emptyset$  do
6:      $D \leftarrow \emptyset$ 
7:     for  $v \in A$  do
8:       for  $(u, v) \in E$  do
9:         if  $rnd(0, 1) \leq IDM(u)$  then
10:           $D \leftarrow D \cup \{u\}$ 
11:        end if
12:      end for
13:    end for
14:     $A \leftarrow D \setminus A^*$ 
15:     $A^* \leftarrow A^* \cup D$ 
16:  end while
17:   $I \leftarrow I + |A^*|$ 
18: end for
19: return  $I/r$ 

```

The algorithm starts by initializing the total number of infected users (step 1). Then, the algorithm performs a number of predefined simulations r (steps 2–18), finding in each iteration which are the influenced nodes starting from the given seed set S . Initially, the set of nodes A^* reached by the initial seed set, S , is actually the seed set (step 3). Then, the method iterates until no new nodes are influenced (steps 5–16). In each iteration of the inner for-loop, the method evaluates the IDM for each node directly related to a recently influenced node (steps 8–12). For each neighbor, a random number is generated. If this number is smaller than the probability of infection p that is determined by the IDM used, then it is considered that the neighbor becomes infected (steps 9–11). At the end, the set of the nodes infected in the previous iteration (step 14) that are not just analyzed as well as infected nodes is updated (step 15). Finally, the algorithm returns the average number of infected nodes among all simulations performed (step 19). Notice that this value is considered as the objective function to be optimized when solving the BIMP or, generally, any SIM problem. Therefore, the seed set which maximizes the *spread value* over the network would compose the optimal solution to the problem. It is worth mentioning that, as infection is a stochastic process, the IDM must be executed a considerably large number of iterations to achieve an appropriate estimation, thus resulting in a Monte Carlo simulation.

4 Algorithmic approach

The proposed algorithm follows the Greedy Randomized Adaptive Search Procedure (GRASP) methodology, which was originally introduced by Feo and Resende (1989) and formally defined in Feo et al. (1994). We refer the reader to Resende Mauricio et al. (2010), Resende Mauricio and Ribeiro (2013) for a complete survey of the last advances in this methodology.

GRASP is a multistart metaheuristic, divided into two distinct phases: construction and local improvement. The first phase consists of a greedy, random, and adaptive construction of a solution, in order to provide a promising starting point. The second phase consists of a method to locally improve the constructed solution to a local optimum with respect to a given neighborhood.

A recent proposal by Lozano-Osorio et al. (2021) uses GRASP method to solve SNIMP, since GRASP methodology is able to find a trade-off between diversification in the stochastic construction phase and the intensification of the local search process, enabling the algorithm to escape from local optima and perform a wider search space exploration.

These two phases are repeated until a termination criterion is met. Notice that this criterion makes the algorithm scalable to eventually large social networks, since the termination criterion can be tuned to perform a smaller number of iterations.

In the context of BIMP, a novel heuristic for selecting the most promising nodes in the construction phase is proposed. Additionally, the local improvement phase considers a move based on the replacement of nodes and it can be limited to avoid large computing times without significantly deteriorating the quality of the solutions found.

4.1 Construction phase

The purpose of the GRASP construction phase is to generate a promising initial solution in a short computing time. In order to do this, the construction phase is usually guided by a greedy selection function, which helps the constructive method to select the most promising elements to be included in the partial solution (see Algorithm 2). It is worth mentioning that, in the context of BIMP, the computational effort required to evaluate the greedy function value should be minimal, since the size of the social networks might lead the algorithm to be extremely slow when a solution is constructed.

Algorithm 2 *Construct*($G = (V, E), \alpha, B$)

```

1:  $v \leftarrow \text{rnd}(V)$ 
2:  $S \leftarrow \{v\}$ 
3:  $B \leftarrow B - C(v)$ 
4:  $CL \leftarrow \{c \in V : c \notin S \wedge C(c) \leq B\}$ 
5: while  $CL \neq \emptyset$  do
6:    $g_{\min} \leftarrow \min_{u \in CL} g(u)$ 
7:    $g_{\max} \leftarrow \max_{u \in CL} g(u)$ 
8:    $\mu \leftarrow g_{\max} - \alpha \cdot (g_{\max} - g_{\min})$ 
9:    $RCL \leftarrow \{v \in CL : g(v) \geq \mu\}$ 
10:   $u \leftarrow \text{rnd}(RCL)$ 
11:   $S \leftarrow S \cup \{u\}$ 
12:   $B \leftarrow B - C(u)$ 
13:   $CL \leftarrow \{c \in CL : c \notin S \wedge C(c) \leq B\}$ 
14: end while
15: return  $S$ 

```

The algorithm starts by randomly selecting the first node to be included in the solution S at random from the set of nodes V (steps 1–2). The random selection of the first element to be included in the solution is customary in GRASP since it favors diversification. When selecting the first node v , the available budget is decremented with the cost of v , $C(v)$ (step 3) The candidate list CL is then created with all nodes whose cost is smaller than the available budget B which are not already in the solution S (step 4). Then, the constructive method iteratively adds new elements to the solution until no candidate nodes with a cost smaller than B are available to be selected (steps 5–14). In each iteration, the minimum and maximum value of the greedy heuristic function is evaluated (steps 6–7). Since the greedy function is a key feature of GRASP, hereinafter each greedy function considered is described. Then, a threshold μ is calculated (step 8), which is required for creating the Restricted Candidate List (RCL) with the most promising nodes (step 9). This threshold directly depends on the value of the input parameter α , with $0 \leq \alpha \leq 1$. Notice that this parameter indicates the greediness or randomness of the constructive procedure. On the one hand, if $\alpha = 0$, then the threshold is evaluated as g_{\max} , becoming a totally greedy algorithm (i.e., the RCL only includes those nodes with the maximum greedy function value). On the other hand, if $\alpha = 1$ then $\mu = g_{\min}$, resulting in a completely random method (i.e., the RCL includes every candidate node whose cost is smaller than the available budget). Since this parameter is experimentally tuned, we refer the reader to Sect. 5 to analyze the effect of different values for the α parameter in the final algorithm. Then, the next node is selected at random from the RCL (step 10), including it in the solution under construction (step 11), updating the budget by reducing it with the cost of the selected node (step 12). The CL is also updated (step 13) in the same way as (step 4), being candidate nodes whose associated cost is smaller than the remaining budget.

