A Biased Random Key Genetic Algorithm Applied to Target Set Selection in Viral Marketing

Albert López Serrano
alopez@iiia.csic.es
Artificial Intelligence Research Institute (IIIA-CSIC)
Bellaterra, Spain

Christian Blum
christian.blum@csic.es
Artificial Intelligence Research Institute (IIIA-CSIC)
Bellaterra, Spain

ABSTRACT

The Target Set Selection (TSS) problem is an NP-hard combinatorial optimization problem with origins in the field of social networks. There are various problem variants, all dealing with finding a smallest subset of vertices of a graph such that their influence is propagated to all nodes of the graph under a specific diffusion model. Despite the practical relevance of the problem, most existing research efforts have focused on theoretical properties restricted to certain classes of graphs. The richness in terms of theoretical results is in contrast to the scarceness of research aiming at efficiently solving the TSS problem. In this work we propose a Biased Random Key Genetic Algorithm (BRKGA) for solving the TSS problem in large-scale social networks. We consider the problem in combination with the Linear Threshold diffusion model. The obtained results show that our approach outperforms a recent heuristic from the literature.

CCS CONCEPTS

• **Mathematics of computing → Combinatorial optimization**; • **Computing methodologies → Discrete space search**; • **Theory of computation → Evolutionary algorithms**.

KEYWORDS

biased random key genetic algorithm, target set selection, social networks

ACM Reference Format:
Albert López Serrano and Christian Blum. 2022. A Biased Random Key Genetic Algorithm Applied to Target Set Selection in Viral Marketing. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3512290.3528785

1 INTRODUCTION

Research has shown that individuals are affected by the opinions and behaviors of those surrounding them [22]. This makes the modeling of word-of-mouth [28] transmission of opinions and beliefs of interest for the viral marketing community. Recently, related concepts have been applied in the context of mass media campaigns in political elections [40, 44], which makes the understanding of such belief propagation dynamics a public interest. Nevertheless, the relevance of being able to identify small groups of highly influential individuals in large social networks extends to many other areas such as, for example, substance abuse prevention [22] and medicine [18, 35, 36].

One of the main aims of viral marketing [28, 34] is reaching the largest amount of potential consumers while, at the same time, minimizing the amount of allocated resources. In the context of viral marketing, a field with an immense economical impact, a wide range of hard optimization problems in social networks have been considered in the last decades. Some of these problems have become of theoretical and practical interest to the community of computer scientists and mathematicians. The so-called Target Set Selection (TSS) problem, as proposed by Kempe et al. [29], is one of these problems. In particular, the TSS problem is a combinatorial optimization problem based on different diffusion models in graphs. The dependence on diffusion models makes the use of exact optimization techniques rather difficult. There are time-step based integer linear programming (ILP) models for similar problems [12]. However, they generally do not scale well with growing graph size.

The following two variants of the TSS problem have been studied so far: the maximization variant aims to maximize the reach of a marketing campaign given a limited amount of available resources, namely a fixed number k of vertices allowed to be considered. The problem then resolves around finding the subset of k users in the social network such that their combined influence can reach the maximum number of users in the whole network (also known as k-max or influence maximization). In contrast, the minimization variant of the problem aims to find the minimum set of users in a given social network such that they can influence the whole network (also known as J-MIN [32]). In this work, we consider the latter problem variant.

Clearly, the utilized influence propagation model is crucial for the complexity of the problem. The two most relevant models considered in the literature are the following ones:

- The *Independent Cascade (IC)* [24, 28] model is a probabilistic diffusion model which assigns a weight w to each connection in the network. This weight represents the probability that, in a given moment, if one of the two end points (users) of the node is influenced, the other one will be influenced too. Given its probabilistic nature this model is usually paired with a Monte-Carlo simulation of the diffusion process. It is based on how a single node influences its neighbors, and thus the simulation allows for parallelization.
- The *Linear Threshold (LT)* [26, 47] model is a deterministic diffusion model inspired by the adoption of behaviors

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528785

of individuals corresponding to the majoritarian behavior found in their neighborhood [22]. An LT model generalizes this approach by giving a threshold to each node in the network, representing the number of neighbors that must be influenced such that the node itself will be influenced. Note that a bottleneck of this model is the simulation time for the diffusion process. Very few works about speeding up this simulation time can be found.

When looking at the problem in terms of a combinatorial optimization problem, any subset of the nodes of a given graph—also called a target set—is a potential candidate solution. The nodes of the target set are considered as being influenced right from the start. In order to check the validity of a target set, the corresponding influence propagation model must be applied, producing a cascade of influenced nodes. If, at the end of this process, all the nodes of the graph are influenced, the corresponding target set is considered a valid solution. Note that both models outlined above preserve the activation—that is, the state of being influenced—of a node once it is produced. This is in contrast to other diffusion models in which a node can eventually be deactivated [45]. Finally, note that the aim of the TSS problem is to identify a target set of minimal size. In this paper we will focus on the TSS problem under the LT diffusion model, because the LT model is very popular in the context of social network diffusion [48].

Paper outline. In Section 2 we will give a short introduction to related work. The technical problem statement is provided in Section 3. A theoretical result for speeding up the solution evaluation is presented in Section 4, while a new heuristic and a biased random key genetic algorithm (Brkga) approach are presented in Section 5. Finally, a comprehensive experimental evaluation is described in Section 6 and conclusions and an outlook to future work are given in Section 7.

2 RELATED WORK

As previously stated, the problem we are concerned with was first proposed in 2003 [29], but was independently proposed in [32]. Since then, most of the research has been focused on finding exact methods under restricted conditions. In this section we will mention some relevant results published for the TSS problem version that we will consider in this work. A first relevant result is that the problem is NP-hard [8, 32].

The problem has been shown to be solvable in special types of graphs. For instance, both on trees [8] and on cliques [37, 43] the TSS problem is solvable in linear time. In fact, it has been shown that the tree-width of a graph governs the complexity of the problem [4], while the clique-width also parametrizes the problem along with the maximum threshold [27]. Other graph properties being studied in this context include, for example, the diameter of a graph [38]. These results have motivated theoretical research around studying the problem for specific types of graphs [19] like block-cactus, chordal or Hamming graphs, and honeycomb graphs [11].

