Knowledge-Based Systems 212 (2021) 106623

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A multi-objective linear threshold influence spread model solved by swarm intelligence-based methods

Rodrigo Olivares, Francisco Muñoz, Fabián Riquelme *

Escuela de Ingeniería Informática, Universidad de Valparaíso, Valparaíso, Chile

ARTICLE INFO

Article history:
Received 24 July 2020
Received in revised form 15 October 2020
Accepted 21 November 2020
Available online 1 December 2020

MSC:
91D30
68W25
90C27
05C21
05C22

Keywords:
Social network
Influence maximization
Influence spread model
Multi-objective optimization
Swarm intelligence

ABSTRACT

The influence maximization problem (IMP) is one of the most important topics in social network analysis. It consists of finding the smallest seed of users that maximizes the influence spread in a social network. The main influence spread models are the linear threshold model (LT-model) and the independent cascade model (IC-model). These models have mainly been treated by using the single-objective paradigm which covers just one perspective: maximize the influence spread starting by given seed size, or minimize the seed set to reach a given number of influenced nodes. Sometimes, this minimization problem has been called the least cost influence problem (LCI). In this work, we propose a new optimization model for both perspectives under conflict, through the LT-model, by applying a binary multi-objective approach. Swarm intelligence methods are implemented to solve our proposal on real networks. Results are promising and suggest that the new multi-objective solution proposed can be properly solved in harder instances.

1. Introduction

A social network can be represented as a graph, whose nodes are the actors of the network, and its edges the interpersonal ties among the actors [1]. One of the main problems in social network analysis is to study the influence spread, starting from a given seed of individuals or actors. The underlying intuitive idea is trying to maximize the influence spread by minimizing the available resources. This is a well-known phenomenon, present in a high diversity of disciplines and applications, such as viral marketing [2], information propagation [3], expertise recommendation [4], community systems [5], social customer relationship management [6], percolation theory [7], cooperative game theory [8], search strategies [9], collective decision-making [10,11], networks centrality [12,13], among others.

The influence spread phenomenon can be studied on weighted, labeled digraphs, namely influence graphs. In influence graphs, the edge weights represent the influence power exerted by one actor over another, and the node labels quantify the resistance of each actor to be influenced by other nodes pointing to it [8]. There are two well-known models to represent the influence spread phenomenon through social networks: The linear threshold model (LT-model) [14,15], based on some ideas of collective behavior [16,17], and the independent cascade model (IC-model) [14,15], based on marketing ideas [14,18].

Regardless of the influence spread model considered, the influence maximization problem (IMP) remains the same: given an influence graph, find the minimum seed of nodes (initial activation) that maximize the influence spread through the network. Seen like this, IMP seems to be a min–max optimization problem, formed by two objective functions. However, so far the problem has only been addressed from a single-objective perspective: usually, by maximizing the influence spread, given a seed size [15], and sometimes, by minimizing the initial seed set, given a fixed number of nodes to be influenced [19]. As a minimization problem, it is also known as the least cost influence (LCI) problem [20].

As a decision problem, it is known that both problems are NP-hard [15,19] because the number of solutions can increase exponentially when a new node is included in the network. Therefore, several algorithm techniques have been modified to try to solve them as best as possible. Despite both the maximization and the minimization approaches are properly applied, we consider they are not exclusive. In this work, we model both the IMP and LCI problems under a binary multi-objective optimization paradigm where each node will be related with a decision variable. We try to achieve the maximum influence spread considering a minimum seed set.

* Corresponding author.
E-mail address: fabian.riquelme@uv.cl (F. Riquelme).

https://doi.org/10.1016/j.knosys.2020.106623

We solve this problem using two well-known paradigms: an exact algorithm and swarm intelligence methods. The first one was performed to test smaller instances, and thus know their optimal solutions. Swarm intelligence methods allow us to achieve near-optimal solutions in reduced solving times. They are usually adopted due to their ability to synchronously search for robust solutions and perform better global exploration of the search space [21,22]. Swarm intelligence-based techniques are smart procedures that try to solve various kinds of problems, mainly optimization problems, by mimicking the collective behavior of certain individuals [23]. These bio-inspired algorithms operate typically with a set of agents, called population, interacting locally with each other and their environment. Nature is the main inspiration source for swarm algorithms, especially biological systems. Some highest recognized swarm methods are ant colony systems, echolocation featuring, bird flocking, animal hunting, among several others [24,25]. We implement three familiar swarm intelligence algorithms: particle swarm optimization (PSO), bat algorithm (BA), and black hole optimization (BHO). PSO is selected because it was one of the first swarm intelligence algorithms used to solve the IMP [26]; BA, because it has been recently applied in the same kind of problems [27], and BHO, as an alternative that has not yet been used to solve these problems. Moreover, these techniques have a common collective behavior, so they are quite similar to each other. Initially, these three algorithms are tested on the smallest network (the one used by the exact algorithm), to adjust the parameters and thus ensure reaching the optimal solution, at least in this case. We employ this bio-inspired approach, given its undeniable performance when treating optimization problems. Results show that by balancing the maximum influence spread with the minimum seed set, attractive results can be reached in reduced computation time, specially for PSO.

The manuscript continues as follows. Section 2 presents an extensive bibliographic search for related work in the field. The results of this search justify the novelty of the multi-objective proposal of this work. Section 3 describes the IMP under the LT-model, with no restrictions on seed size, to be addressed in Section 4 using a new multi-objective approach. Section 5 describes the experimental setup, while Section 6 presents and discusses the main results obtained. Finally, conclusions and future work are included in Section 7.

2. Related work

The influence maximization problem (IMP) was first defined as a discrete optimization problem in a renowned work of 2003 [15]. The problem was addressed using the two most widely used influence spread models to date: the independent cascade model (IC-model) and the linear threshold model (LT-model), formally defined two years earlier in [14]. From the beginning, IMP was already defined as a maximization problem, that is, using a single-objective approach. Since then, numerous proposals have been developed to try to solve the problem as best as possible, in terms of accuracy, temporal complexity, spatial complexity, or scalability [28].

Despite the large amount of existing work, there are just a few surveys related to the algorithms and techniques used to solve the problem [29,28,30]. Table 1 presents what, as far as we know, is the most exhaustive collection of works related to this topic. The minimization problem, also known as the Least cost influence (LCI) problem [20], was formally defined in 2009 [31], although the first experimental study was published two years later, using the IC-model [19]. Table 1 shows that the minimization problem has been much less studied than the original maximization problem. Indeed, of more than 60 works listed, only six solve the LCI problem, while all the rest focus on IMP. Note that none of them addresses both problems together. Therefore, as far as we know, the problem has never been studied using a multi-objective approach. Regarding the influence spread model, note that only 9 works use both the IC-model and LT-model, 39 use only the IC-model, and 13 only the LT-model. Although these are the most widely used spread models, there are also variations [32] or generalizations [33,28] of them, as well as other less known models [34–36].

Regarding the types of algorithms used, greedy algorithms are used in 32 cases, heuristics-based algorithms in 15 cases, and metaheuristics in only 11 cases. The first metaheuristic used was the simulated annealing (SA) in 2011 [37]. Years later, some results were introduced with genetic algorithms [38–40], while the discrete particle swarm optimization (discrete PSO) method was implemented for the first time in 2016 [26]. Other bio-inspired algorithms such as artificial bee colony (ABC) [36,41], discrete bat (DBA) [27], gray wolf [42], whale optimization [42], and more recently, discrete shuffled frog-leaping (DSFLA) [43] have been applied. All of these swarm intelligence approaches have used the IC-model to solve IMP, except for the case of ABC, which focuses on a different influence spread model adjusted to the bees' behavior [36]. Finally, evolutionary algorithms have also been used recently [44–46]. The works of Weskida et al. are the only ones to date that have used the LT-model to solve both IMP [44] and the LCI problem [46] with metaheuristics. Furthermore, in the recent work of Sheikhahmadi and Zareie [41], the authors, for the first time, use a multi-objective approach to solve IMP. However, they use it for a different purpose than ours, and they remain in solving the maximization problem only. As can be seen, since 2016, metaheuristics are the most researched method to solve IMP. In this research, we will expand the use of bio-inspired metaheuristics using the LT-model.

The IMP and the LCI are defined on a static social network or a "shot" of a dynamic network. However, both problems have also been studied on multiplex social networks [20,94]; networks changing over time (for example, through Markov chains [95]); networks with partial knowledge, whose social structure is initially unknown [96]; signed networks [97], among other variations.

Besides, several generalizations and variations of the IMP have emerged from its formal definition [98]. For instance, the IMP in competitive social networks, i.e., where different influence spread strategies are competing to each other, has been defined [99,100]; the resource allocation problem on modular social networks [101]; the IMP on evolving or dynamic social networks [30], also known as influential node tracking (INT) problem [102–104] or incremental influence maximization problem [105,106]; the budgeted influence maximization (BIM) problem, which considers an additional constraint of limited budget [107,108]; the location-aware influence maximization problem, where the actors' geographical location matters [109]; the multiple influence maximization (MIM) problem, where different types of activations are allowed at the same time [110]; the reverse influence maximization (RIM) problem [111], among others. Again, in most cases, the IC-model is used instead of the LT-model.

3. Preliminaries

In this section, we introduce the influence spread process within a social network. In particular, we formally define the linear threshold model, one of the two main influence spread models, together with the independent cascade model. After that, we explain the influence maximization problem, whose modeling as a multi-objective optimization problem is developed in Section 4. We use notation from [8].

Table 1
Different solutions of the IMP and LCI problem (see Refs. [47–93]).