The method ends when no more elements can be included in the seed set (i.e., there is no candidate node whose cost is smaller than the available budget), returning the constructed solution S (step 15).

As it was aforementioned, the greedy heuristic function g used in steps 6–7 is one of the key features when designing a constructive procedure in the context of GRASP. In particular, this greedy function must select the most promising nodes without requiring large computing times. In this work, we adapt two existing greedy functions originally proposed for SNIMP, and propose a novel one to analyze its performance against the well-established ones.

The first greedy function, named g_{deg} , considers the out-degree of a node as heuristic value. Given a node u , let us define out-degree as $d_u^+ = |N_u^+|$, where $N_u^+ = \{v \in V : (u, v) \in E\}$ In mathematical terms,

$$g_{deg}(u) = d_u^+$$

The second greedy function, named g_{2step} , was originally used for SNIMP (Lozano-Osorio et al. 2021). It is a heuristic based on the first and second degree neighbors of a given node, usually known as 2-step neighbors in the context of SN analysis (Stanley and Katherine 1994). The evaluation of this greedy function over a certain node u can be formally defined as:

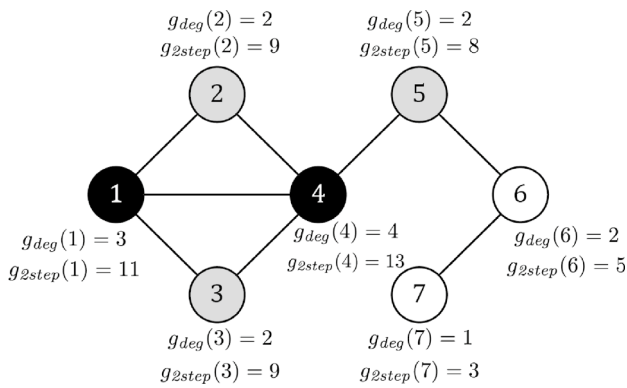
$$g_{2step}(u) = d_u^+ + \sum_{v \in N_u^+} d_v^+$$

Additionally, this work proposes a novel heuristic, named g_{dist} which leverages the node seed distribution. This method prioritizes nodes that do not have selected neighbors as a seed node, with the aim of reaching a larger number of non-influenced users by exploring regions that have been mainly ignored until that point. In order to do so, the greedy function value of a node is directly its degree, but penalizes it if some of its neighbor nodes have already been selected. The penalization has been experimentally set by halving the degree. More formally

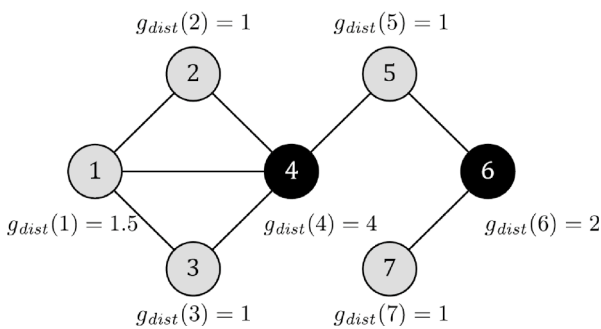
$$g_{dist} = \begin{cases} d_u^+ & \text{if } v \notin S, \forall v \in N_u^+ \\ \frac{d_u^+}{2} & \text{otherwise} \end{cases}$$

Let us illustrate the behavior of each proposed greedy heuristic function with an example of SN with 7 nodes and 8 relations, depicted in Fig. 1. The value of each heuristic function is presented close to each node.

Figure 1a shows the evaluation of g_{deg} and g_{2step} , since both of them result in the same solution. In the case of g_{deg} , the first selected node is 4, since it is the node with the largest degree. Then, node 1 is selected as the one with the second largest degree. This seed set is able to directly influence up to three nodes, reducing the possibility of influencing



(a) Solution $S_1 = \{1, 4\}$, considering g_{deg} and g_{2step} heuristic, which may influence up to three nodes: 2, 3, and 5.



(b) Solution $S_2 = \{4, 6\}$, considering the g_{dist} heuristic, which may influence all nodes: 1, 2, 3, 5 and 7.