The fact that the TSS problem under the linear threshold diffusion model can be described as a combinatorial optimization problem has motivated research in Operations Research. In fact, the problem can be stated as an Integer Linear Programming (ILP) problem, which allows for solving small problem instances. Two models have been proposed: a time-dependant ILP [12], which derives instantly from the definition, and a time-independent one [1]. Even though their are not very useful for graphs of practical size, such representation of the problem allow, for example, to discover upper and lower bounds governing the size of the optimal target set of a graph [1].

Other relevant results regard the approximability conditions of the problem under threshold bounding conditions. It is known that the inaproximability of the problem can be parametrized [3]. Research shows [5, 12] that for bounded thresholds the problem can become tractable. Nonetheless, the proposed algorithms are still exponential, triggering the need of more flexible algorithms to tackle the problem in real world settings.

While the previous results are certainly of interest and while they help to shed light on the graph properties that govern the complexity of the problem, all of them deal with exactly solving the problem. This also holds for variations of the problem, such as the latency bounded TSS (which aims to activate all the nodes of a graph in a bounded number of rounds), for which recent research is focused on exact methods for specific cases or making use of certain properties [6, 13, 31].

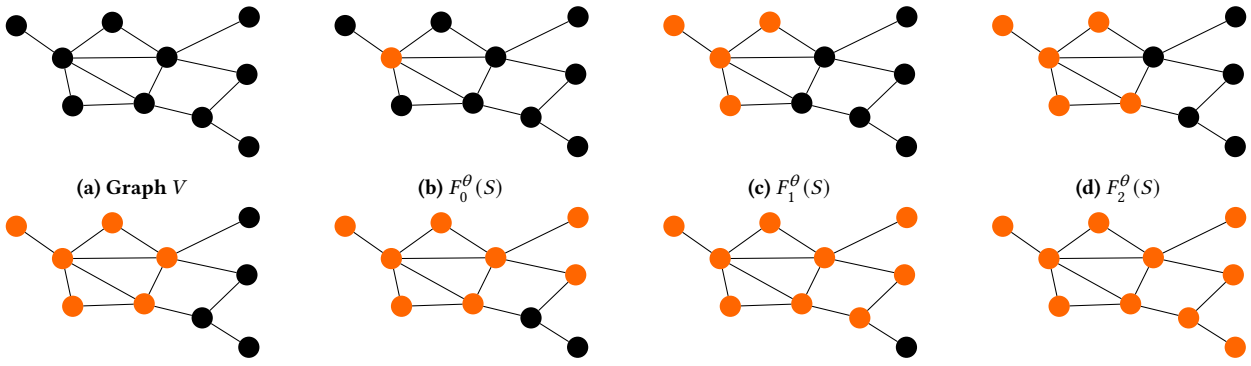
Very few works from the literature deal with practical, approximate methods for finding close-to-optimal target sets. For planar graphs, for example, a $O(\sqrt{n})$ algorithm was devised [20]. While this result is interesting for a wide range of graphs, certainly not all real-life networks are planar. Apart from this algorithm, some heuristics for the general case have been proposed. They either fall into the class of additive methods, building target sets from scratch [9, 10, 29, 32] or substractive methods, starting from the whole graph and removing nodes at each step [49]. One of the current state-of-the-art methods is a greedy depreciative heuristic from [14–16]. This algorithm, henceforth called Mts, is based on the interactions between nodes at neighborhood level, and is notable for avoiding to compute the diffusion process of a target set, as opposed to other known non-trivial heuristics.

The literature on metaheuristics for solving TSS problem variants is rather scarce. Swarm intelligence algorithms have been applied to a multi-objective problem dealing with maximizing the spread of influence of a set while minimizing its size [39]. An evolutionary algorithm (EA) was applied in[42] to a variant of the TSS problem. Finally, a set-based genetic algorithm was proposed in [50] for the TSS variant considered in this paper. This latter algorithm, however, was only successfully applied to very small networks (up to 1600 vertices).

3 TECHNICAL PROBLEM STATEMENT

Note that, for this section, we adopted the notation from [7, 32]. Let $G = (V, E)$ be an undirected graph, modeling a social network. Moreover, let $\theta : V \rightarrow \mathbb{N}$ be a function that assigns a threshold to each vertex of the graph in such a way that $\theta(v) \leq deg(v)$ for all $v \in V$, where $deg(v)$ is the degree of vertex $v \in V$. Note that the vertex degree is defined as $deg(v) := |N(v)|$, where $N(v) = \{u \in V | (u, v) \in E\}$ is the set of neighbors of v in G. The linear threshold diffusion model then defines a unique sequence of sets for the threshold function θ:

$$S = F_0^\theta(S) \subseteq F_1^\theta(S) \subseteq F_2^\theta(S) \subseteq ... \subset \sigma^\theta(S) \subseteq V \quad , \qquad (1)$$



(a) Graph V **(b)** $F_0^\theta(S)$ **(c)** $F_1^\theta(S)$ **(d)** $F_2^\theta(S)$

(e) $F_3^\theta(S)$ **(f)** $F_4^\theta(S)$ **(g)** $F_5^\theta(S)$ **(h)** $F_6^\theta(S) = \sigma(S)$

Figure 1: Diffusion process for $\theta(v) = \left\lceil \frac{deg(v)}{2} \right\rceil \ \forall v \in V$. The initial target set (consisting of only one node) is shown in (b).

where $F_t^\theta(S)$ is the set of vertices influenced at time t under the threshold function θ. A vertex v will become active (influenced) at time t if (1) v was not active at time $t-1$ and (2) the number of active neighbors of v at time $t-1$ is greater or equal to the threshold $\theta(v)$ of v. Note also that $S = F_0^\theta(S)$ is called the target set, that is, the set of influenced vertices at time 0. With these definitions we can technically describe the diffusion process by the following expression:

$$F_t^\theta(S) = F_{t-1}^\theta(S) \cup \left\{ v \in \bigcup_{u \in F_{t-1}^\theta(S)} N(u) : |N(v) \cap F_{t-1}^\theta(S)| \geq \theta(v) \right\} \tag{2}$$

$\forall t \in \mathbb{N}$. Notice how this definition satisfies Equation 1.