#	Ref.	Year	Problem solved	Algorithm type	Metaheuristic used	Spread model
1	[15]	2003	IMP	greedy	–	IC, LT
2	[47]	2006	IMP	greedy	–	IC
3	[48]	2007	IMP	greedy	–	IC, LT
4	[49]	2007	IMP	greedy	–	IC, LT
5	[50]	2009	IMP	greedy, heuristic	–	IC
6	[51]	2009	IMP	greedy	–	LT
7	[52]	2010	IMP	greedy, heuristic	–	LT
8	[53]	2010	IMP	heuristic	–	IC
9	[54]	2010	IMP	greedy	–	IC
10	[55]	2011	IMP	heuristic	–	IC
11	[19]	2011	LCI	greedy	–	IC
12	[56]	2011	IMP	heuristic	–	LT
13	[57]	2011	IMP	greedy	–	IC
14	[37]	2011	IMP	metaheuristic	simulated annealing (SA)	IC
15	[58]	2011	IMP	heuristic	–	LT
16	[32]	2012	IMP	greedy	–	LT variation
17	[59]	2012	LCI	greedy	–	IC
18	[60]	2012	IMP	heuristic	–	IC
19	[61]	2012	IMP	approximation	–	IC
20	[62]	2012	IMP	greedy	–	IC, LT
21	[63]	2012	LCI	heuristic	–	LT
22	[64]	2012	IMP	greedy	–	IC
23	[65]	2013	IMP	greedy	–	IC
24	[66]	2013	LCI	approximation	–	LT
25	[67]	2013	IMP	approximation	–	IC
26	[5]	2013	IMP	clustering	–	IC
27	[68]	2014	IMP	approximation, greedy	–	IC
28	[34]	2014	IMP	community-based	–	other
29	[69]	2014	IMP	greedy, heuristic	–	IC
30	[70]	2014	IMP	greedy	–	IC
31	[71]	2014	IMP	greedy	–	IC
32	[35]	2014	IMP	greedy	–	other
33	[72]	2014	IMP	greedy	–	LT
34	[73]	2014	IMP	greedy	–	IC
35	[74]	2014	IMP	heuristic	–	IC
36	[75]	2014	IMP	approximation, greedy	–	IC, LT
37	[76]	2014	IMP	greedy	–	IC
38	[77]	2014	LCI	greedy	–	IC
39	[78]	2015	IMP	greedy	–	IC
40	[79]	2015	IMP	greedy, heuristic	–	IC
41	[80]	2015	IMP	other	–	LT
42	[81]	2015	IMP	community-based, greedy	–	IC
43	[82]	2015	IMP	greedy	–	IC, LT
44	[38]	2015	IMP	greedy, metaheuristic	genetic	IC
45	[83]	2015	IMP	greedy	–	LT
46	[39]	2016	IMP	metaheuristic	genetic	IC
47	[84]	2016	IMP	clustering, memetic	–	IC
48	[26]	2016	IMP	metaheuristic	discrete PSO (DPSO)	IC
49	[85]	2016	IMP	community-based	–	LT
50	[86]	2016	IMP	greedy	–	IC
51	[87]	2016	IMP	approximation	–	IC, LT
52	[36]	2016	IMP	metaheuristic	artificial bee colony (ABC)	other
53	[88]	2016	IMP	heuristic	–	IC
54	[89]	2016	IMP	heuristic	–	other
55	[44]	2016	IMP	metaheuristic	evolutionary	LT
56	[90]	2017	IMP	random walk	–	IC
57	[91]	2017	IMP	community-based	–	IC
58	[92]	2017	IMP	heuristic	–	IC
59	[40]	2017	IMP	metaheuristic	genetic	LT
60	[45]	2018	IMP	metaheuristic	evolutionary	IC
61	[93]	2018	IMP	heuristic	–	IC, LT
62	[42]	2018	IMP	metaheuristic	gray wolf, whale optimization	IC
63	[27]	2018	IMP	metaheuristic	discrete bat (DBA)	IC
64	[28]	2019	IMP	community-based	–	IC, LT (general)
65	[46]	2019	LCI	metaheuristic	evolutionary	LT
66	[43]	2020	IMP	metaheuristic	discrete shuffled frog-leaping (DSFLA)	IC
67	[41]	2020	IMP	metaheuristic	artificial bee colony (ABC)	other

For what follows, we denote a *weighted graph* as a pair (G, w), where $G = (V, E)$ is a directed graph or digraph with node set $V(G)$ and edge set $E(G)$, and $w : E(G) \to \mathbb{N}$ is a *weight function* that assigns a weight to every edge. Thus, we can represent a social network as a weighted graph (G, w), where $V(G)$ is the set of actors, $E(G)$ is the set of interpersonal ties among the actors within the network, and a weight $w(i, j)$ represents the influence power exerted by an actor i over j. Let be $n = |V|$ and $m = |E|$.

For more clarity, everything shown in this section is represented in set notation. However, in Section 4, we shall use also an equivalent vector notation. As a matter of fact, there exists a direct relationship between sets $X \subseteq \{1, \ldots, n\}$ and vectors

$x = (x_1, \ldots, x_n) \in \{0,1\}^n$. For doing so, we use the notation $X(x) = \{0 \le i \le n \mid x_i = 1\}$, and $x(X) = (x_1, \ldots, x_n)$ with $x_i = 1$ if $i \in X$, and $x_i = 0$ otherwise [11]. Also, we denote $w(i,j) = w_{ij}$ for any edge $(i,j) \in E(G)$.

3.1. Influence spread model

In social network analysis, besides the weighted function that represents the strength of interpersonal ties within the social network, we can associate labels to the nodes in order to represent the resistance of each actor to be influenced by others. This leads us to a more general representation of social networks through the use of influence graphs [8].

An *influence graph* is a tuple (G, w, f) where (G, w) is a weighted graph, and $f : V \to \mathbb{N}$ is a *labeling function* that assigns a label to each node. Note that undirected graphs can be seen as symmetric digraphs, so they are also included in this background.

Given an influence graph (G, w, f) and a seed or initial activation set $X \subseteq V$, the *spread of influence* of X is the set $F(X) \subseteq V$ formed by the nodes activated through an iterative process as follows. Let us $F_t(X)$ denote the set of nodes activated at step t. Initially, at step 0, only the seed is active, so $F_0(X) = X$. Then, at step $i > 0$, the set of nodes activated is formed by all the nodes of $F_{i-1}(X)$, plus some additional amount of nodes that depends on the influence spread model considered. For the *linear threshold model* (*LT-model*), we add all the nodes whose labels are less or equal than the total weight of the edges pointing to them from nodes in $F_{i-1}(X)$, i.e.,

$$F_i(X) = F_{i-1}(X) \cup \left\{ v \in V \,\middle|\, \sum_{\{u \in F_{i-1}(X) \mid (u,v) \in E\}} w(u,v) \ge f(v) \right\} \tag{1}$$

This process stops when no additional activation occurs. The final set of activated nodes is $F(X) = F_t(X)$, where $t = \min\{i \in \mathbb{N} \mid F_i(X) = F_{i+1}(X)\} \le n$.

Fig. 1 illustrates an example of influence spread process in an influence graph. The spread of influence $F(X)$ starts from the seed $X = \{a\}$. In the first step is obtained $F_1(X) = \{a, c\}$ and in the second step (the last one), $F_2(X) = \{a, c, d\}$.

3.2. Influence maximization problem

Let (G, w, f) be a social network represented as an influence graph. The *influence maximization problem* (IMP) aims to select a desirable seed $X \subseteq V(G)$ such that the spread of influence $F(X)$ is maximum under a given influence spread model. A seed X should meet two conditions to be more desirable:

- Be formed by a low number of actors, i.e., have a low $|X|$ value.
- Be minimal, i.e., if any actor i is removed from the seed, then $|F(X \setminus \{i\})| < |F(X)|$.

The maximum influence spread is reached when $|F(X)| = n$. For the influence graph of Fig. 1, the most desirable seed is $X = \{b\}$. However, note that for some cases, there could be several desirable seeds with different numbers of actors.

Let X^* be a desirable seed, the influence maximization problem can be stated as follows [43]:

$$X^* = \operatorname{argmax}_{X \subseteq V,\ |X| = k} F(X) \tag{2}$$

where k is given. This optimization problem is NP-hard [15]. In the present work, k should also be minimized.

4. Developed solution

In this section, we present a multi-objective model for the influence maximization problem (IMP) and then, we describe three widely known swarm intelligent techniques that treat it.

4.1. Multi-objective model

In optimization, several problems, such as multiple criteria decision-making, task team formation, web-based virtual collaboration environments, network resource allocation, among others, have been tacked by using the multi-objective paradigm [112]. In countless works, optimization problems are formulated through either minimization or maximization functions. Multi-objective optimization can be modeled as the problem of finding a vector of decision variables that satisfies constraints and optimizes a vector of objective functions whose elements represent the objective functions [112]. This vector is mainly composed of mathematical formulas which are typically in conflict when none of the objective functions can improve their value without degrading some of the other objective values. Therefore, there is no single solution that simultaneously optimizes the vector of objective functions. Instead, there is a set of efficient solutions formed by non-dominated solutions that are located in a part of the target space. This set of efficient solutions is called the Pareto-optimal front.

Based on the multi-objective paradigm, we state a model to treat the multi-objective problem. Firstly, we must find the vector of efficient solutions $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which satisfy the p inequality and the q equality constraints:

$$g_i(\bar{x}) \gtreqless 0, \quad h_j(\bar{x}) = 0, \quad \forall i = \{1, 2, \ldots, p\} \wedge \forall j = \{1, 2, \ldots, q\}$$

where $\gtreqless$ means $\le$ or $\ge$, and then, optimize the vector of functions:

$$f(\bar{x}) = [f_1(\bar{x}), f_s(\bar{x}), \ldots, f_k(\bar{x})]^T$$

where $\bar{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables.

By considering the multi-objective paradigm, we can state a new model for the influence maximization problem, that also integrates the LCI problem:

$$\begin{aligned} f(\bar{x}) = &\left\{ \texttt{maximize } |F(\bar{x})| \; ; \; \texttt{minimize } \sum_{i=1}^{n} x_i \right\} \\ &\texttt{s.t.} \quad \sum_{i=1}^{n} x_i \le |F(x)| \\ &x_i \in \{0,1\}, \quad \forall i \in \{1, \ldots, n\} \end{aligned} \tag{3}$$

In this case study, the vector of objectives functions is composed of two elements: maximize the spread of influence and minimize the seed. Clearly, these functions conflict, since decreasing the seed or initial activation set may also decrease the influence spread. Analogously, increasing the influence spread may require increasing seed size. To solve this instance, we transform the minimization problem into a maximization problem by using its negative value. Thus, the vector of objective functions becomes as follows:

$$\texttt{maximize}\, f(x) = \left\{ |F(x)| \; ; \; -\sum_{i=1}^{n} x_i \right\} \tag{4}$$

To solve this instance, we employ a scalarizing approach on the vector of objective functions [113]. The scalarizing is an *a priori* method for multi-objective optimization problem resolution. It works by balancing the single-objective functions to find ideal and anti-ideal points that are optimal for the multi-objective optimization problem (efficient solutions) [114].

4.2. Swarm intelligence methods

The swarm intelligence methods are a type of metaheuristic algorithms that work under a common collective behavior. Generally, individuals interact with each other in order to update and

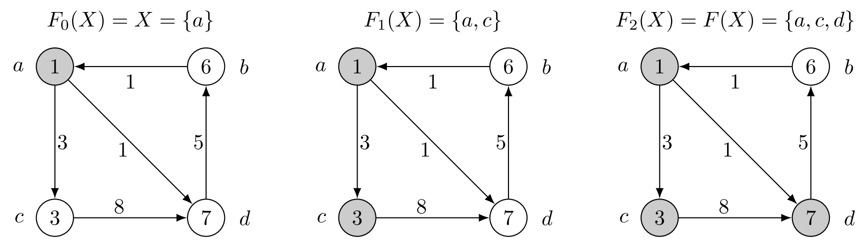


Fig. 1. Influence spread process within an influence graph.

improve themselves. For this work, we chose three similar swarm intelligence procedures to tackle the proposed multi-objective approach: Particle swarm optimization, bat algorithm, and black hole optimization. These techniques have had a long journey in bio-inspired optimization and have proven to be efficient in addressing large-scale instances of various problems.