Fig. 1 Evaluation of the three considered heuristic functions to select the initial seed set over a SN with 7 nodes and 8 relations

nodes 6 and 7. In particular, there will be a possibility of influencing nodes 6 and 7 if and only if node 5 is influenced. The same behavior can be seen when considering g_{2step} : the first selected node is 4, which presents the largest value and, then, node 1 is selected. Finally, Fig. 1b shows the resulting solution when considering g_{dist} heuristic. In this case, the first node is selected with respect to its degree, resulting in node 4. Then, the heuristic value of the nodes directly connected to node 4 is evaluated as their degree reduced by half, while the heuristic value of non-directly connected nodes still remains as their degree. Then, the second node selected is 6, which presents the largest value for g_{dist} . Notice that, in this example, both g_{deg} and g_{2step} reports the same solution, although the value of each greedy function is different. However, the idea of penalizing those nodes connected to already selected ones, used in g_{dist} , lead the constructive procedure to reach a better solution. The impact and the influence of each greedy constructive procedure will be deeply analyzed in Sect. 5.

4.2 Local improvement

The second phase of GRASP is responsible for locally improving each solution generated by the constructive procedure with the aim of reaching a local (ideally global) optimum. In the context of GRASP, this phase can be accomplished by using a simple local search procedure or a more complex heuristic (even a complete metaheuristic) like Tabu Search (see Martí et al. 2018). The elevated complexity of the problem under consideration has led us to propose a simple yet effective local search procedure to reduce the computational effort required.

Before defining a local search method, it is necessary to introduce the neighborhood to be explored. The neighborhood of a solution S is defined as the set of solutions that can be reached by performing a single move over S . Then, it is necessary to define the move that will be considered in the context of BIMP. Specifically, the move, named as $Replace(S, u, P)$, involves removing node u from the solution and replacing it with the set of nodes in P , with $P \in V \setminus S$. Notice that, in order to reach a feasible solution, the sum of the cost of nodes in P must be smaller or equal than $B + C(u)$ (since u will be removed, its cost must not be taken into account). More formally,

$$Replace(S, u, P) = S \setminus \{u\} \cup P$$

Then, given a solution S , the neighborhood $N_R(S)$ is defined as the set of feasible solutions that can be reached with a single $Replace$ move. In mathematical terms,

$$N_R(S) = \{S' \leftarrow Replace(S, u, P) \mid \forall u \in S \wedge \forall P \in V \setminus S: \sum_{p \in P} C(p) \leq B + C(u)\}$$

Having defined the neighborhood which will be explored in the local search, the next step consists of defining the way in which the neighborhood $N_R(S)$ is explored. Even considering an efficient implementation of the objective function evaluation, the vast size of the resulting neighborhood makes the complete exploration of the neighborhood not suitable for the BIMP. Therefore, we limit the number of evaluations that the local search performs with the aim of having a computationally efficient method. It is worth mentioning that, if the number of iterations Ψ is limited, then it is interesting to firstly explore the most promising neighbors of the considered neighborhood. Therefore, an intelligent neighborhood exploration strategy is presented.

Hansen and Mladenović (2006) performed an empirical study on the well-known Traveling Salesman Problem to compare first and best improvement strategies in the context of local search. The authors conclude that both strategies present similar results in terms of quality, but the first improvement approach is faster when considering randomness in the constructive phase. Following their recommendations and due to the computational effort required to evaluate

a solution for the BIMP, we propose a first improvement approach. This strategy does not need to explore all solutions in the neighborhood for each iteration, thus reducing the number of objective function evaluations required and consequently the overall run time.

The proposed strategy filters the nodes that are involved in every iteration of the local search method. In particular, the nodes considered for removal from the solution S are selected at random, but the ones to be later included are selected by their contribution to the objective function value if they are included in the incumbent solution. Notice that evaluating the contribution requires performing a Monte Carlo simulation, which is rather time consuming. With the aim of reducing the computational effort of this evaluation, a single Monte Carlo execution is performed, i.e., the value of r in Algorithm 1 is an input parameter of the local search method named Δ (see Sect. 5.1 where an experiment to analyze the performance of r value is done). Furthermore, in order to increase the efficiency, the Monte Carlo simulation is not performed from scratch. Instead, since the solution has already been evaluated, the influenced nodes are known. Then, to evaluate the contribution of inserting a new node v in the solution, it is only required to evaluate which are the new nodes influenced by v , resulting in an efficient way of estimating the contribution of including v in the incumbent solution. Then, the candidate nodes to be included are those with the largest contribution to the objective function value. The number of candidates to be evaluated is determined by the maximum number of evaluations Ψ (see Sect. 5.1 where an experiment with different Ψ values is carried out). The pseudocode of the local search LS is shown in Algorithm 3.