Definition 1. *Let $G = (V, E)$ be an undirected graph and $\theta : V \to \mathbb{N}$ a threshold function such that $\theta(v) \leq deg(v) \ \ \forall v \in V$. Then a set $S \subset V$ is a* valid target set *for this threshold function if there exists a $t_0 \in \mathbb{N}$ such that $F_{t_0}^\theta(S) = V$.*

Now we can finally state the TSS problem in technical terms: given an undirected graph $G = (V, E)$ and a threshold function $\theta : V \to \mathbb{N}$ such that $\theta(v) \leq deg(v) \ \ \forall v \in V$, find a valid target set S of minimal size. The diffusion process concerning a target set S is graphically illustrated in Figure 1.

Before moving on, we formally clarify that the problem is well-defined. Based on the deterministic rule from Equation 2 it can be shown that the diffusion process for any finite graph and any potential target set is always finite in time.

Property 1. *For every undirected graph $G = (V, E)$ with threshold function θ and every potential target set $S \subset V$, the diffusion process is finite in time and there exists a $t_0 \in \mathbb{N}$ such that:*

$$\begin{aligned} S = F_0^\theta(S) \subsetneq F_1^\theta(S) \subsetneq \ldots \subsetneq F_{t_0-1}^\theta(S) \subsetneq \\ \subsetneq F_{t_0}^\theta(S) = F_{t_0+1}^\theta(S) = \ldots = F_{t_0+k}^\theta(S) = \ldots \end{aligned} \tag{3}$$

The set of vertices $F_{t_0}^\theta(S)$ reached by applying the diffusion process to target set S is henceforth denoted by $\sigma(S)$, that is, $\sigma(S) := F_{t_0}^\theta(S)$.

Proof. We first start by proving that if for a given $t \in \mathbb{N} \cup \{0\}$ $F_t^\theta(S) = F_{t+1}^\theta(S)$, then $F_t^\theta(S) = F_{t+k}^\theta(S) \quad \forall k \in \mathbb{N}$. Using the formula of Eq. 2, we have that by hypothesis $\{v \in \bigcup_{u \in F_t^\theta(S)} N(u) : |N(v) \cap F_t^\theta(S)| \geq \theta(v)\} = \emptyset$. Since $F_t^\theta(S) = F_{t+1}^\theta(S)$, the following is true:

$$\{v \in \bigcup_{u \in F_{t+1}^\theta(S)} N(u) : |N(v) \cap F_{t+1}^\theta(S)| \geq \theta(v)\} = \emptyset$$

And thus, $F_{t+1}^\theta(S) = F_{t+2}^\theta(S)$, and by induction the proposed statement is true $\forall k \in \mathbb{N}$.

Now, since the graph is finite, there can be at most $|V \setminus S| < +\infty$ steps verifying $F_t^\theta(S) \subsetneq F_t^\theta(S)$. Thus, at some point $F_t^\theta(S) = F_t^\theta(S)$ and therefore t_0 exists and the problem is well defined. □

The previous property, although trivial, is required for the following section.

4 INCREMENTAL OBJECTIVE FUNCTION CALCULATION

In this section we will propose and prove a proposition that will allow us to speed up the diffusion process during the construction of solutions in later sections. Let us start by recalling some useful properties of the LT diffusion model.

Properties 1. *Let there be $A, B \subseteq V$ such that $A \subseteq B$. The following properties directly derive from the definition of the LT diffusion process:*

(1) $F_t^\theta(A) \subseteq F_t^\theta(B) \ \forall t \in \mathbb{N} \cup \{0\}$ and thus $\sigma(A) \subseteq \sigma(B)$
(2) $F_{t+k}^\theta(A) = F_t^\theta(F_k^\theta(A)) = F_k^\theta(F_t^\theta(A))$ for any $t, k \in \mathbb{N}$
(3) $A \subseteq F_t^\theta(A)$ for all $t \in \mathbb{N} \cup \{0\}$

All of them arise from the definition. Now we can move on to the proposition.

Proposition 1. *Let there be $S \subseteq V$ and $v \in V$. Then $\sigma(S \cup \{v\}) = \sigma(\sigma(S) \cup \{v\})$*

Proof. We will prove this statement by a double inclusion. First, notice that with the first property from above it holds that $\sigma(S \cup$

$\{v\}) \subseteq \sigma(\sigma(S) \cup \{v\})$, because $(S \cup \{v\}) \subseteq (\sigma(S) \cup \{v\})$ due to the third property from above.

Now, let us prove that $\sigma(\sigma(S) \cup \{v\}) \subseteq \sigma(S \cup \{v\})$. Let $A, B \subseteq V$, then $F_t(A) \subseteq F_t(A \cup B)$ because of the first property from above, and $B \subseteq F_t(A \cup B)$ because of the third property. Then by the second property it holds that $F_{t+1}(A \cup B) = F_1(F_t(A \cup B))$ and then $F_1(F_t(A) \cup B) \subseteq F_1(F_t(A \cup B))$. By induction it holds, therefore, for all $t, k \in \mathbb{N}$ that $F_t(F_k(A) \cup B) \subseteq F_{t+k}(A \cup B)$. With this it follows that $\sigma(\sigma(S) \cup \{v\}) \subseteq \sigma(S \cup \{v\})$. □

The former fact will allow us to speed up the computation of the diffusion process when adding vertices to a partial target set S for which we already know $\sigma(S)$.