Particle swarm optimization (PSO) and bat algorithm (BA) are two of the most popular swarm intelligence methods. The first one is inspired by the behavior of birds flocking or fish schooling when they move from one place to another. The second is based on the echolocation phenomenon that is present in the species of microbats, which allows them to avoid obstacles while flying and to locate food or shelter. Both techniques describe a similar behavior handling velocity changes and position updates. Another optimization technique is the black hole algorithm (BHO), which is inspired by the absorption feature present in the homonymous phenomenon [115]. A black hole is a region of space that has so much mass concentrated in it that there is no way for a nearby object to escape its gravitational pull. Anything falling into a black hole, including light, cannot escape.

4.2.1. Particle swarm optimization

In particle swarm optimization, each bird or fish represents a particle with two components: position and velocity. A set of particles (the candidate solutions) forms the swarm that evolves during several iterations giving rise to a powerful optimization method [116,117]. The method operates altering velocity through the search space, and then updating its position according to its own experience and that of neighboring particles.

PSO can be identified as a smart system with two phases. During the initial phase, when the algorithm reaches large velocities, the current solutions focus more on diversification. Next, as velocities slow toward zero, the current solutions will focus more on intensification. This second phase will occur around the positions memorized as pBest, so the goal of the initial phase is to find pBest positions that are members of the fittest attraction basins.

The standard particle swarm optimization is governed by two vectors, the velocity $V_i = \langle v_i^1, v_i^2, \ldots, v_i^D \rangle$ and the position $X_i = \langle x_i^1, x_i^2, \ldots, x_i^D \rangle$. First, the particles are randomly positioned in a D-dimensional heuristic space with random velocity values. During the evolution process, each particle updates its velocity (Eq. (5)) and position (Eq. (6)):

$$v_i^d = \omega v_i^d + c_1 \phi_1^d (pBest_i^d - x_i^d) + c_2 \phi_2^d (gBest^d - x_i^d) \tag{5}$$

$$x_i^d = x_i^d + v_i^d \tag{6}$$

where $d = \{1, 2, \ldots, D\}$, the positive constants ω, c_1, and c_1 are acceleration coefficients, ϕ_1 and ϕ_2 are two uniformly distributed random numbers in the range [0, 1], $pBest_i$ is the previous best position of ith particle, and $gBest$ is the global best position found by all particles during the resolution process.

4.2.2. Bat algorithm

This algorithm uses the concept of a *virtual bat*, which is employed to describe an artificial bat of arbitrary species. It follows three rules [118]:

(1) All virtual bats are assumed to use echolocation to determine distances and being able to distinguish between food, prey, and background obstacles.
(2) A virtual bat b_i flies at a position x_i with a velocity v_i. The pulses of sound emitted have the following features: a frequency f_i, a loudness A_0, and a rate of pulse emission $r \in [0, 1]$.
    (a) Both the velocity and the position are vectors $V_i = \langle v_i^1, v_i^2, \ldots, v_i^D \rangle$ and $X_i = \langle x_i^1, x_i^2, \ldots, x_i^D \rangle$, respectively, in a D-dimensional heuristic space for an optimization problem.
    (b) Simplifying, the frequency varies in a range $[f_{min}, f_{max}]$ and is chosen to be comparable with the size of the interest domain.
    (c) All sound features can be automatically adjusted depending on the proximity of the target.
(3) Although the loudness can vary in many ways, it is assumed that it ranges from a large (positive) value A_0 to a minimum constant value A_{min}.

The algorithm begins with an initial population of virtual bats. In each iteration, the best solution is chosen according to its performance and it is called the global solution. For a virtual bat solving an optimization problem, the frequencies (Eq. (7)), velocities (Eq. (8)), and positions (Eq. (9)) of its behavior rules must be defined.

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{7}$$

$$v_i^d = (x_{best}^d - x_i^d) f_i \tag{8}$$

$$x_i^d = x_i^d + v_i^d \tag{9}$$

where $d = \{1, 2, \ldots, D\}$, β is a vector of uniformly distributed random values in the range [0, 1], f_{min} is set to have a small value and f_{max} varies according to the *max* variance allowed in each time step. Finally, x_{best} represents the global best position found by all virtual bat during the resolution process.

Bat optimization is governed by two general phases: diversification and intensification. The diversification phase is led by the random walk trajectory to select a solution among the current best solutions. The new solution is generated on the bases of current loudness A_i of the virtual bat and maximum allowed variance $max(var)$ during a time step as:

$$x_{new}^d = x_{old}^d + \epsilon A_i max(var) \tag{10}$$

where ϵ is a random value in $[-1, 1]$.

Finally, the intensification phase is governed by the variation between loudness and pulse emission. It appears when a bat found its prey. Here, the loudness decreases, and the rate of pulse emission increases, thus other bats will move toward the optimal solution:

$$A_i = \alpha A_i, \quad r_i = r_i^{time=0}(1 - e^{-\gamma(time=t)}) \tag{11}$$

where α and γ are ad-hoc constants to control the intensification phase. For $0 < \alpha < 1$ and $\gamma > 0$, we get $A_i \to 0$, $r_i \to r_i^{(time=0)}$, $t \to 0$.

4.2.3. Black hole optimization

The black hole optimization starts with an randomly generated initial population of potential solutions or positions $X_i = \langle x_i^1, x_i^2, \ldots, x_i^D \rangle$, in an D-dimensional heuristic space for an optimization problem. In this case, at each iteration, the best candidate is selected to be the *black hole*, which then starts pulling other candidates around it, called *stars*. If a star gets too close to the black hole, it will be swallowed, and it is gone forever. In such a case, a new star (potential solution) is randomly generated and placed in the search space and starts a new search.

The black hole can absorb the stars that surround it. After initializing the black hole and stars, the black hole begins by absorbing the stars around it, and all the stars start moving toward the black hole. The absorption of stars by the black hole is formulated as follows (Eqs. (12) and (13)):

$$v_i^d = r(x_{bh}^d - x_i^d) \tag{12}$$

$$x_i^d = x_i^d + v_i^d \tag{13}$$

where $d = \{1, 2, \ldots, D\}$, r is a vector of uniformly distributed random values in the range [0, 1], and x_{bh} represents the global best location (or black hole), found by all stars during the resolution process.

The event horizon plays an important role in algorithm and controls the balance between diversification and intensification, that is, controlling the global and local search. Every star that crosses the event horizon of the black hole will be absorbed by the black hole. The radius of the event horizon in the black hole algorithm is calculated as follows (Eq. (14)):

$$R = \frac{f(x_{bh})}{\sum_{i \in I} f(x_i)} \tag{14}$$

where I represents the numbers of stars, $f(x_{BH})$ is the fitness value of the black hole, and $f(x_i)$ is the fitness value of the ith star. When the distance between a ith star and the black hole ($diff_i$) is less than the event horizon R, then the star collapses into the black hole. The distance between a star and black hole is the Euclidean distance (Eq. (15)) calculated as follows:

$$diff_i = \sqrt{(x_{BH}^1 - x_i^1)^2 + (x_{BH}^2 - x_i^2)^2 + \cdots + (x_{BH}^d - x_i^d)^2} \tag{15}$$

where x_{BH}^k and x_i^k, with $k = \{1, 2, \ldots, D\}$, are decision variables for an optimization problem. Every time, a star is absorbed by the black hole and a new star is automatically generated in a random manner. After all stars have been moved, the black hole algorithm iterates. The algorithm continues with the black hole in the new location and the stars start moving toward this new location.

4.2.4. Multi-objective swarm intelligence methods

Techniques inspired on collective behavior, such as swarm intelligence methods, typically consider a population of virtual agents interacting locally with one another and with their environment. The agents or individuals follow a set of rules that allow them to work based on a decentralized control structure and lay the groundwork about how individuals should behave. Based on this, we designed a common architecture for swarm intelligence methods. This procedure structure can be seen in Algorithm 1. This architecture allowed us to implement the problem-solver.

The general structure of swarm intelligence methods begins loading the problem data to define parameter values of the instance. Next, Lines (5–9) describe a sub-procedure to compute the initial population randomly. Here, the position and the velocity take a binary random value during the loop statement is processed. Later, the multi-objective evaluation is calculated for each agent. After this initial sub-procedure, the main loop structure is executed. This statement is split into two parts. Lines (14–22) detail the diversification and intensification phases. These processes modify the position value using mainly mathematical distributions that are controlled by the originally proposed methods. For example, in PSO, both the diversification and the intensification phases directly work on the velocity update action, unlike the bat algorithm or the black hole optimizer. Both the bat algorithm and the black hole optimization include formal methods to explore and to exploit the search space. In the bat algorithm technique, a random work distribution is proposed in Eq. (10) to control the solution diversity. A self-adjust of the pulse rate and loudness is used to increase/decrease the promising zone (see Eq. (11))). Finally, in the black hole algorithm, the event horizon phenomenon manages such diversification and intensification states. The intensification process runs the closer you get to the black hole. However, if a star (solution) crosses the event horizon, it is absorbed and then disappears from the constellation, spontaneously creating a new star in a random place (see Eq. (14)).

At the end of the sub-procedure, the multi-objective evaluation is computed again. The second part updates the position and velocity through equations modeled on the inspiration of the phenomenon that governs the swarm intelligence method. This sub-process is composed of a set of instructions depicted in Lines (25–26), that evolve agents toward more promising positions. Updated solutions will probably change their domain. To avoid this inconsistency, a binarization step is applied between Lines (29–33) by using a standard Sigmoid function compared to a uniform random value between 0 and 1 [119,120]. S-Shape and V-Shape functions are mathematical methods widely employed to linearly convert real numbers into binary values. For example, in [121], a recent binary swarm intelligence method is proposed for a feature selection.

5. Experimental setup

The experiments were carried out on three networks of different sizes, on the order of 10, 10^3, and 10^4 nodes. These networks come from datasets available online and had to be pre-processed to convert them into appropriate influence graphs for the application of the LT-model. The smaller network was also used to configure the algorithm parameters. The different networks used are detailed below, as well as the fundamental aspects related to the implementation of metaheuristics.

5.1. Datasets

The three network datasets together with their number of nodes (n) and edges (m) are summarized in Table 2. According to Wanjing et al. [122], the Football network (G_1) contains "the relationship data of American high school football summer class A regular season in 2000". Each node represents a different team, and the weighted edges are relationships between teams. We interpret an edge (a, b) as the team a exerting a certain influence on the team b, with a strength $w(a, b)$.

The Bitcoin Alpha network (G_2) represents trust relationships in bitcoin exchanges through the BTC Alpha platform. In these kind of platforms, users are anonymous and do not know the identity of the person they are dealing with. Therefore, to prevent

Algorithm 1 Common structure for a swarm intelligence method.