Algorithm 3 $LS(G, IDM, S, \Psi, \Delta)$

```

1:  $R \leftarrow rnd(S, \Psi)$ 
2: for  $u \in R$  do
3:    $\Psi = \Psi - 1$ 
4:    $P^* \leftarrow \arg \max_{p \in V \setminus S} MC(G, S, IDM, \Delta) : \sum_{p \in P} C(p) \leq B + C(u)$ 
5:    $S' \leftarrow Replace(S, u, P^*)$ 
6:   if  $MC(G, IDM, S', \Delta) > MC(G, IDM, S, \Delta)$  then
7:      $S \leftarrow S'$ 
8:     go to 1 ▷ Improvement found
9:   end if
10: end for
11: return  $S$ 

```

The method starts by creating the set of nodes whose removal is tested (step 1). In particular, it consists of a random set of Ψ nodes extracted from the nodes which are not already in the solution. If the number of available nodes is smaller than Ψ , then all nodes are candidates. For each candidate node (steps 2–10), the available number of evaluations is decremented (step 3) and, then, the set of most promising nodes to be included P^* is created as those maximizing

the contribution to the objective function value if included in the solution satisfying the cost constraint when removing u from S (step 4). Once both the candidate node to be removed u and the set of most promising nodes to be included P^* are selected, the *Replace* move is performed, resulting in a neighbor solution S' (step 5). Then, S is updated if S' results in a better solution (step 7), restarting the search since the local search method follows a first improvement strategy (step 8). The method ends returning the best solution found during the search (step 11).

5 Computational experiments and analysis of results

The aim of this section is to describe the computational experiments designed to evaluate the performance of the proposed algorithms and to analyze the obtained results. All experiments have been performed in an Intel Core i7-9750 H (2.6 GHz) with 16GB RAM and the algorithms were implemented using Java 17 and the *Metaheuristic Optimization framework*² (MORK) 10, designed to facilitate the implementation of algorithms for solving \mathcal{NP} -hard problems. The source code of the proposed methods has also been made publicly available.³

The set of SNs considered in this paper has been entirely obtained from the best algorithm found for BIMP in the literature, to provide a fair comparison among the analyzed algorithms. All of them are publicly available in Stanford Network Analysis Project (SNAP).⁴ The datasets used were the Epinions dataset (Richardson et al. 2003; Xu et al. 2016) which consists of 75879 nodes and 508837 edges, the dataset HepTh with 27770 nodes, and 352807 edges and finally CondMat which has 23133 and 93497 edges (Leskovec et al. 2007).

Since this research is designed for improving the analysis of users in an infodemics context, we have generated a real-life instance with tweets related to a health case published about the announcement of the American Health Care Act (AHCA) in 2017. The new dataset contains 54836 nodes and 89059 edges (we refer the reader to Sect. 5.3 for a more detailed description about this instance).

Table 1 shows the following details of the datasets used: number of nodes in the largest connected-component (LCC), total number of connected-components

² <https://github.com/rmartinsanta/mork/>.

³ <https://grafo.etsii.urjc.es/BIMP>.

⁴ <https://snap.stanford.edu/>.

Table 1 Metrics of the used datasets

Instance	Nodes	Edges	lLCCI	TC	ADPN
soc-Epinions1	75879	405740	75877	2	10.69
HC Twitter	54836	89059	47257	3060	3.25
CA-CondMat	23133	93497	21363	567	8.08
CA-HepT	9877	25998	8638	429	5.26

(TC) and average out degree for all nodes (more formally $\frac{1}{n} \sum_{i=1}^n (deg(v_i))$).

It is worth mentioning that the original SNs derived from SNAP do not have any weight in the nodes. In the context of BIMP, every node has an associated cost to be selected, so it is necessary to perform this assignment. With the aim of having a fair comparison, we contact the authors of the previous work for their exact cost assignment and the source code to execute the algorithm in the same platform. Unfortunately, we did not receive any response, so we implement their algorithm carefully following the detailed description provided in the manuscript (Banerjee et al. 2019). Additionally, we generate a random uniform cost for each node following the suggestions of the previous authors. In order to ease further comparisons, we have made publicly available the exact instances used in this work.

First of all, it is important to indicate the number of repetitions performed in the Monte Carlo simulation. As it is customary in SIM problems, 100 Monte Carlo simulations are performed on all IDMs models. The total budget B to conform a solution is selected in the range $B = \{2000, 6000, 10000, 140000, 180000, 22000, 26000\}$ as stated in Banerjee et al. (2019), thus obtaining $3 \cdot 7 = 21$ different problem instances for each IDM. Taking into account that 4 IDMs are considered as described in Sect. 3, the total number of instances are $21 \cdot 4 = 84$.

The experiments are divided into two parts: preliminary and final experimentation. The former (Sect. 5.1) refers to those experiments performed to select the best parameters to set up our algorithm, while the latter (Sect. 5.2) validates the best configuration, comparing it with the best method found in the state of the art.

All experiments developed report the following performance metrics: Avg., the average of the number of influenced nodes; Time (s), the average execution time of the algorithm measured in seconds; Dev (%), the average deviation with respect to the best solution found in the experiment, evaluated as $\frac{f_{best} - f_a}{f_{best}} \cdot 100$, where f_{best} is the objective function of the best solution found in the experiment and f_a is the objective function value of the best solution found by the algorithm; and finally, #Best, the number of times that the algorithm matches the best solution in the experiment. Tables report a summary to provide a global view of each

Table 2 Results of the constructive procedure when generating 50 solutions, considering different α values for every heuristic function

Greedy function	α	Avg.	Time (s)	Dev (%)	#Best
g_{dist}	0.25	3446.91	14.96	0.87	6
	0.50	3430.71	14.67	2.06	4
	0.75	3346.41	14.00	7.14	1
	RND	3462.20	14.67	0.81	10
g_{deg}	0.25	3337.21	9.77	5.54	2
	0.50	3338.72	9.38	5.73	2
	0.75	3332.81	9.06	6.96	0
	RND	3417.53	9.68	3.44	6
g_{2step}	0.25	3146.76	9.60	11.92	0
	0.50	3158.04	9.20	12.28	0
	0.75	3172.05	9.38	11.60	0
	RND	3246.42	9.72	9.05	0

Best results are highlighted with bold font

algorithm by averaging the results obtained along all the considered instances. Individual results per instance are included in the public repository, where the code is also available.