5 THE BKRGA APPROACH

In this section we will describe our Brkga approach to the TSS problem. A BRKGA [25] is a genetic algorithm in which individuals are represented as random keys, that is, vectors of floating point values in the range $[0, 1]$. The general algorithmic framework is problem-independent. The only problem-dependent part concerns the decoding of floating point vectors into solutions to the problem. Our main reason for choosing such a decoder-based EA—in contrast to EAs working on populations of more directly encoded solutions—is that BRKGAs are ideal for solving optimization problems for which well-working greedy heuristics already exist. This is because greedy heuristic can be readily used for solution decoding. Examples of BRKGA approaches to graph-based optimization problems can be found in [17, 41, 46].

5.1 General Framework

A Brkga (see Algorithm 1) works on a fixed-size population of individuals $\vec{w}$, where $w_i \in [0, 1]$ for all $i = 1, \ldots, n$. The evolution mechanism, which is problem-independent, makes use of the following three principles: elitism, mutation and crossover. As a termination condition we will use a predefined computation time limit instead of a convergence-based criterion. In the following we will outline the different algorithmic components.

Algorithm 1 Brkga

Input: Undirected graph $G = (V, E)$, threshold function θ
Input: parameter values for t_{lim}, n_{ind}, p_e, p_m, $prob_{elite}$, $seed$
1: $P :=$ `GenerateInitialPopulation`$(n_{ind}, seed)$
2: `Evaluate`(P)
3: **while** computation time limit not reached **do**
4: $\quad P_e :=$ `EliteSolutions`(P, p_e)
5: $\quad P_m :=$ `Mutants`(P, p_m)
6: $\quad P_c :=$ `Crossover`$(P, P_e, prob_{elite})$
7: $\quad$ `Evaluate`$(P_m \cup P_c)$
8: $\quad P := P_e \cup P_m \cup P_c$
9: **end while**
Output: Best solution in P

The function `GenerateInitialPopulation`$(n_{ind}, seed)$ initializes the population of the algorithm by generating n_{ind} vectors of length $|V|$. In case $seed$ = `true`, the first one of these individuals is given value 0.5 for all positions of the random key, while the rest of the individuals is generated by drawing random values from a uniform distribution in the range $[0, 1]$. In case $seed$ = `false`, all individuals are random individuals. Afterwards, all individuals from P are evaluated—that is, decoded into solutions—in function `Evaluate`(P). The decoding procedure is outlined in detail below.

After these two initial steps, the evolutionary mechanism starts. While the time limit is not reached, a new population is generated at each iteration by executing the following three main mechanisms. In particular, the new population is composed of the union of three distinct sub-populations:

- P_e is the set of the $\lceil p_e * n_{ind} \rceil$ fittest individuals of the previous generation, also called the elites. This allows the preservation of the best traits found so far, and of the fittest individual(s) along the evolution process.
- P_m corresponds to the mutant population. More specifically, P_m is a set of $\lceil p_m * n_{ind} \rceil$ new randomly generated individuals. Note that adding new random individuals at each iteration serves the purpose of exploring the search space.
- P_c is a set of $n_{ind} - |P_e| - |P_m|$ offspring individuals generated by crossover. In the Brkga framework, the standard crossover operator is a two-parent crossover that works as follows. First, one elite solution and one none-elite solution are selected randomly from the current population P. Then, an offspring individual $\vec{w}$ is generated by choosing a value for each position w_i ($i = 1, \ldots, |V|$). In particular, this value is taken from position i of the elite parent with probability $prob_{elite}$, and from position i of the none-elite parent otherwise.

This concludes the description of the problem-independent part of our Brkga approach. In the following we will describe the decoding of an individual.

Algorithm 2 Algorithm by Kempe et al. [29]

Input: Graph $G = (V, G)$
1: $S := \emptyset$
2: **while** $\sigma(S) \neq V$ **do**
3: $\quad v_i := \text{argmax}_{v_j \in V \setminus \sigma(S)} \{|\sigma(S \cup \{v_j\})|\}$
4: $\quad S := S \cup \{v_i\}$
5: **end while**

5.2 Decoding of an individual

For the decoding of an individual $\vec{w}$ into a solution to the TSS problem we will use the mechanism of a greedy heuristic. In particular, we will use the mechanism of additive methods such as the one proposed by Kempe et al. [29], whose pseudo-code is given in Algorithm 2. This heuristic starts with an empty target set $S = \emptyset$. Then, at each step of the solution construction process, it adds the node $v_i \in V \setminus \sigma(S)$ that maximizes $|\sigma(S \cup \{v_j\})|$ for all $v_j \in V \setminus \sigma(S)$. Note, however, that this version of the algorithm does not scale well with increasing network size, because the diffusion simulation process must be executed at each step for all $v_j \in V \setminus \sigma(S)$, which is very time consuming.

Therefore, we decided to change this heuristic in the following way; see also Algorithm 3. First, the diffusion simulations for choosing a new vertex at each construction step are simply replaced by choosing the vertex with the highest degree. This is done with the intuition that vertices with a high degree potentially allow for covering larger portions of the network than vertices with a low degree. Second, we take profit from our theoretical result described in Section 4 in order to avoid repetitive diffusion simulation for already covered parts of the network. We henceforth refer to this new heuristic as the maximum degree heuristic (Mdg); see Algorithm 3.

Algorithm 3 Maximum Degree Heuristic (Mdg)

Input: Graph $G = (V, G)$
1: $S := \emptyset$
2: $Cov := \emptyset$
3: **while** $Cov \neq V$ **do**
4: $\quad v_i := \text{argmax}_{v_j \in V \setminus Cov}\{deg(v_j)\}$
5: $\quad Cov := \sigma(Cov \cup \{v_i\})$
6: $\quad S := S \cup \{v_i\}$
7: **end while**

Note that Mdg can, first of all, be applied as a standalone greedy heuristic for the TSS problem. However, we will also use it for decoding individuals $\vec{w}$. This is done by replacing line 4 of Mdg with the following one:

$$v_i := \text{argmax}_{v_j \in V \setminus Cov}\{deg(v_j) \cdot w_j\}$$

In other words, the degree $deg(v_j)$ of each candidate node $v_j \in V \setminus Cov$ is multiplied by its weight value w_j from individual $\vec{w}$. In this context, note that the first individual of the initial population in case of $seed$ = `true` (in which all vertices v_i have a weight value of 0.5) is decoded into the solution of the Mdg heuristic.