Require: problem input data, *popSize*, T
Ensure: a set of efficient solutions that resolve the multi-objective IMP.
loadProblemData()
{the D value defines the number of nodes of the IMP.}
objective functions $f_1(x)$ and $f_2(x)$, $x = \langle x^1, \dots, x^D \rangle$.
{produce the first generation of *popSize* agents}
for all agent a_i, $(\forall\ i = \{1, \dots, popSize\})$ **do**
  for all dimension d, $(\forall\ d = \{1, \dots, D\})$ **do**
    position $x_i^d \leftarrow Random\{0, 1\}$
    velocity $v_i^d \leftarrow Random\{0, 1\}$
  end for
  maximize $\{f_1(x_i); -f_2(x_i)\}$
end for
{produce T-generations of *popSize* agents}
while $t < T$ **do**
  for all agent a_i, $(\forall\ i = \{1, \dots, popSize\})$ **do**
    if the diversification process is invoked **then**
      position x_i^d is modified to explore more promising regions
    end if
    if the intensification process is invoked **then**
      a new position x_i^d is selected among the best solutions
    end if
    maximize $\{f_1(x_i); -f_2(x_i)\}$
  end for
  for all agent a_i, $(\forall\ i = \{1, \dots, popSize\})$ **do**
    for all dimension d, $(\forall\ d = \{1, \dots, D\})$ **do**
      {generate new solutions}
      velocity v_i^d is updated according to the distance between the best and current velocity.
      position x_i^d is updated as fallows: $x_i^d = x_i^d + v_i^d$
      {then, position value must be brought to a binary domain}
      if $\dfrac{1}{1 + e^{-x_i^d}} > Random[0, 1)$ **then**
        $x_i^d \leftarrow 1$
      else
        $x_i^d \leftarrow 0$
      end if
    end for
    maximize $\{f_1(x_i); -f_2(x_i)\}$
  end for
end while
return post-process results and visualization

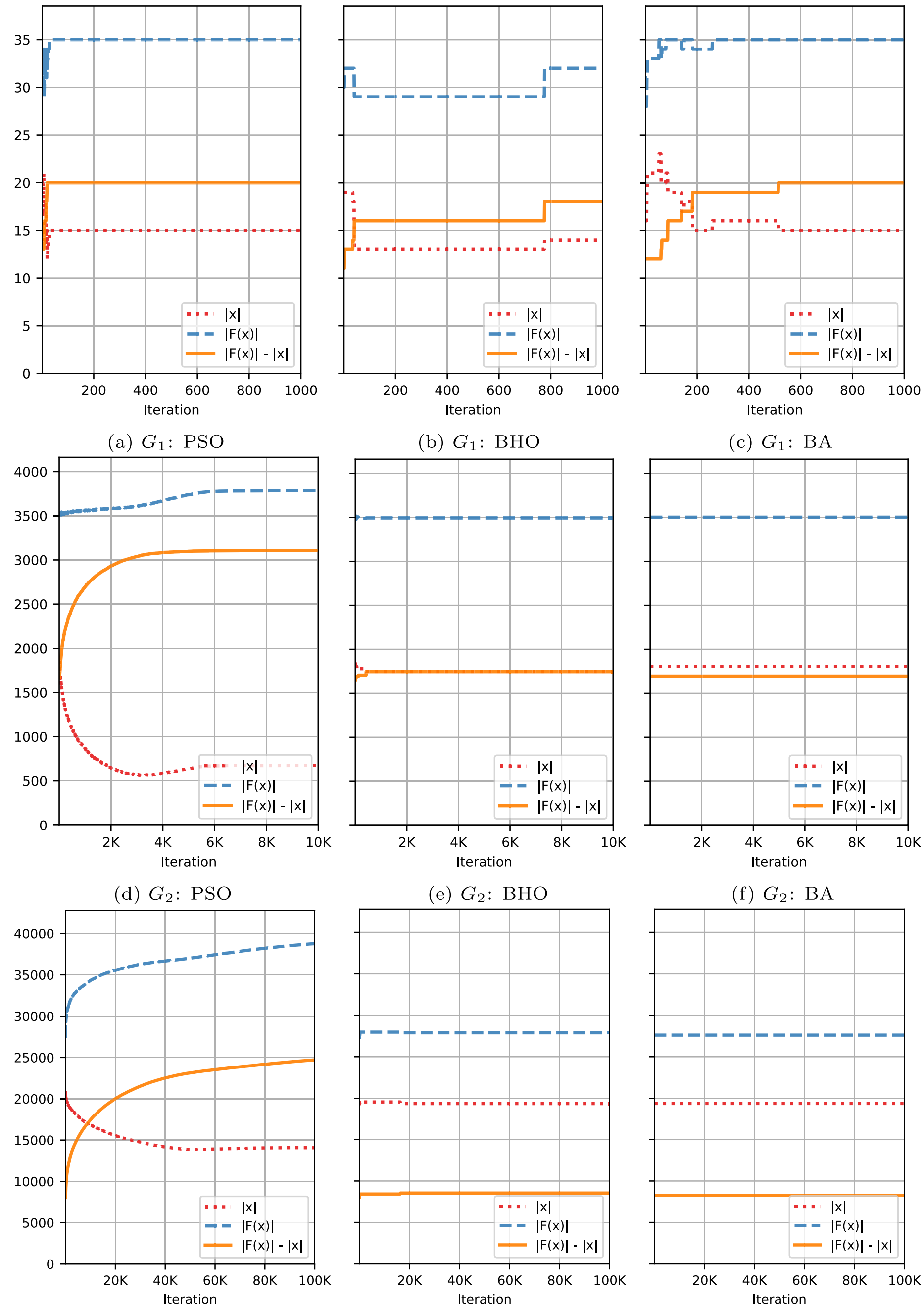
Table 2
Network datasets used for experiments.

ID	Name	Ref.	n	m	URL
G_1	Football	[123]	35	118	http://networkrepository.com/football.php
G_2	Bitcoin Alpha	[124,125]	3783	24186	https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html
G_3	Higgs Twitter	[126]	38918	32523	https://snap.stanford.edu/data/higgs-twitter.html

risky or fraudulent transactions, users must evaluate their transactions, to record the reputation of each investor. The original network works as follows. An edge (a, b) represents the bitcoin transactions received by user a from user b. The weight of this edge, $w(a, b)$, represents the evaluation of user a to user b for these transactions, which corresponds to an integer value between -10 (total distrust) and $+10$ (total trust). In this case, to represent an influence graph, we change the evaluation between -10 and 10 by values between 1 and 21, and invert the direction of all edges, so now $w(a, b)$ represents the "influence" power exerted by user a on user b. In other words, if a evaluates b very well (very poorly), this means that b exerts a high (low) influence on a. This makes sense, if we think that a person will prefer to continue trading with those who give them the most confidence. Note that the `time` parameter of the original dataset is not taken into account, since IMP does not consider time windows.

The Higgs Twitter network (G_3) represents reply relationships among Twitter users collected between 1st and 7th July 2012, during and after the announcement of the discovery of a new particle with features consistent with the Higgs boson on 4th July 2012. This dataset was updated of March 31, 2015. In this case, an edge (a, b) represents that actor a replied actor b a number of $w(a, b)$ times. Analogously to the previous case, here we also invert the direction of the edges, so now $w(a, b)$ represents the influence power exerted by a on b. Again, the timestamps are not considered.

Finally, in order to apply the LT-model, we need to assign labels to the nodes of all networks. These labels represent the



(a) G_1: PSO (b) G_1: BHO (c) G_1: BA

(d) G_2: PSO (e) G_2: BHO (f) G_2: BA

(g) G_3: PSO (h) G_3: BHO (i) G_3: BA

Fig. 2. Convergence graphs (iterations vs. nodes) for each network and metaheuristics type.

resistance of each node to be influenced. To do this, we use the majority criterion [127,12,13], which assigns each node a label that exceeds half the influence power exerted on it, that is, given the influence graph $(G = (V, E), w, f)$, for each actor $i \in V$, $f(i) = \lfloor \bar{w}(i)/2 \rfloor + 1$, where $\bar{w}(i) = \sum\{w(j, i) \mid (j, i) \in E\}$.

5.2. Implementation

Firstly, an exact and brute force algorithm was implemented, which allows all possible seed combinations to be generated one by one, to compute their influence spread according to Eq. (1). If for any seed X a maximum spread is found, that is, such that $|F(X)| = n$, then seeds X' with $|X'| > |X|$ are no longer generated, since we know that $F(X') = F(X)$. For all the algorithms in this work we use the Python programming language. This method was run for the G_1-Football network (see Table 2). It took 5.25 days of computing, with 8 processes running in parallel on an i7 8700 workstation with 32 GB RAM. The best solution obtained corresponds to a seed X with $|X| = 15$, for which was obtained $|F_3(X)| = |F(X)| = 35$. No seed with $|X| < 15$ manages to activate all the network nodes. Furthermore, there are 20 solutions with $|X| = 16$ and 190 with $|X| = 17$ for which $|F(X)| = 35$ is obtained. Nine of the seeds with $|X| = 17$ manage to influence the 35 network nodes in just two steps, while all the rest do so in three. All solutions were saved in a human readable file to identify the efficient solutions, using 2.5 TB of memory. Running this algorithm for the G_2 and G_3 networks, even without saving information, is computationally intractable.

The PSO, BA and BHO algorithms were implemented based on the generic structure presented in Algorithm 1, following their differences described in Section 4.2. All the codes are available in [128]. The algorithms were first used on the G_1 network so that their parameters could be modified if necessary to achieve the efficient solutions found with the brute force algorithm. For the PSO algorithm, we obtained efficient solutions by fixing the parameters as follows, according to [129]: $c_1 = c_2 = 1$, $\phi_1 = \phi_2 = Random(0, 1)$, $w = 1$. For the BA algorithm, the parameters were subtly adjusted as follows: α and γ were left at 0.9 (the author postulates a range between 0.9 and 0.998 [130]), ϵ was set to 1, and the best values for f_{min} and f_{max} were obtained in this network for 0.5 and 1.5, respectively, although these frequency values are adjusted independently for each problem. The same number of virtual agents (population size) was used: 25 individuals. All these algorithms were run on the same hardware as the brute force algorithm.

6. Results and discussion

The three algorithms (PSO, BA, and BHO) were executed on G_1, G_2 and G_3 networks with 1000, 10,000 and 100,000 iterations, respectively. We carry out 30 individual tests for each case.

The general results are summarized in Table 3. Note that all of these results are based on the multi-objective paradigm. Therefore, the 'min' and 'max' results indicated in Table 3 correspond to the ideal-points obtained for $\min\{|x|\}$ and $\max\{|F(x)|\}$, respectively, while seeking to maximize the $\max\{|F(x)| - |x|\}$ difference. This value represents a significance measure to evidence how these swarm techniques are finding efficient solutions while they run their search intelligence procedures. In this context, Fig. 2 presents the convergence graphs showing the evolution of the nodes reached in terms of the number of iterations. Note that 'max', 'min', and 'max − min' correspond to different executions, as they represent the best solutions of each type.