5.1 Preliminary experimentation to setup the final GRASP method

In the preliminary experiments, 6 representative SNs are evaluated with each IDM, resulting in 24 instances. The set of preliminary instances comprehends the 28% of the total set of 84 instances. This selection of instances is done to avoid overfitting in the model.

The purpose of the first preliminary experiment is to obtain the best greedy heuristic function together with the value of α . For this purpose, all greedy heuristic methods, g_{deg} , g_{2step} , and g_{dist} , have been analyzed when considering $\alpha = (0.25, 0.50, 0.75, RND)$. Notice that $\alpha = RND$ indicates that a random value in the range 0-1 is selected for each construction. The GRASP method used 50 iterations in all experiments.

Table 2 collects the final results from this competitive testing. Notice that Avg. is not an integer value since it is the average value of the 100 repetitions of the Monte Carlo simulation.

As it can be drawn from the table, the best results are consistently provided by the greedy function based on the new heuristic procedure, g_{dist} . In particular, the best results are obtained when considering $\alpha = RND$, with 10 best solutions and 0.81% of average deviation. The small deviation value indicates that, even in the cases in which it is not able to reach the best solution, it remains really close to it. It is worth mentioning that the heuristic g_{2step} , which is the

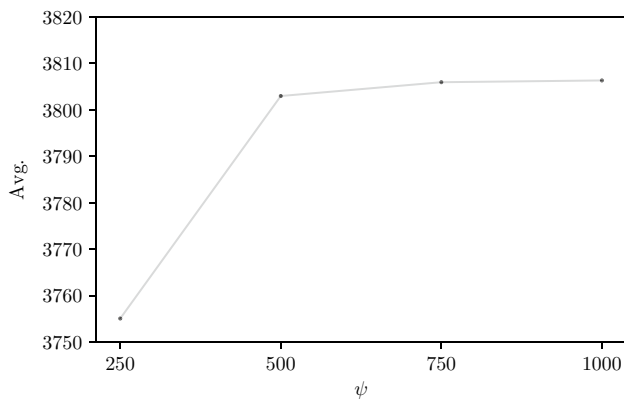


Fig. 2 Analysis of the effect of the value of Ψ in the local search phase

Table 3 Results of the GRASP algorithm when generating 50 solutions, considering $\alpha = RND$ versus the constructive procedure executed isolatedly

IDM	Method	Avg.	Time (s)	Dev (%)	#Best
ICM(1%)	g_{dist}	3379.44	18.79	15.62	0
	GRASP	4005.01	31.95	0.00	9
ICM(2%)	g_{dist}	7246.65	26.98	7.94	0
	GRASP	7872.00	62.43	0.00	9
WC	g_{dist}	2477.73	15.24	4.81	0
	GRASP	2603.01	21.07	0.00	9
TV	g_{dist}	615.81	8.85	39.75	0
	GRASP	1022.11	12.11	0.00	9
Summary	g_{dist}	3461.23	17.47	9.96	0
	GRASP	3844.21	31.89	0.00	36

Best results are highlighted with bold font

best greedy function in the context of SNIMP in Lozano-Osorio et al. (2021), produces the worst values in the context of BIMP, independently of the selected α -value. This result highlights the relevance of proposing new heuristics for this problem. Regarding the computing time, although g_{dist} requires slightly larger computing times on average, the difference with the other greedy heuristic functions are negligible. Therefore, we select g_{dist} as the best constructive procedure with $\alpha = RND$.

The next experiment is devoted to analyze the effect of the maximum number of iterations Ψ in the local search phase in terms of quality and computing time. Figure 2 shows the improvement when increasing the value of Ψ . Notice that the quality of the solutions significantly improves upon reaching $\Psi = 500$. At that point, the search seems to stagnate and no considerable improvement is found, thus leading us to select $\Psi = 500$ for the local search phase.

Table 4 Competitive testing of the proposed GRASP algorithm with respect to state of the art algorithm ComBIM

IDM	Algorithm	Avg	Time (s)	Dev (%)	#Best
ICM(1%)	ComBIM	8319.68	214.97	17.64%	0
	GRASP	8872.61	117.06	0.00%	21
ICM(2%)	ComBIM	14467.65	215.31	6.49%	3
	GRASP	14828.77	146.21	0.07%	18
WC	ComBIM	2277.79	214.04	57.49%	0
	GRASP	10087.08	97.80	0.00%	21
TV	ComBIM	1976.11	214.68	39.10%	0
	GRASP	2677.58	69.65	0.00%	21
Summary	ComBIM	6760.31	214.75	30.18%	3
	GRASP	9116.51	107.68	0.02%	81

Best results are highlighted with bold font

The third preliminary experiment is devoted to analyze the contribution of the local search phase in the complete GRASP algorithm. In order to do so, the constructive procedure considering g_{dist} and $\alpha = RND$ is executed and compared with the complete GRASP framework. The results are shown in Table 3.