Finally, a relevant consideration to be made is the heuristic bias used in our decoding mechanism. Since our algorithm is developed to be used in social networks, and under the assumption that most of them exhibit scale-free-like degree distributions [23], our mechanism rewards highly connected nodes in the networks. This provides a significant selection bias towards solutions containing the most connected nodes. As a collateral effect, this might mean that—when applied to graphs with different degree distributions—our algorithm might require longer than usual computing times in order for the evolutionary algorithm to compensate for the heuristic bias used by default.

6 EXPERIMENTAL EVALUATION

In this section we will describe the experimental evaluation of both Mdg and Brkga, in comparison to Mts (our implementation) from the literature. All algorithms were implemented in C++ and the experiments were performed on a cluster of machines with Intel® Xeon® CPU 5670 CPUs with 12 cores of 2.933 GHz and a minimum of 32 GB RAM.

Experimental setting. For the experiments we have chosen to apply all algorithms to 27 real-life social networks. Some of these graphs are included in the Stanford Network Analysis Project (SNAP) open access repository [30]. The chosen instances are a

Table 1: Brkga parameters, domains, and tuning results

Parameter	Description	Domain	Value
n_{ind}	population size	$\{5, \ldots, 50\}$	46
p_e	elite proportion	$[0.1, 0.25]$	0.24
p_m	mutant proportion	$[0.1, 0.3]$	0.13
$prob_{elite}$	elite inheritance prob.	$[0.51, 0.8]$	0.69
$seed$	init. population seeding	{`true`, `false`}	`true`

widely used in the context of combinatorial optimization problems on graphs, and thus are of interest. They combine small, medium and large-scale networks, allowing to study the scale implications on the performance of the algorithms.

Although all three algorithms can be applied with any given threshold function θ, we have opted for selecting a fixed threshold for all the networks: $\theta(v) = \left\lceil \frac{deg(v)}{2} \right\rceil$ for all $v \in V$. This decision was motivated by two factors: first, our decision was motivated by the assumptions made in the context of other problems around the behavioral interactions of individuals within their communities, such as the Minimum Positive Influence Dominating Set Problem [51]. Second, fixed values of θ were studied in other works proposing algorithm for the same problem [15]. In fact, the setting $\theta(v) = \left\lceil \frac{deg(v)}{2} \right\rceil$ for all $v \in V$ has been adopted in several other works such as, for example [21].

Finally, note that the following computation time limit was applied for all runs of Brkga: $\max\left\{100, \frac{|V|}{100}\right\}$ CPU seconds. In this way, the computation time limit depends on the graph size, that is, more computation time is allowed for the application to larger graphs. On the other side, a minimum of 100 CPU seconds is allowed for each graph.

Parameter tuning. We made use of the scientific tuning software `irace` [33] for obtaining well-working parameter values for Brkga. In particular, `irace` was applied with a budget of 2000 algorithm runs, a real number precision of two positions behind the comma, and with tuning instances `graph_CA-AstroPh`, `socfb-Mich67` and `Amazon0302`. The obtained parameter values, which are used for the final experimental evaluation, are provided in Table 1.

6.1 Results

Table 2 provides the obtained numerical results in the following format. The first three table columns provide the instance name as well as the number of vertices and the number of edges of the respective network. Next, the results of Mts and of our own greedy algorithm Mdg are provided by means of two table columns for each algorithm. The first one (with heading 'Result') provides the target set size of the generated solutions, while the second one (with heading 'Time') shows the required computation time (in seconds). Furthermore, the results of Brkga are provided by means of three table columns. The first one (heading 'Best') indicates the best result obtained within 10 independent runs, while the second one (heading 'Average') provides the average of the best solutions obtained in 10 runs. Finally, the last column regarding Brkga shows the average time at which the best solutions of each run were found. The last three columns present the improvement (in percent) of the

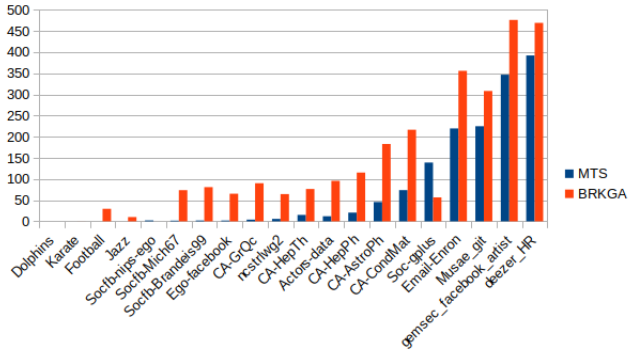
Table 2: Numerical results for 27 social networks