6.1. Football network

For the G_1 network, PSO reached efficient solutions (verified by the exact algorithm described in Section 5.2) on all individual tests, with seeds of $|x| = 15$ nodes and an influence spread of $|F(x)| = n = 35$ nodes. This explains the standard deviations and IQR equal to zero in Table 3. Furthermore, in Fig. 2(a) a very fast convergence is observed, within the first 30 iterations. Fig. 3 illustrates a zoom-in of how the algorithm evolved in these 30 iterations, for each of the individual tests.

For the same network, BA also reached an efficient solution set, although it required about 500 iterations (Fig. 2(c)) and did not exhibit a good performance for all individual tests. In fact, the smallest seed reached has 12 nodes, less than the amount necessary to spread the influence throughout the network. It reached seeds of average size $15.77 > 15$, achieving an average influence spread of $33.67 < 35$ nodes, with an average difference of $17.90 < 20$ nodes.

Meanwhile, BHO failed to achieve efficient solutions. In general it experienced little variation throughout the iterations (Fig. 2(b)), with late jumps, after several iterations without modification. This problem, as we will see, was accentuated for BHO and BA in the two largest networks G_2 and G_3. The lowest seed reached 11 nodes, with an average of $14.13 < 15$, and the highest influence spread only reached 32 nodes, with an average of $30.47 < 35$ nodes. The average difference was $16.33 < 20$ nodes, with a standard deviation of 0.54, well below the 1.45 for BA.

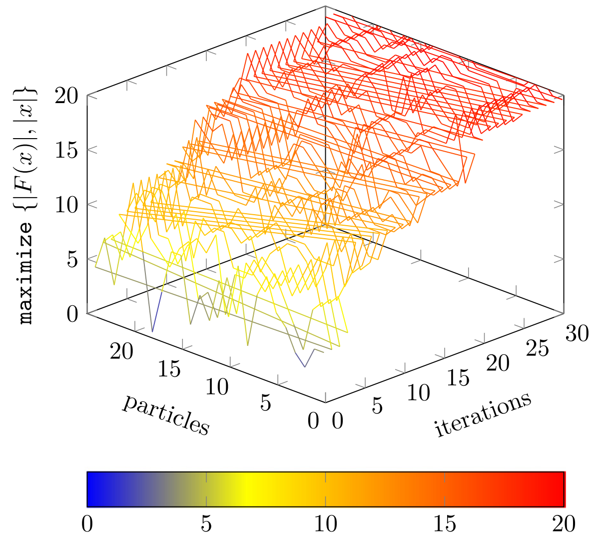


Fig. 3. Evolution of PSO on G_1 for the first 30 iterations and for each individual test (particles).

6.2. Bitcoin Alpha network

For this network and the next one, we do not know the set of efficient solutions, since it is not feasible to compute them for the exact brute force algorithm.

Notwithstanding the foregoing, PSO managed in all tests to find seeds capable of spreading its influence up to 3783 nodes in the G_2 network. The minimum seed size was 676 nodes, with an average of 729.73 nodes and a standard deviation of 29.97. The best result for the difference was 3107 nodes, with an average of 3053.27 nodes and a standard deviation of 29.97. Note that, according to Fig. 2(d), for the best solutions found, the algorithm converged after approximately 6000 iterations, demonstrating sustained and especially rapid improvement during the first 2000 iterations.

In contrast, the results for the other two algorithms were very poor. BHO only managed to mutate a little in the first 500 iterations (Fig. 2(e)), but could not escape local optimums, very far from the solutions obtained by PSO. BA's behavior, on the other hand, was completely linear (Fig. 2(f)), failing to improve throughout the 10000 iterations.

6.3. Higgs Twitter network

For the G_3 network, PSO found a seed with an influence spread of 99.7% of the nodes (38794 of 38918). Furthermore, Fig. 2(g) shows a constant growth that does not converge, hence with a greater number of iterations, even better solutions could be obtained, possibly reaching all the nodes of the network. On average it reached 38719 nodes, with a low standard deviation of 19.06. The smallest seed found has 14067 nodes, averaging 14118 and an even smaller standard deviation of 12.77. The largest difference obtained was 24704 nodes, with an average of 24626 nodes and a standard deviation of 21.04.

In contrast, as for G_2, the results for BHO and BA were very poor, with a minimal improvement of BHO in the first 2000 iterations (Fig. 2(h)), and with a completely linear behavior for the BA algorithm (Fig. 2(i)).

Table 3
Computational results. $\hat{x}$ denotes the best result, $\check{x}$ the worst, $\bar{x}$ the average, σ the standard deviation, $\tilde{x}$ the median value, μ the interquartile range (IQR), min denotes $\min\{|x|\}$ and max denotes $\max\{|F(x)|\}$.

		PSO			BHO			BA		
Network:		G_1	G_2	G_3	G_1	G_2	G_3	G_1	G_2	G_3
min	$\hat{x}$	15	676	14067	11	1739	19339	12	1805	19260
	$\check{x}$	15	770	14118	16	1789	19586	19	1892	19626
	$\bar{x}$	15	729.73	14086	14.13	1769.87	19440.17	15.77	1842.77	19476.37
	σ	0	29.97	12.77	1.52	14.30	76.46	1.80	22.16	100.89
	$\tilde{x}$	15	728.50	14085	14.50	1774	19440.50	16	1844	19482
	μ	0	46.50	19.50	2	17	140	2	16	142
max	$\hat{x}$	35	3783	38794	32	3523	28100	35	3538	27768
	$\check{x}$	35	3783	38719	27	3472	27800	31	3480	27332
	$\bar{x}$	35	3783	38754.70	30.47	3502.80	27918.03	33.67	3501.60	27533
	σ	0	0	19.06	1.48	11.72	80.33	1.07	13.53	111.76
	$\tilde{x}$	35	3783	38758	31	3502	27909	34	3500.50	27529.50
	μ	0	0	31.50	2	18	126	1.50	17	148.50
max − min	$\hat{x}$	20	3107	24704	18	1747	8547	20	1696	8254
	$\check{x}$	20	3013	24626	16	1725	8422	14	1633	7850
	$\bar{x}$	20	3053.27	24668.70	16.33	1732.93	8477.87	17.90	1658.83	8056.63
	σ	0	29.97	21.04	0.54	6.02	30.61	1.45	14.69	90.41
	$\tilde{x}$	20	3054.50	24666	16	1731	8470	18	1654	8055
	μ	0	46.50	35.50	1	10.50	38.50	2	18	142

6.4. Discussion

For networks of different sizes, PSO proved to be an excellent alternative to find solutions under the multi-objective paradigm. This algorithm maintains a sustained improvement as the number of iterations increases, and a low standard deviation in its efficient solutions. In contrast, BHO and BA were not useful in networks with thousands of nodes. BA allowed to find efficient solutions for small networks but did not present any improvement for larger networks. BHO did not reach a set of efficient solutions even on the smallest network G_1, but unlike BA, it achieved small and slow improvements due to mutations in G_2 and G_3, still being far from PSO.

The disadvantages that BA and BHO showed compared to PSO are because their ways of exploring do not serve much in binary domains, being much more useful in real ones. PSO can improve significantly and steadily since its memory capacity and registration by particles (solution vector) prevents it from getting worse. At the same time, this is a disadvantage at runtime, as each particle handles two solution vector lists: `x` and `pBest` [129], which in the case of networks with large amounts of nodes, it constantly implies cloning this list in case of improvement.

The fitness computation, which in this case is the maximum value of the influence spread $F(x)$ obtained from the minimum initial seed x, becomes slower as the iterations progress, noticing much more in PSO. This is because the depth of $F(x)$ (i-parameter in Eq. (1)) grows as the iterations progress, increasing its computation time. This is especially clear for the PSO executions on G_2, where the first iterations were much faster than the last ones. In effect, PSO started iteration 0 with a seed of 1850 nodes on average, influencing an average of 3490 nodes, almost double. Due to the above, and the fact that BA and BHO do not handle memory (list by solution vectors), the PSO execution times were almost double that of BA and BHO, on G_2 and G_3.

6.5. Comparison with centrality measures

The IMP (as well as the multi-objective version studied in this work) are different from the centrality problem [12]. The solution seeds obtained from the IMP and the multi-objective version reflect a set of influential actors. In contrast, the output of the centrality measures is a ranking that allows determining the individual relevance of each actor within the network in terms of influence, activity, or popularity [131].

Notwithstanding the above, if we consider the actors of a solution seed of our problem as influential actors, then it is possible to compare the k actors obtained from a solution seed with the top k users obtained by a centrality measure. To make these comparisons, we apply three classic centrality measures used for directed, weighted graphs (out-degree [132], closeness [132], and PageRank [133]) on the Football network (G_1). These three measures are implemented and well documented in Python's NetworkX library.[1]

As mentioned in Section 6.1, an efficient solution seed for G_1 yielded 15 nodes, which correspond to the smallest set capable of spreading its influence to the 35 nodes of the network. For the top 15 nodes obtained with the out-degree measure, ten nodes coincided with the solution seed. However, the second most relevant node for out-degree does not belong to the optimal solution of our multi-objective IMP, while two nodes of the solution seed are at the bottom of the ranking for out-degree. Since the out-degree measure only depends on the number of salient edges of each node, these differences in the results are understandable.

On the other hand, from the top 15 obtained for closeness and PageRank, no match is obtained with the nodes of the solution seed. Indeed, both the closeness and the PageRank depend on the trajectories of each node toward the other network nodes. Therefore, these measures punish nodes that do not have arrival edges (low in-degree). However, in the case of our problem, if we want to maximize the influence spread in the network, the nodes with in-degree zero or close to zero must necessarily belong to the solution seed. Otherwise, they could never be activated through the influence spread process from a seed that does not contain them.

As we can see, each of these algorithms give different results as they seek to solve different problems. That is why both problems (centrality and IMP) have had so much development separately (see Table 1 and [131]).

7. Conclusions and future work

The influence maximization problem (IMP) is one of the most relevant problems in social network analysis. In this work, based on an extensive bibliographic review (see Table 1), we have seen that solutions through metaheuristics have dominated the proposals in the last five years. A lesser-known problem is the

[1] https://networkx.github.io/

least cost influence (LCI) problem, the minimization version of IMP, which has only been recently studied using metaheuristics, in particular through evolutionary algorithms. As far as we know, a multi-objective strategy has never been used until now to solve both problems at the same time, that is, seek to minimize the size of the seeds while at the same time maximizing the influence spread.

In this paper, we have modeled this multi-objective problem and implemented three swarm intelligence-based methods to solve it: Particle swarm optimization (PSO), Bat algorithm (BA), and Black hole optimization (BHO). As case studies, we consider three social networks represented as influence graphs, of 35, 3783 and 38918 nodes, respectively. To exert the influence spread phenomenon, we use the linear threshold model (LT-model).