As it can be derived from the results, the local search requires twice the computational time than the constructive procedure isolatedly, on average, in each IDM. This increase is justified since it is able to reach a considerably better solution, as it can be seen in the large average deviation values presented by the constructive procedure without local search. In particular, the average deviation is 9.96% on average, reaching a maximum value of 39.75% in the case of TV. Notice that, in the case of TV, the random selection of the probability of being influenced affects on the obtained results, being ICM and WC are more robust in the comparison. Regarding the number of best solutions found, it can be seen how the local search phase is able to improve the initial solutions, since the constructive procedure is not able to reach any best value.

Having performed the preliminary experiments, the best results are obtained with the following values: the greedy heuristic function selected is g_{dist} , the parameter of the constructive procedure is set to $\alpha = RND$, the maximum number of evaluations is $\Psi = 500$, and the number of Monte Carlo simulations is set to $\Delta = 10$ in the local search (this value has been set since no significant differences have been found testing different values in the range [10,100]). These parameter values will be used to set up the final version of the algorithm.

5.2 Competitive testing

In order to analyze the quality of the proposed algorithm, a competitive testing is performed with the best method found in the state of the art, ComBIM, by considering the complete set of 84 instances.

Table 4 collects the final results obtained in this competitive testing, where for each IDM the same metrics as in the preliminary experimentation are reported: Avg., Time (s), Dev(%), and #Best.

The results show how GRASP is able to obtain high-quality solutions (81 best solutions out of 84), and this values are obtained in half of the computing time (107.68 s vs 214.75 s). Although GRASP is able to outperform ComBIM in all IDMs considered, the most remarkable results in terms of quality are obtained when using WC and TV. Specifically, ComBIM is able to reach the best solution just in three instances when using ICM (2%). In this case, the deviation of GRASP is 0.07%, indicating that it is really close to that best solution. In view of these results, GRASP emerges as one of the most competitive algorithms for BIMP.

We finally perform over all instances the well-known non-parametric Wilcoxon statistical test for pairwise comparisons, which answers the question: do the solutions generated by both algorithms represent two different populations? The resulting p -value smaller than 0.0001 when comparing GRASP with ComBIM confirms the superiority of the proposed GRASP algorithm. In particular, GRASP is able to obtain 81 out of 84 positive ranks, 3 negative ranks, and 0 ties. Therefore, GRASP emerges as one of the most competitive algorithms for the BIMP, being able to reach high-quality solutions in small computing times.

5.3 An infodemic case study

This section shows an infodemic case study, based on tweets retrieved from the George Washington University's publicly available dataset called Tweetsets (Wrubel et al. 2020). Existing tweets where a user shares a tweet, that means that the user has been influenced by the original tweet, are used to build this instance. The tweetset used is related to infodemics in the area of Healthcare, related to the announcement of the American Health Care Act (AHCA) in 2017. This dataset consists of 386384 tweets, where 284131 are retweets. The original dataset contains all the identifiers of the tweets and, in order to generate this instance, we have retrieved it from Twitter, resulting in 96705 tweets. Notice that 187426 tweets have been removed from Twitter due to fake news filters or suspended accounts (Tretiakov et al. 2022). The final dataset has 54836 users, 96705 tweets, and 2060 components, where the largest component has 47257 nodes. The available budget for BIMP is generated following the same procedure of the previous instances: a random

Table 5 Competitive testing of the proposed GRASP algorithm with respect to state-of-the-art algorithm ComBIM

IDM	Algorithm	Avg	Time (s)	Dev (%)	#Best
ICM(1%)	ComBIM	6621.69	318.91	0.65%	1
	GRASP	6663.23	42.39	0.04%	6
ICM(2%)	ComBIM	12833.06	319.01	0.00%	6
	GRASP	12722.24	55.70	0.86%	1
WC	ComBIM	23898.77	319.32	23.09%	0
	GRASP	31267.49	95.12	0.00%	7
TV	ComBIM	1845.83	318.69	17.48%	0
	GRASP	2247.41	95.12	0.00%	7
Summary	ComBIM	11299.84	318.98	10.31%	7
	GRASP	13225.09	72.08	0.22%	21

Best results are highlighted in bold font

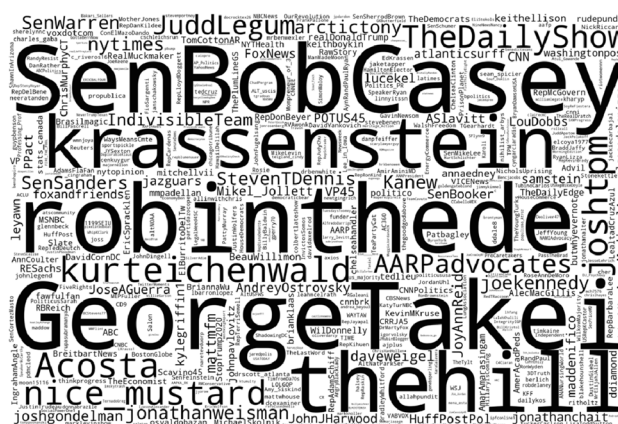


Fig. 3 Most influential users in the HC Twitter dataset detected by GRASP over different IDM

uniform cost is generated for each node. In order to compare our proposal, a competitive testing is performed.