Network			Mts		Mdg		Brkga			Improvement (%)		
	Vertices	Edges	Result	Time	Result	Time	Best	Average	Avg. Time	Brkga	Mdg	Brkga (avg)
Karate	62	159	7	< 0.01	8	< 0.01	**6**	6.0	< 0.01	14.28	-14.28	14.28
Football	115	613	29	< 0.01	31	< 0.01	**22**	22.3	29.6	24.13	-6.89	23.10
Dolphins	34	78	**3**	< 0.01	**3**	< 0.01	**3**	3.0	< 0.01	0	0	0
Jazz	198	2742	33	< 0.01	31	< 0.01	**20**	20.4	10.0	39.39	6.06	38.18
CA-AstroPh	18772	198050	2228	45.54	1638	0.05	**1500**	1508.6	182.7	32.67	26.48	32.28
CA-GrQc	5242	14484	992	3.98	1033	0.01	**942**	947.4	90.0	5.04	-4.13	4.49
CA-HepPh	12008	118489	1832	20.5	1529	0.03	**1394**	1402.6	115.2	23.9	16.53	23.43
CA-HepTh	9877	25973	1388	15.1	1388	0.01	**1307**	1312.	76.4	5.83	0	4.04
CA-CondMat	23133	93439	3156	73.76	2938	0.05	**2760**	2777.6	216.4	12.54	6.90	11.98
Email-Enron	36692	183831	3143	219.48	2881	0.1	**2745**	2759.9	355.6	12.66	8.33	12.18
ncstrlwg2	6396	15872	1080	5.7	1108	0.01	**1027**	1045.7	64.3	4.9	-2.59	3.17
actors-data	10042	145682	1250	11.93	1014	0.02	**937**	943.4	95.6	25.04	2.88	24.52
ego-facebook	4039	88234	580	1.87	528	0.01	**493**	501.1	65.2	15.00	8.96	13.60
socfb-Brandeis99	3898	137567	421	1.94	395	0.02	**338**	358.0	80.8	19.71	6.17	14.96
socfb-nips-ego	2888	2981	**10**	2.18	**10**	< 0.01	**10**	10.0	< 0.01	0	0	0
socfb-Mich67	3748	81903	330	1.58	202	0.01	**168**	172.2	73.6	49.09	38.78	47.81
soc-gplus	23628	39194	76	138.82	69	0.01	**62**	62.9	56.5	18.42	9.21	17.23
musae_git	37700	289003	574	224.66	196	0.08	**176**	182.7	308.0	69.33	65.85	68.17
loc-gowalla_edges	196591	950327	9486	6792.23	5670	0.7	**5546**	5580.5	1835.3	41.53	40.22	41.17
gemsec_facebook_artist	50515	819090	1956	346.38	790	0.2	**702**	716.0	475.9	64.11	59.61	63.39
deezer_HR	54573	498202	3816	391.93	2346	0.1	**2223**	2252.9	468.9	41.74	38.52	40.96
com-youtube	1134890	2987624	n/a	n/a	39090	12.45	**39023**	39037.3	10897.5	n/a	n/a	n/a
com-dblp	317080	1049866	38148	16828.5	37197	4.1	**37017**	37056.2	2970.3	2.96	2.49	2.86
Amazon0302	262111	899792	**35289**	7939.59	35766	2.4	35685	35717.1	2148.5	-1.12	-1.35	-1.21
Amazon0312	400727	2349869	36479	18201.4	31165	3.5	**31085**	31096.8	3263.4	14.78	14.56	14.75
Amazon0505	410236	2439437	37198	19086.7	31926	3.7	**31842**	31857.7	3702.5	14.39	14.17	14.35
Amazon0601	403394	2443408	37177	18287.2	31507	3.5	**31455**	31475.2	3558.5	15.39	15.25	15.33

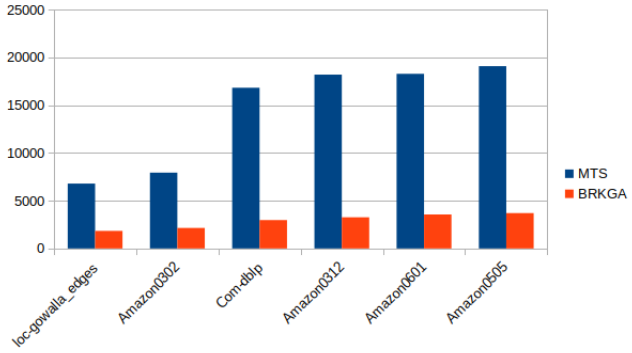
results of Brkga,Mdg and the average results of Brkga over the results of Mts.

The following observations can be made. First, Brkga is clearly the best-performing algorithm, obtaining the best results for all problem instances, except for one. It outperforms Mts, on average, by 21.75%. Moreover, Brkga seems to scale much better with growing network size than Mts. This is graphically shown by means of a computation time comparison in Figure 2. In fact, we stopped the execution of Mts for instance com-youtube (the largest of the considered instances) after four days of computation without any results. This is indicated by 'n/a' in Table 2. Second, Mdg performs not only much better than Mts—with an average improvement over Mts of 13.52%—but does so in a significantly shorter computation time. More specifically, Mdg is between two and three orders of magnitude faster than Mts. In fact, the success of Mdg is the basis for the success of Brkga, as Brkga makes use of the functionality and the greedy bias of Mdg in the procedure for decoding individuals. In Figure 3 we show the evolution of the solutions found by Brkga over time for four exemplary instances. Note that the dashed horizontal lines indicate the result of Mts. Moreover, the first solutions found by Brkga correspond to the solutions of Mdg, as the initial population of each run is seeded with the result of Mdg. Instance socfb-Brandeis99 (see Figure 3a) represents the standard case in which the Mdg-solution is already better than the Mts-solution. Moreover, Brkga is able to improve significantly over the Mdg solution. Next, instances CA-GrQc and ncstrlwg2 are examples for those cases in which the Mdg-solution is worse than the Mts-solution. The graphs, however, are still small enough for Brkga to be able to quickly improve over the Mts-solution. In fact, solutions of the quality of the Mts-solution are found after a few seconds. Finally, the last example shows the case of instance Amazon0302, the only case in which Mts outperforms Brkga. In this case, the Mdg-solution is significantly worse than the Mts-solution. Moreover, the instance is so large that Brkga is not able to improve a lot over the Mdg-solution. In fact, this is a weakness of our algorithm (in the context of very large problem instances) which is certainly subject to future work.