The results show that the PSO algorithm achieved excellent results for the three networks, showing robust convergent behavior toward obtaining efficient solutions. The BA algorithm reached efficient solutions for the smallest network, but failed to evolve for the largest networks. BHO showed a much slower evolution capacity than PSO. Unlike BA, it managed to evolve subtly for the largest networks but failed to achieve a set of efficient solutions even for the smallest network. Thus, although the use of the internal memory of PSO increases its execution times, this seems to be a very good option to solve the influence spread problem under a multi-objective paradigm. Instead, BA and BHO were unable to properly adapt to the binary domain, so the authors suggest ruling them out for future experiments, at least in their more standard versions. An alternative to consider could be the Crow search algorithm [134], which, like PSO, also uses internal memory and it already has successfully been applied to 0/1 combinatorial problems [135].

Finally, as future work, it is expected to apply this proposed solution in other application domains, such as cooperative game theory, collective decision-making or network centrality.

CRediT authorship contribution statement

Rodrigo Olivares: Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing. **Francisco Muñoz:** Software, Validation, Investigation, Writing. **Fabián Riquelme:** Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing.

Declaration of competing interest

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

Acknowledgments

F. Riquelme has been partially supported by Fondecyt de Iniciación 11200113, Chile, and by funds from the Spanish Ministry of Economy and Competitiveness (MINECO) and from the European Union (FEDER funds) under grant PID2019-104987GB-I00 (MINECO/FEDER).

References

[1] J. Sun, J. Tang, A survey of models and algorithms for social influence analysis, in: Social Network Data Analytics, Springer US, 2011, pp. 177–214, http://dx.doi.org/10.1007/978-1-4419-8462-3_7.
[2] D. Easley, J. Kleinberg, Networks, Crowds, and Markets, Cambridge University Press, 2010, http://dx.doi.org/10.1017/cbo9780511761942.
[3] D. Gruhl, D. Liben-Nowell, R. Guha, A. Tomkins, Information diffusion through blogspace, ACM SIGKDD Explor. Newsl. 6 (2) (2004) 43–52, http://dx.doi.org/10.1145/1046456.1046462.
[4] X. Song, B.L. Tseng, C.-Y. Lin, M.-T. Sun, Personalized recommendation driven by information flow, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR'06, ACM Press, 2006, pp. 509–516, http://dx.doi.org/10.1145/1148170.1148258.
[5] X. Zhang, J. Zhu, Q. Wang, H. Zhao, Identifying influential nodes in complex networks with community structure, Knowl.-Based Syst. 42 (2013) 74–84, http://dx.doi.org/10.1016/j.knosys.2013.01.017.
[6] J. Li, W. Peng, T. Li, T. Sun, Q. Li, J. Xu, Social network user influence sense-making and dynamics prediction, Expert Syst. Appl. 41 (11) (2014) 5115–5124, http://dx.doi.org/10.1016/j.eswa.2014.02.038.
[7] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (7563) (2015) 65–68, http://dx.doi.org/10.1038/nature14604.
[8] X. Molinero, F. Riquelme, M. Serna, Cooperation through social influence, European J. Oper. Res. 242 (3) (2015) 960–974, http://dx.doi.org/10.1016/j.ejor.2014.11.006.
[9] N. Cassavia, E. Masciari, C. Pulice, D. Sacca, I. Trubitsyna, Evaluating the influence of user searches on neighbors, in: 2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, IEEE, 2018, pp. 165–170, http://dx.doi.org/10.1109/wetice.2018.00038.
[10] X. Molinero, F. Riquelme, M. Serna, Satisfaction and power in unanimous majority influence decision models, Electron. Notes Discrete Math. 68 (2018) 197–202, http://dx.doi.org/10.1016/j.endm.2018.06.034.
[11] X. Molinero, F. Riquelme, M. Serna, Measuring satisfaction and power in influence based decision systems, Knowl.-Based Syst. 174 (2019) 144–159, http://dx.doi.org/10.1016/j.knosys.2019.03.005.
[12] F. Riquelme, P. Gonzalez-Cantergiani, X. Molinero, M. Serna, Centrality measure in social networks based on linear threshold model, Knowl.-Based Syst. 140 (2018) 92–102, http://dx.doi.org/10.1016/j.knosys.2017.10.029.
[13] F. Riquelme, P. Gonzalez-Cantergiani, X. Molinero, M. Serna, The neighborhood role in the linear threshold rank on social networks, Physica A 528 (2019) 121430, http://dx.doi.org/10.1016/j.physa.2019.121430.
[14] P.M. Domingos, M. Richardson, Mining the network value of customers, in: D. Lee, M. Schkolnick, F.J. Provost, R. Srikant (Eds.), Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 26–29, 2001, ACM, 2001, pp. 57–66.
[15] D. Kempe, J. Kleinberg, É. Tardos, 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, 2003, pp. 137–146, http://dx.doi.org/10.1145/956750.956769.
[16] M. Granovetter, Threshold models of collective behavior, Am. J. Sociol. 83 (6) (1978) 1420–1443, http://dx.doi.org/10.1086/226707.
[17] T. Schelling, Micromotives and Macrobehavior, Norton, 1978.
[18] J. Goldenberg, B. Libai, E. Muller, Using Complex Systems Analysis to Advance Marketing Theory Development: Modeling Heterogeneity Effects on New Product Growth Through Stochastic Cellular Automata, Tech. Rep., Academy of Marketing Science Review, 2001.
[19] C. Long, R.C.-W. Wong, Minimizing seed set for viral marketing, in: 2011 IEEE 11th International Conference on Data Mining, IEEE, 2011, pp. 427–436, http://dx.doi.org/10.1109/icdm.2011.99.
[20] D.T. Nguyen, S. Das, M.T. Thai, Influence maximization in multiple online social networks, in: 2013 IEEE Global Communications Conference, GLOBECOM, IEEE, 2013, pp. 3060–3065, http://dx.doi.org/10.1109/glocom.2013.6831541.
[21] S. Zhu, L. Xu, E.D. Goodman, Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy, Knowl.-Based Syst. (2019) 105018, http://dx.doi.org/10.1016/j.knosys.2019.105018.
[22] K. Liagkouras, A new three-dimensional encoding multiobjective evolutionary algorithm with application to the portfolio optimization problem, Knowl.-Based Syst. 163 (2019) 186–203, http://dx.doi.org/10.1016/j.knosys.2018.08.025.
[23] N.E. Karkalos, A.P. Markopoulos, J.P. Davim, Swarm intelligence-based methods, in: Computational Methods for Application in Industry 4.0, Springer International Publishing, 2018, pp. 33–55, http://dx.doi.org/10.1007/978-3-319-92393-2_3.
[24] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: Algorithms and applications, Swarm Evol. Comput. 33 (2017) 1–17, http://dx.doi.org/10.1016/j.swevo.2016.12.005.
[25] F. Valdez, A review of optimization swarm intelligence-inspired algorithms with type-2 fuzzy logic parameter adaptation, Soft Comput. 24 (1) (2019) 215–226, http://dx.doi.org/10.1007/s00500-019-04290-y.
[26] M. Gong, J. Yan, B. Shen, L. Ma, Q. Cai, Influence maximization in social networks based on discrete particle swarm optimization, Inform. Sci. 367–368 (2016) 600–614, http://dx.doi.org/10.1016/j.ins.2016.07.012.