Table 5 shows the results resulting from the comparison of GRASP and ComBIM over this case study. As it can be derived, GRASP obtains 21 best solutions out of 28, requiring less than four times the computing time from ComBIM (72.08 vs 318.98 s). ComBIM method performs better with the ICM influence diffusion model than WC and TV, showing the same behavior as in the previous instances. On the contrary, our method adapts to each IDM due to the evaluation of the objective function with the Monte Carlo simulation. We perform the Wilcoxon non-parametric statistical test resulting in a p -value smaller than 0.0001, confirming that GRASP is statistically better than ComBIM.

Having selected the most influential users with GRASP, it is necessary to analyze who are those users and how they are related to the context under evaluation. For this purpose, a word cloud has been constructed so that the weight of a user is directly proportional to the times that it has selected

by GRASP as an influential user (seed node) for each IDM. This weight determines the size of the font used in the word cloud.

Figure 3 highlighted the most influential users. For instance, *SenBobCasey* is an US senator from the Democratic Party which was really active in the context of Health Care; *krassenstein* is the social account of a famous independent investigative journalist focused on detecting hate in infomedics; *robinthede* and *GeorgeTakei* are writers and famous comedians who published some viral jokes about this context; and *thehill*, which is an American newspaper and political journalism website published in Washington D. C. since 1994. All these accounts were really active with the Health Care proposal (both supporting or opposing it), with more than 9 million followers as a whole. This result suggests that the information spread by these users should be carefully analyzed mainly due to the high impact of diffusion.

6 Conclusions

In this paper, an efficient and effective GRASP algorithm for solving the BIMP has been presented. Three different diffusion models are used for BIMP considering the probabilistic algorithm Monte Carlo for the evaluation of the objective function.

Three greedy heuristic functions have been proposed for generating the initial solutions of GRASP. The first one is based on the two-step neighborhood, recently published in a problem of the same family, with the aim of analyzing how it adapts to a similar problem. The second one is based on the degree of each node and, finally, a new heuristic that penalizes those nodes whose neighbors have been already selected as a seed node is proposed, with the aim of expanding to new regions in the graph. Furthermore, the idea of using local information allows the algorithm to construct a complete solution in small computing time.

The local search method proposed is based on a new move named *Replace*, whose objective is to remove a seed node replacing it with the most promising ones. To make a scalable algorithm, and to avoid an exhaustive search which is not suitable for this problem, the local search is limited and can be configured according to the time requirements.

Comparing the presented GRASP algorithm versus the best algorithm found in the state of the art, GRASP obtained the best solution in 81 out of 84 available instances by requiring half of the computing time. The results reported are supported by the well-known pairwise Wilcoxon statistical test, confirming the superiority of the proposal with respect to the classical and state-of-the-art solution procedures for the BIMP.

Finally, an infodemic case study is analyzed from the influence maximization perspective. Specifically, an instance is built based on 386384 tweets about the American Health Care Act (AHCA). An experiment is performed, showing the superiority of GRASP when comparing it with CombIM in 21 out of 27 available instances. The most influential users are analyzed, showing their relevance in the topic studied, being most of them senators, comedians, writers, or newspapers.

Acknowledgements The authors acknowledge support from the Spanish Ministry of “Ciencia, Innovación MCIN/AEI/10.13039/501100011033/FEDER, UE) under grant ref. PID2021-126605NB-I00 and PID2021-125709OA-C22, the “Comunidad de Madrid” and “Fondos Estructurales” of the European Union with grant reference S2018/TCS-4566.