As mentioned before, in five out of 27 cases the result of Mdg is worse than the result of Mts, even though in most of the remaining cases Mdg significantly outperforms Mts. In order to study possible reasons for this, we first examined the degree distributions of all 27 networks. However, we were not able to find any apparent differences between, for example, the degree distribution of Amazon0302 and the degree distributions of the remaining 26 networks. Therefore, we decided to study the degree distribution of the nodes in the respective solutions of Mts and Brkga (see Figure 4). The plots show a blue dot for each possible degree (0 to $|V|-1$) of a given node in a solution, with the vertical axis displaying the logarithm of the amount of vertices of a given degree and the horizontal axis the logarithm of the mentioned degree. More specifically, the plots



(a) Small and medium sized instances

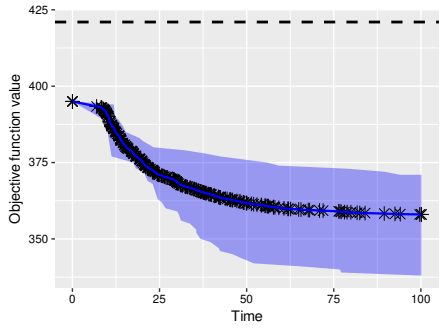


(b) Large instances

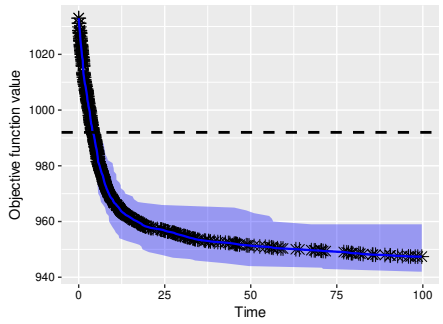
Figure 2: Plots displaying the (average) computation times of MTS and BRKGA, with instances ordered increasingly by number of nodes.

show the collection $\{(log_{10}(1+n), log_{10}(1+|V_n|))\}_{n=0}^{n=|V|-1}$ where V_n is the set of all the vertices in a solution with degree n. This specific type of plot aims to better display the behavior of the degree distributions on power-law graphs [2], such as social networks, which are characterized by having the property $P(deg(v) = k) \sim k^{-\lambda}$, that is, the probability of a given node having a degree k is proportional to $k^{-\lambda}$. By plotting in a double-logarithmic setting, the similarity of the degree distribution to the ideal power-law can be easily visualized as a straight line. Studying the graphics from Figure 4 it can be noticed that, just like the graphs themselves, the solutions also show power-law characteristics, with a small factor of variability discussed below that might explain some performance discrepancies between MTS and BRKGA.

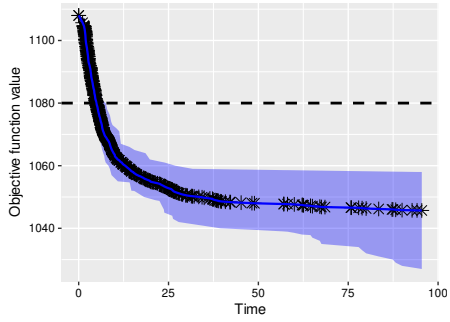
A significant pattern can be observed on the left end of the mentioned plots, clearly differentiating between the solutions of BRKGA and MTS. While in both cases the plots display a significant portion of the degree distribution corresponding with a decreasing straight line, the distribution at the left end of the graphics differs. While the BRKGA-solutions consistently show a positive increase in the number of nodes of a given degree as the degree grows (until a critical point) the MTS-solutions show a more stable distribution, with an almost horizontal line before the principal



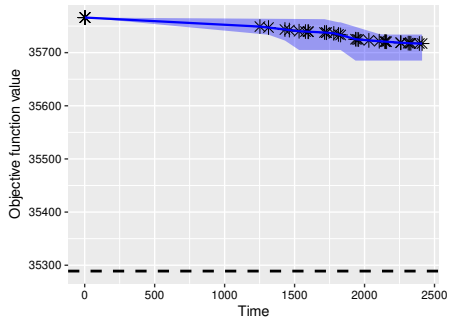
(a) Instance socfb-Brandeis99



(b) Instance CA-GrQc

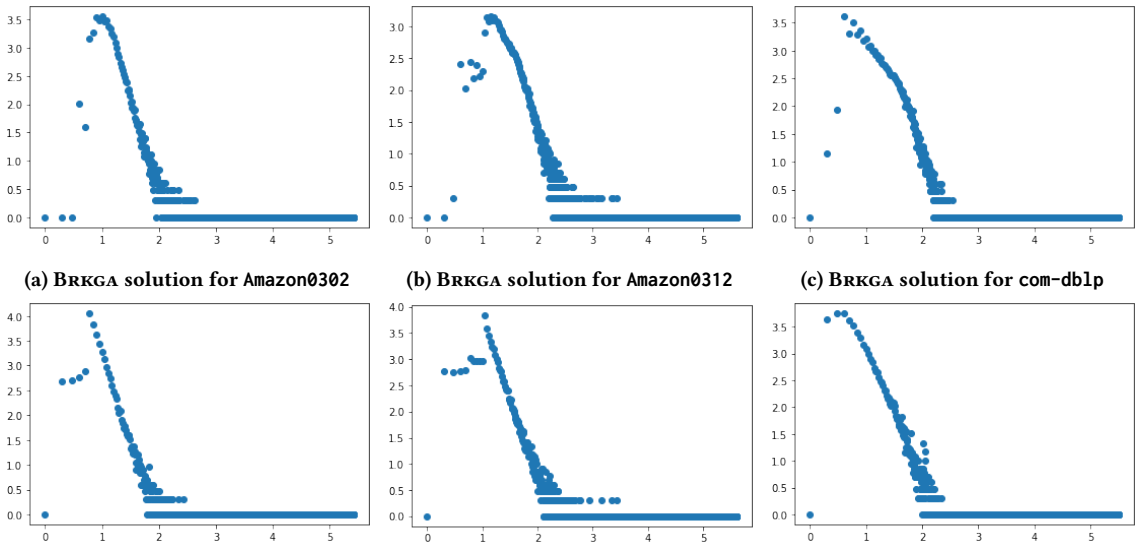


(c) Instance ncstrlwg2



(d) Instance Amazon0302

Figure 3: Evolution of the solutions found over time. The performance of MTS is indicated as a horizontal, dashed line.