[27] J. Tang, R. Zhang, Y. Yao, Z. Zhao, P. Wang, H. Li, J. Yuan, Maximizing the spread of influence via the collective intelligence of discrete bat algorithm, Knowl.-Based Syst. 160 (2018) 88–103, http://dx.doi.org/10.1016/j.knosys.2018.06.013.
[28] S.S. Singh, A. Kumar, S. Mishra, K. Singh, B. Biswas, Influence maximization in social networks, in: Optimization in Large Scale Problems, Springer International Publishing, 2019, pp. 255–267, http://dx.doi.org/10.1007/978-3-030-28565-4_22.
[29] R. Michalski, P. Kazienko, Maximizing social influence in real-world networks - The state of the art and current challenges, in: Intelligent Systems Reference Library, Springer International Publishing, 2015, pp. 329–359, http://dx.doi.org/10.1007/978-3-319-15916-4_14.
[30] N. Hafiene, W. Karoui, L.B. Romdhane, Influential nodes detection in dynamic social networks: A survey, Expert Syst. Appl. (2020) 113642, http://dx.doi.org/10.1016/j.eswa.2020.113642.
[31] N. Chen, On the approximability of influence in social networks, SIAM J. Discrete Math. 23 (3) (2009) 1400–1415, http://dx.doi.org/10.1137/08073617x.
[32] S. Bhagat, A. Goyal, L.V. Lakshmanan, Maximizing product adoption in social networks, in: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining - WSDM'12, ACM Press, 2012, pp. 603–612, http://dx.doi.org/10.1145/2124295.2124368.
[33] M. Gomez-Rodriguez, L. Song, N. Du, H. Zha, B. Schölkopf, Influence estimation and maximization in continuous-time diffusion networks, ACM Trans. Inf. Syst. 34 (2) (2016) 1–33, http://dx.doi.org/10.1145/2824253.
[34] Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, S.-Y. Lee, CIM: Community-based influence maximization in social networks, ACM Trans. Intell. Syst. Technol. 5 (2) (2014) 1–31, http://dx.doi.org/10.1145/2532549.
[35] H. Li, S.S. Bhowmick, A. Sun, J. Cui, Conformity-aware influence maximization in online social networks, VLDB J. 24 (1) (2014) 117–141, http://dx.doi.org/10.1007/s00778-014-0366-x.
[36] C.P. Sankar, S. Asharaf, K.S. Kumar, Learning from bees: An approach for influence maximization on viral campaigns, PLoS One 11 (12) (2016) e0168125, http://dx.doi.org/10.1371/journal.pone.0168125.
[37] Q. Jiang, G. Song, C. Gao, Y. Wang, W. Si, K. Xie, Simulated annealing based influence maximization in social networks, in: AAAI'11: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, ACM Press, 2011, pp. 127–132.
[38] C.-W. Tsai, Y.-C. Yang, M.-C. Chiang, A genetic NewGreedy algorithm for influence maximization in social network, in: 2015 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2015, pp. 2549–2554, http://dx.doi.org/10.1109/smc.2015.446.
[39] D. Bucur, G. Iacca, Influence maximization in social networks with genetic algorithms, in: Applications of Evolutionary Computation, Springer International Publishing, 2016, pp. 379–392, http://dx.doi.org/10.1007/978-3-319-31204-0_25.
[40] K. Zhang, H. Du, M.W. Feldman, Maximizing influence in a social network: Improved results using a genetic algorithm, Physica A 478 (2017) 20–30, http://dx.doi.org/10.1016/j.physa.2017.02.067.
[41] A. Sheikhahmadi, A. Zareie, Identifying influential spreaders using multi-objective artificial bee colony optimization, Appl. Soft Comput. 94 (2020) 106436, http://dx.doi.org/10.1016/j.asoc.2020.106436.
[42] A. Şimşek, R. Kara, Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks, Expert Syst. Appl. 114 (2018) 224–236, http://dx.doi.org/10.1016/j.eswa.2018.07.038.
[43] J. Tang, R. Zhang, P. Wang, Z. Zhao, L. Fan, X. Liu, A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks, Knowl.-Based Syst. 187 (2020) 104833, http://dx.doi.org/10.1016/j.knosys.2019.07.004.
[44] M. Weskida, R. Michalski, Evolutionary algorithm for seed selection in social influence process, in: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, IEEE, 2016, pp. 1189–1196, http://dx.doi.org/10.1109/asonam.2016.7752390.
[45] L. Cui, H. Hu, S. Yu, Q. Yan, Z. Ming, Z. Wen, N. Lu, DDSE: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks, J. Netw. Comput. Appl. 103 (2018) 119–130, http://dx.doi.org/10.1016/j.jnca.2017.12.003.
[46] M. Weskida, R. Michalski, Finding influentials in social networks using evolutionary algorithm, J. Comput. Sci. 31 (2019) 77–85, http://dx.doi.org/10.1016/j.jocs.2018.12.010.
[47] M. Kimura, K. Saito, Tractable models for information diffusion in social networks, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 259–271, http://dx.doi.org/10.1007/11871637_27.
[48] P.A. Estevez, P. Vera, K. Saito, Selecting the most influential nodes in social networks, in: 2007 International Joint Conference on Neural Networks, IEEE, 2007, pp. 2397–2402, http://dx.doi.org/10.1109/ijcnn.2007.4371333.
[49] M. Kimura, K. Saito, R. Nakano, Extracting influential nodes for information diffusion on a social network, in: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22–26, 2007, Vancouver, British Columbia, Canada, AAAI Press, 2007, pp. 1371–1376.
[50] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'09, ACM Press, 2009, pp. 199–208, http://dx.doi.org/10.1145/1557019.1557047.
[51] Y. Wang, X. Feng, A potential-based node selection strategy for influence maximization in a social network, in: Advanced Data Mining and Applications, Springer Berlin Heidelberg, 2009, pp. 350–361, http://dx.doi.org/10.1007/978-3-642-03348-3_34.
[52] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 88–97, http://dx.doi.org/10.1109/icdm.2010.118.
[53] W. Chen, C. Wang, Y. Wang, 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, 2010, pp. 1029–1038, http://dx.doi.org/10.1145/1835804.1835934.
[54] Y. Wang, G. Cong, G. Song, K. Xie, Community-based greedy algorithm for mining top-k influential nodes in mobile social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'10, ACM Press, 2010, pp. 1039–1048, http://dx.doi.org/10.1145/1835804.1835935.
[55] S. Kundu, C.A. Murthy, S.K. Pal, A new centrality measure for influence maximization in social networks, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 242–247, http://dx.doi.org/10.1007/978-3-642-21786-9_40.
[56] A. Goyal, W. Lu, L.V. Lakshmanan, SIMPATH: An efficient algorithm for influence maximization under the linear threshold model, in: 2011 IEEE 11th International Conference on Data Mining, IEEE, 2011, pp. 211–220, http://dx.doi.org/10.1109/icdm.2011.132.
[57] A. Goyal, W. Lu, L.V. Lakshmanan, CELF++, in: Proceedings of the 20th International Conference Companion on World Wide Web - WWW'11, ACM Press, 2011, pp. 47–48, http://dx.doi.org/10.1145/1963192.1963217.
[58] R. Narayanam, Y. Narahari, A shapley value-based approach to discover influential nodes in social networks, IEEE Trans. Autom. Sci. Eng. 8 (1) (2011) 130–147, http://dx.doi.org/10.1109/tase.2010.2052042.
[59] A. Goyal, F. Bonchi, L.V.S. Lakshmanan, S. Venkatasubramanian, On minimizing budget and time in influence propagation over social networks, Soc. Netw. Anal. Min. 3 (2) (2012) 179–192, http://dx.doi.org/10.1007/s13278-012-0062-z.
[60] K. Jung, W. Heo, W. Chen, IRIE: Scalable and robust influence maximization in social networks, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 918–923, http://dx.doi.org/10.1109/icdm.2012.79.
[61] X. Liu, S. Li, X. Liao, L. Wang, Q. Wu, In-time estimation for influence maximization in large-scale social networks, in: Proceedings of the Fifth Workshop on Social Network Systems - SNS'12, ACM Press, 2012, pp. 1–6, http://dx.doi.org/10.1145/2181176.2181179.
[62] Z.-L. Luo, W.-D. Cai, Y.-J. Li, D. Peng, A PageRank-based heuristic algorithm for influence maximization in the social network, in: Lecture Notes in Electrical Engineering, Springer Berlin Heidelberg, 2012, pp. 485–490, http://dx.doi.org/10.1007/978-3-642-28798-5_65.
[63] P. Shakarian, D. Paulo, Large social networks can be targeted for viral marketing with small seed sets, in: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, IEEE, 2012, pp. 1–8, http://dx.doi.org/10.1109/asonam.2012.11.
[64] C. Wang, W. Chen, Y. Wang, Scalable influence maximization for independent cascade model in large-scale social networks, Data Min. Knowl. Discov. 25 (3) (2012) 545–576, http://dx.doi.org/10.1007/s10618-012-0262-1.
[65] S. Cheng, H. Shen, J. Huang, G. Zhang, X. Cheng, StaticGreedy, in: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management - CIKM'13, ACM Press, 2013, pp. 509–518, http://dx.doi.org/10.1145/2505515.2505541.
[66] S. Goldberg, Z. Liu, The diffusion of networking technologies, in: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2013, pp. 1577–1594, http://dx.doi.org/10.1137/1.9781611973105.113.
[67] J. Kim, S.-K. Kim, H. Yu, Scalable and parallelizable processing of influence maximization for large-scale social networks? in: 2013 IEEE 29th International Conference on Data Engineering, ICDE, IEEE, 2013, pp. 266–277, http://dx.doi.org/10.1109/icde.2013.6544831.
[68] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2014, pp. 946–957, http://dx.doi.org/10.1137/1.9781611973402.70.
[69] S. Cheng, H. Shen, J. Huang, W. Chen, X. Cheng, IMRank, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR'14, ACM Press, 2014, pp. 475–484, http://dx.doi.org/10.1145/2600428.2609592.

[70] E. Cohen, D. Delling, T. Pajor, R.F. Werneck, Sketch-based influence maximization and computation, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM'14, ACM Press, 2014, pp. 629–638, http://dx.doi.org/10.1145/2661829.2662077.

[71] J.-R. Lee, C.-W. Chung, A fast approximation for influence maximization in large social networks, in: Proceedings of the 23rd International Conference on World Wide Web - WWW'14 Companion, ACM Press, 2014, pp. 1157–1162, http://dx.doi.org/10.1145/2567948.2580063.

[72] Z. Lu, L. Fan, W. Wu, B. Thuraisingham, K. Yang, Efficient influence spread estimation for influence maximization under the linear threshold model, Comput. Soc. Netw. 1 (1) (2014) http://dx.doi.org/10.1186/s40649-014-0002-3.

[73] N. Ohsaka, T. Akiba, Y. Yoshida, K. Kawarabayashi, Fast and accurate influence maximization on large networks with pruned Monte–Carlo simulations, in: C.E. Brodley, P. Stone (Eds.), Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27–31, 2014, Québec City, Québec, Canada, AAAI Press, 2014, pp. 138–144.

[74] S.K. Pal, S. Kundu, C.A. Murthy, Centrality measures, upper bound, and influence maximization in large scale directed social networks, Fund. Inform. 130 (3) (2014) 317–342, http://dx.doi.org/10.3233/FI-2014-994.

[75] Y. Tang, X. Xiao, Y. Shi, Influence maximization, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data - SIGMOD'14, ACM Press, 2014, pp. 75–86, http://dx.doi.org/10.1145/2588555.2593670.

[76] T. Zhu, B. Wang, B. Wu, C. Zhu, Maximizing the spread of influence ranking in social networks, Inform. Sci. 278 (2014) 535–544, http://dx.doi.org/10.1016/j.ins.2014.03.070.

[77] P. Zhang, W. Chen, X. Sun, Y. Wang, J. Zhang, Minimizing seed set selection with probabilistic coverage guarantee in a social network, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '14, ACM Press, 2014, pp. 1306–1315, http://dx.doi.org/10.1145/2623330.2623684.

[78] M. Heidari, M. Asadpour, H. Faili, SMG: Fast scalable greedy algorithm for influence maximization in social networks, Physica A 420 (2015) 124–133, http://dx.doi.org/10.1016/j.physa.2014.10.088.

[79] S. Kundu, S.K. Pal, Deprecation based greedy strategy for target set selection in large scale social networks, Inform. Sci. 316 (2015) 107–122, http://dx.doi.org/10.1016/j.ins.2015.04.024.

[80] K. Rahimkhani, A. Aleahmad, M. Rahgozar, A. Moeini, A fast algorithm for finding most influential people based on the linear threshold model, Expert Syst. Appl. 42 (3) (2015) 1353–1361, http://dx.doi.org/10.1016/j.eswa.2014.09.037.

[81] G. Song, X. Zhou, Y. Wang, K. Xie, Influence maximization on large-scale mobile social network: A divide-and-conquer method, IEEE Trans. Parallel Distrib. Syst. 26 (5) (2015) 1379–1392, http://dx.doi.org/10.1109/tpds.2014.2320515.

[82] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data - SIGMOD'15, ACM Press, 2015, pp. 1539–1554, http://dx.doi.org/10.1145/2723372.2723734.

[83] Q. Zhao, H. Lu, Z. Gan, X. Ma, A K-shell decomposition based algorithm for influence maximization, in: Engineering the Web in the Big Data Era, Springer International Publishing, 2015, pp. 269–283, http://dx.doi.org/10.1007/978-3-319-19890-3_18.

[84] M. Gong, C. Song, C. Duan, L. Ma, B. Shen, An efficient memetic algorithm for influence maximization in social networks, IEEE Comput. Intell. Mag. 11 (3) (2016) 22–33, http://dx.doi.org/10.1109/mci.2016.2572538.

[85] M. Hosseini-Pozveh, K. Zamanifar, A.R. Naghsh-Nilchi, A community-based approach to identify the most influential nodes in social networks, J. Inf. Sci. 43 (2) (2016) 204–220, http://dx.doi.org/10.1177/0165551515621005.

[86] W.-X. Lu, C. Zhou, J. Wu, Big social network influence maximization via recursively estimating influence spread, Knowl.-Based Syst. 113 (2016) 143–154, http://dx.doi.org/10.1016/j.knosys.2016.09.020.

[87] H.T. Nguyen, M.T. Thai, T.N. Dinh, Stop-and-stare, in: Proceedings of the 2016 International Conference on Management of Data - SIGMOD'16, ACM Press, 2016, pp. 695–710, http://dx.doi.org/10.1145/2882903.2915207.

[88] A. Sheikhahmadi, M.A. Nematbakhsh, Identification of multi-spreader users in social networks for viral marketing, J. Inf. Sci. 43 (3) (2016) 412–423, http://dx.doi.org/10.1177/0165551516644171.