Data availability Data availability is public at <https://grafo.etsii.urjc.es/BIMP/>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Banerjee S, Jenamani M, Pratihari DK (2019) CombIM: a community-based solution approach for the budgeted influence maximization problem. *Expert Syst Appl* 125:1–13. <https://doi.org/10.1016/j.eswa.2019.01.070>
- Banerjee S, Jenamani M, Pratihari DK (2020) A survey on influence maximization in a social network. *Knowl Inf Syst* 62:3417–3455. <https://doi.org/10.1007/s10115-020-01461-4>
- Barabási A-L, Pósfai M (2016) *Network science*. Cambridge University Press, Cambridge
- Bello-Orgaz G, Hernandez-Castro J, Camacho D (2017) Detecting discussion communities on vaccination in twitter. *Future Gener Comput Syst* 66:125–136. <https://doi.org/10.1016/j.future.2016.06.032>
- Chen W, Wang C, Wang Y (2010) Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining—KDD '10*. ACM Press, Hoboken. <https://doi.org/10.1145/1835804.1835934>
- Chen E, Lerman K, Ferrara E (2020) Tracking social media discourse about the covid-19 pandemic: development of a public coronavirus twitter data set. *JMIR Public Health Surveill* 6(2):e19273. <https://doi.org/10.2196/19273>
- Feo TA, Resende MGC (1989) A probabilistic heuristic for a computationally difficult set covering problem. *Oper Res Lett* 8(2):67–7. [https://doi.org/10.1016/0167-6377\(89\)90002-3](https://doi.org/10.1016/0167-6377(89)90002-3)
- Feo TA, Resende MGC, Smith SH (1994) A greedy randomized adaptive search procedure for maximum independent set. *Oper Res* 42(5):860–878. <https://doi.org/10.1287/opre.42.5.860>
- Gong M, Song C, Duan C, Ma L, Shen B (2016) An efficient memetic algorithm for influence maximization in social networks. *IEEE Comput Intell Mag* 11(3):22–33. <https://doi.org/10.1109/mci.2016.2572538>

- Goyal A, Lu W, Lakshmanan Laks VS (2011) CELF++. In: Proceedings of the 20th international conference companion on World wide web—WWW '11. ACM Press, Hoboken. <https://doi.org/10.1145/1963192.1963217>
- Granovetter M (1978) Threshold models of collective behavior. *Am J Sociol* 83(6):1420–1443
- Guney E, Cakir V, Ozdemir Y, Duzdar I (2015) Budgeted influence maximization in social networks with independent cascade diffusion model. In: Proceedings of the 4th international symposium & 26th national conference on operational research, pp 291–296
- Han S, Zhuang F, He Q, Shi Z (2014) Balanced seed selection for budgeted influence maximization in social networks. In: *Advances in knowledge discovery and data mining*, pp 65–77. Springer International Publishing, Berlin. https://doi.org/10.1007/978-3-319-06608-0_6
- Hansen P, Mladenović N (2006) First vs. best improvement: an empirical study. IV. ALIO/EURO workshop on applied combinatorial optimization. *Discrete Appl Math* 154(5):802–817. <https://doi.org/10.1016/j.dam.2005.05.020>
- Hayes JL, Britt BC, Evans W, Rush SW, Towery NA, Adamson AC (2021) Can social media listening platforms' artificial intelligence be trusted? Examining the accuracy of crimson hexagon's (now brandwatch consumer research's) ai-driven analyses. *J Advert* 50(1):81–91. <https://doi.org/10.1080/00913367.2020.1809576>
- Jos N-V, del Mar G-PM, Villar-Rodríguez G, Martín A, Camacho D (2023) Disinformation and vaccines on social networks: behavior of hoaxes on twitter. *Rev Latina Comun Soc* 81:44–62
- Kempe D, Kleinberg J, Tardos E (2015) Maximizing the spread of influence through a social network. *Theory Comput* 11(1):105–147. <https://doi.org/10.4086/toc.2015.v011a004>
- Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining—KDD '03*. ACM Press, Hoboken. <https://doi.org/10.1145/956750.956769>
- Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution. *ACM Trans Knowl Discov Data* 1(1):2. <https://doi.org/10.1145/1217299.1217301>
- Lozano-Osorio I, Sánchez-Oro J, Duarte A, Cordon Ó (2021) A quick GRASP-based method for influence maximization in social networks. *J Ambient Intell Hum Comput* 1:1. <https://doi.org/10.1007/s12652-021-03510-4>
- Nguyen H, Zheng R (2013) On budgeted influence maximization in social networks. *IEEE J Sel Areas Commun* 31(6):1084–1094. <https://doi.org/10.1109/jsac.2013.130610>
- Rafael M, Anna M-G, Jesús S-O, Abraham D (2018) Tabu search for the dynamic bipartite drawing problem. *Comput. Oper. Res.* 91:1–12
- Resende Mauricio GC, Martí R, Gallego M, Duarte A (2010) GRASP and path relinking for the max-min diversity problem. *Comput Oper Res* 37(3):498–508. <https://doi.org/10.1016/j.cor.2008.05.011>
- Resende Mauricio GC, Ribeiro Celso C (2013) GRASP: Greedy randomized adaptive search procedures. In: *Search methodologies*, pp 287–312. Springer, New York. https://doi.org/10.1007/978-1-4614-6940-7_11
- Reza Z, Ali AM, Huan L (2014) *Social media mining: an introduction*. Cambridge University Press, Cambridge. <https://doi.org/10.1017/CBO9781139088510>
- Richardson M, Agrawal R, Domingos P (2003) Trust management for the semantic web. In: *Lecture notes in computer science*, pp 351–368. Springer, Berlin Heidelberg. https://doi.org/10.1007/978-3-540-39718-2_23
- Stanley W, Katherine F (1994) *Social network analysis*. Cambridge University Press, Cambridge. <https://doi.org/10.1017/cbo9780511815478>
- Tretiakov A, Martín A, Camacho D (2022) Detection of false information in spanish using machine learning techniques. In: Hujun Y, David C, Peter T (eds) *Intelligent data engineering and automated learning—IDEAL 2022*, pp 42–53. Springer International Publishing, Cham
- Wenzheng X, Weifa L, Lin Xiaola Yu, Jeffrey X (2016) Finding top-k influential users in social networks under the structural diversity model. *Information Sciences* 355–356:110–126. <https://doi.org/10.1016/j.ins.2016.03.029>
- Wrubel L, Littman J, Bonnett W, Kerchner D (2020) gwu-libraries/tweetsets: Version 1.1.1. <https://zenodo.org/record/1289426>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.