[89] X. Wang, Y. Su, C. Zhao, D. Yi, Effective identification of multiple influential spreaders by degreepunishment, Physica A 461 (2016) 238–247, http://dx.doi.org/10.1016/j.physa.2016.05.020.

[90] S. Kim, D. Kim, J. Oh, J.-H. Hwang, W.-S. Han, W. Chen, H. Yu, Scalable and parallelizable influence maximization with random walk ranking and rank merge pruning, Inform. Sci. 415–416 (2017) 171–189, http://dx.doi.org/10.1016/j.ins.2017.06.018.

[91] J. Shang, S. Zhou, X. Li, L. Liu, H. Wu, CoFIM: A community-based framework for influence maximization on large-scale networks, Knowl.-Based Syst. 117 (2017) 88–100, http://dx.doi.org/10.1016/j.knosys.2016.09.029.

[92] S. Stein, S. Eshghi, S. Maghsudi, L. Tassiulas, R.K.E. Bellamy, N.R. Jennings, Heuristic algorithms for influence maximization in partially observable social networks, in: M.G. Armentano, J. Tang, V. Yannibelli (Eds.), Proceedings of the 3rd International Workshop on Social Influence Analysis Co-Located with 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19, 2017, in: CEUR Workshop Proceedings, vol. 1893, CEUR-WS.org, 2017, pp. 20–32.

[93] M.-E.G. Rossi, B. Shi, N. Tziortziotis, F.D. Malliaros, C. Giatsidis, M. Vazirgiannis, MATI: An efficient algorithm for influence maximization in social networks, PLoS One 13 (11) (2018) e0206318, http://dx.doi.org/10.1371/journal.pone.0206318.

[94] D.T. Nguyen, H. Zhang, S. Das, M.T. Thai, T.N. Dinh, Least cost influence in multiplex social networks: Model representation and analysis, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 567–576, http://dx.doi.org/10.1109/icdm.2013.24.

[95] B. Nettasinghe, V. Krishnamurthy, Influence maximization over Markovian graphs: A stochastic optimization approach, IEEE Trans. Signal Inf. Process. Netw. 5 (1) (2019) 1–14, http://dx.doi.org/10.1109/tsipn.2018.2832011.

[96] B. Wilder, N. Immorlica, E. Rice, M. Tambe, Maximizing influence in an unknown social network, in: S.A. McIlraith, K.Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18, the 30th Innovative Applications of Artificial Intelligence, IAAI-18, and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI-18, New Orleans, Louisiana, USA, February 2–7, 2018, AAAI Press, 2018, pp. 4743–4750.

[97] W. Ju, L. Chen, B. Li, W. Liu, J. Sheng, Y. Wang, A new algorithm for positive influence maximization in signed networks, Inform. Sci. 512 (2020) 1571–1591, http://dx.doi.org/10.1016/j.ins.2019.10.061.

[98] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'07, ACM Press, 2007, pp. 420–429, http://dx.doi.org/10.1145/1281192.1281239.

[99] S. Bharathi, D. Kempe, M. Salek, Competitive influence maximization in social networks, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 306–311, http://dx.doi.org/10.1007/978-3-540-77105-0_31.

[100] T. Carnes, C. Nagarajan, S.M. Wild, A. van Zuylen, Maximizing influence in a competitive social network, in: Proceedings of the Ninth International Conference on Electronic Commerce - ICEC'07, ACM Press, 2007, pp. 351–360, http://dx.doi.org/10.1145/1282100.1282167.

[101] T. Cao, X. Wu, S. Wang, X. Hu, Maximizing influence spread in modular social networks by optimal resource allocation, Expert Syst. Appl. 38 (10) (2011) 13128–13135, http://dx.doi.org/10.1016/j.eswa.2011.04.119.

[102] H. Zhuang, Y. Sun, J. Tang, J. Zhang, X. Sun, Influence maximization in dynamic social networks, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 1313–1318, http://dx.doi.org/10.1109/icdm.2013.145.

[103] X. Chen, G. Song, X. He, K. Xie, On influential nodes tracking in dynamic social networks, in: Proceedings of the 2015 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2015, pp. 613–621, http://dx.doi.org/10.1137/1.9781611974010.69.

[104] S. Xu, N. Xu, J. Zhang, F. Li, S. Li, Seed set selection in evolving social networks, in: 2017 3rd IEEE International Conference on Computer and Communications, ICCC, IEEE, 2017, pp. 2323–2328, http://dx.doi.org/10.1109/compcomm.2017.8322950.

[105] X. Liu, X. Liao, S. Li, S. Zheng, B. Lin, J. Zhang, L. Shao, C. Huang, L. Xiao, On the shoulders of giants: Incremental influence maximization in evolving social networks, Complexity 2017 (2017) 1–14, http://dx.doi.org/10.1155/2017/5049836.

[106] Y. Wang, J. Zhu, Q. Ming, Incremental influence maximization for dynamic social networks, in: Communications in Computer and Information Science, Springer Singapore, 2017, pp. 13–27, http://dx.doi.org/10.1007/978-981-10-6388-6_2.

[107] H. Nguyen, R. Zheng, On budgeted influence maximization in social networks, IEEE J. Sel. Areas Commun. 31 (6) (2013) 1084–1094, http://dx.doi.org/10.1109/jsac.2013.130610.

[108] Y. Wang, A.V. Vasilakos, Q. Jin, J. Ma, PPRank: Economically selecting initial users for influence maximization in social networks, IEEE Syst. J. 11 (4) (2017) 2279–2290, http://dx.doi.org/10.1109/jsyst.2014.2369526.

[109] G. Li, S. Chen, J. Feng, K. lee Tan, W. syan Li, Efficient location-aware influence maximization, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data - SIGMOD'14, ACM Press, 2014, pp. 87–98, http://dx.doi.org/10.1145/2588555.2588561.

[110] H. Sun, X. Gao, G. Chen, J. Gu, Y. Wang, Multiple influence maximization in social networks, in: Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication - IMCOM'16, ACM Press, 2016, pp. 1–8, http://dx.doi.org/10.1145/2857546.2857591.
[111] A. Talukder, M.G.R. Alam, N.H. Tran, D. Niyato, C.S. Hong, Knapsack-based reverse influence maximization for target marketing in social networks, IEEE Access 7 (2019) 44182–44198, http://dx.doi.org/10.1109/access.2019.2908412.
[112] J.-H. Cho, Y. Wang, I.-R. Chen, K.S. Chan, A. Swami, A survey on modeling and optimizing multi-objective systems, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1867–1901, http://dx.doi.org/10.1109/comst.2017.2698366.
[113] A.P. Wierzbicki, A mathematical basis for satisficing decision making, Math. Modelling 3 (5) (1982) 391–405, http://dx.doi.org/10.1016/0270-0255(82)90038-0.
[114] K. Deb, Multi-objective optimisation using evolutionary algorithms: An introduction, in: Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing, Springer London, 2011, pp. 3–34, http://dx.doi.org/10.1007/978-0-85729-652-8_1.
[115] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184, http://dx.doi.org/10.1016/j.ins.2012.08.023.
[116] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/icnn.1995.488968.
[117] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995, pp. 39–43, http://dx.doi.org/10.1109/mhs.1995.494215, https://doi.org/10.1109/mhs.1995.494215.
[118] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization, NICSO 2010, Springer Berlin Heidelberg, 2010, pp. 65–74, http://dx.doi.org/10.1007/978-3-642-12538-6_6.
[119] B. Crawford, R. Soto, G. Astorga, J. García, C. Castro, F. Paredes, Putting continuous metaheuristics to work in binary search spaces, Complexity 2017 (2017) 1–19, http://dx.doi.org/10.1155/2017/8404231.
[120] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary particle swarm optimization, Swarm Evol. Comput. 9 (2013) 1–14, http://dx.doi.org/10.1016/j.swevo.2012.09.002.
[121] A.I. Hammouri, M. Mafarja, M.A. Al-Betar, M.A. Awadallah, I. Abu-Doush, An improved dragonfly algorithm for feature selection, Knowl.-Based Syst. 203 (2020) 106131, http://dx.doi.org/10.1016/j.knosys.2020.106131.
[122] J. Wanjing, C. Hong, Z. Linlu, Multi-influences Co-existence based independent cascade model and conflict resolution strategy in social networks, in: Social Media Retrieval and Mining, Springer Berlin Heidelberg, 2013, pp. 95–105, http://dx.doi.org/10.1007/978-3-642-41629-3_8.
[123] R.A. Rossi, N.K. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, in: AAAI'15, AAAI Press, 2015, pp. 4292–4293.
[124] S. Kumar, F. Spezzano, V.S. Subrahmanian, C. Faloutsos, Edge weight prediction in weighted signed networks, in: 2016 IEEE 16th International Conference on Data Mining, ICDM, IEEE, 2016, pp. 221–230, http://dx.doi.org/10.1109/icdm.2016.0033.
[125] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, V. Subrahmanian, REV2, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM'18, ACM Press, 2018, pp. 333–341, http://dx.doi.org/10.1145/3159652.3159729.
[126] M.D. Domenico, A. Lima, P. Mougel, M. Musolesi, The anatomy of a scientific rumor, Sci. Rep. 3 (1) (2013) http://dx.doi.org/10.1038/srep02980.
[127] X. Molinero, F. Riquelme, M. Serna, Power indices of influence games and new centrality measures for agent societies and social networks, in: Ambient Intelligence - Software and Applications, Springer International Publishing, 2014, pp. 23–30, http://dx.doi.org/10.1007/978-3-319-07596-9_3.
[128] F. Muñoz, R. Olivares, F. Riquelme, Swarm intelligence algorithms for multi-objective IMP, 2020, http://dx.doi.org/10.6084/M9.FIGSHARE.13046342.
[129] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, A novel particle swarm optimization algorithm with adaptive inertia weight, Appl. Soft Comput. 11 (4) (2011) 3658–3670, http://dx.doi.org/10.1016/j.asoc.2011.01.037.
[130] X.S. Yang, X. He, Bat algorithm: Literature review and applications, Int. J. Bio-Inspired Comput. 5 (3) (2013) 141, http://dx.doi.org/10.1504/ijbic.2013.055093.
[131] F. Riquelme, P. González-Cantergiani, Measuring user influence on Twitter: A survey, Inf. Process. Manage. 52 (5) (2016) 949–975, http://dx.doi.org/10.1016/j.ipm.2016.04.003.
[132] L.C. Freeman, Centrality in social networks conceptual clarification, Social Networks 1 (3) (1978) 215–239.
[133] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Tech. Rep., Stanford InfoLab, 1999.
[134] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12, http://dx.doi.org/10.1016/j.compstruc.2016.03.001.
[135] S. Valdivia, R. Soto, B. Crawford, N. Caselli, F. Paredes, C. Castro, R. Olivares, Clustering-based binarization methods applied to the crow search algorithm for 0/1 combinatorial problems, Mathematics 8 (7) (2020) 1070, http://dx.doi.org/10.3390/math8071070.