(a) BRKGA solution for Amazon0302 (b) BRKGA solution for Amazon0312 (c) BRKGA solution for com-dblp

(d) MTS solution for Amazon0302 (e) MTS solution for Amazon0312 (f) MTS solution for com-dblp

Figure 4: Degree distributions of solutions obtained by BRKGA and MTS

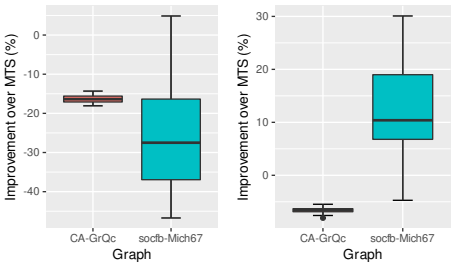


Figure 5: Results with random threshold functions. The boxplots show the improvement (in percent) of MDG over MTS (left), respectively BRKGA over MTS (right).

straight descent starts. In other words, this pattern indicates a difference in the way in which nodes with small degrees are included in the solutions. Nevertheless, this difference in pattern is shown both for instances in which BRKGA outperforms MTS (instances Amazon0302 and com-dblp) and for the instances where this is not the case (Amazon0312). Therefore, we conclude that, while most of the times the way in which BRKGA (and MDG) include low-degree nodes into a solution is very beneficial, for a few exceptions this is not the case. Nonetheless, further research is needed to explain this discrepancy in the results.

Finally, we conducted prelininary experiments concerning random threshold functions, that is, threshold functions $\theta : V \rightarrow \mathbb{N}$ s.t. $\theta(v)$ is chosen for each $v \in V$ randomly from $\{1, \ldots, |N(v)|\}$. We generated 20 threshold functions for two of the graphs: CA-GrQc and socfb-Mich67. The results are shown in terms of boxplots in Figure 5. They show that, first, MDG is clearly worse than MTS when random threshold functions are concerned. Nevertheless, even with this unfavorable heuristic bias BRKGA is able to outperform MTS for graph socfb-Mich67. Unfortunately, this is not the case for graph CA-GrQc where BRKGA is about 7% worse than MTS. Therefore, we conclude that the heuristic bias of BRKGA should be changed when random threshold functions are considered.

7 CONCLUSIONS

In this paper we have tackled a combinatorial optimization problem from the context of viral marketing that has, so far, been largely ignored by the metaheuristics community. Existing methods for the target set selection problem mainly focus on heuristics or on integer linear programming models. In contrast, we proposed a biased random key genetic algorithm for solving this problem under the linear threshold diffusion model. Our algorithm is based on an efficient method for decoding random keys (individuals) into solutions to the problem. This method is based on a new theoretical result that allows for an acceleration of the diffusion process when adding new vertices to partial solutions. The obtained results show that our biased random key genetic algorithm outperforms a state-of-the-art heuristic for the target set selection problem. Moreover, even our new heuristic, which is at the basis of our solution decoding mechanism, outperforms the approach from the literature.

Nevertheless, our algorithm shows difficulties in the context of very large networks. The improvement of our approach for the application such networks is one of the avenues for future research. Another one concerns the combination of our heuristic techniques with machine learning approaches that might be used to accelerate the diffusion simulation process for the evaluation of solutions.

ACKNOWLEDGMENTS

This work was supported by grant PID2019-104156GB-I00 funded by MCIN/AEI/10.13039/501100011033, and by project 202150E087 (Metaheurísticas Híbridas) funded by CSIC.

REFERENCES

[1] Eyal Ackerman, Oren Ben-Zwi, and Guy Wolfovitz. 2010. Combinatorial model and bounds for target set selection. *Theoretical Computer Science* 411, 44-46 (2010), 4017–4022.

[2] Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (1999), 509–512. https://doi.org/10.1126/science.286.5439.509 arXiv:https://www.science.org/doi/pdf/10.1126/science.286.5439.509

[3] Cristina Bazgan, Morgan Chopin, André Nichterlein, and Florian Sikora. 2014. Parameterized inapproximability of target set selection and generalizations. *Computability* 3, 2 (2014), 135–145.

[4] Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. 2011. Treewidth governs the complexity of target set selection. *Discrete Optimization* 8, 1 (2011), 87–96.

[5] Ivan Bliznets and Danil Sagunov. 2018. Solving Target Set Selection with Bounded Thresholds Faster than 2^n. *arXiv preprint arXiv:1807.10789* (2018).

[6] Moses Charikar, Yonatan Naamad, and Anthony Wirth. 2016. On approximating target set selection. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[7] Moses Charikar, Yonatan Naamad, and Anthony Wirth. 2016. On Approximating Target Set Selection. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 60)*, Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 4:1–4:16. https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2016.4

[8] Ning Chen. 2009. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics* 23, 3 (2009), 1400–1415.

[9] Wei Chen, Chi Wang, and Yajun Wang. 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*. 1029–1038.

[10] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 199–208.

[11] Chun-Ying Chiang, Liang-Hao Huang, and Hong-Gwa Yeh. 2013. Target set selection problem for honeycomb networks. *SIAM Journal on Discrete Mathematics* 27, 1 (2013), 310–328.

[12] Morgan Chopin, André Nichterlein, Rolf Niedermeier, and Mathias Weller. 2014. Constant thresholds can make target set selection tractable. *Theory of Computing Systems* 55, 1 (2014), 61–83.

[13] Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, Martin Milanič, and Ugo Vaccaro. 2014. Latency-bounded target set selection in social networks. *Theoretical Computer Science* 535 (2014), 1–15.

[14] Gennaro Cordasco, Luisa Gargano, Marco Mecchia, Adele A Rescigno, and Ugo Vaccaro. 2015. A fast and effective heuristic for discovering small target sets in social networks. In *Combinatorial Optimization and Applications*. Springer, 193–208.

[15] Gennaro Cordasco, Luisa Gargano, and Adele Anna Rescigno. 2015. Influence Propagation over Large Scale Social Networks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015* (Paris, France) *(ASONAM '15)*. Association for Computing Machinery, New York, NY, USA, 1531–1538. https://doi.org/10.1145/2808797.2808888

[16] Gennaro Cordasco, Luisa Gargano, and Adele A Rescigno. 2016. On finding small sets that influence large networks. *Social Network Analysis and Mining* 6, 1 (2016), 1–20.

[17] Guillem Rodríguez Corominas, Christian Blum, and Maria J. Blesa. 2019. A biased random key genetic algorithm for the weighted independent domination problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, Manuel López-Ibáñez, Anne Auger, and Thomas Stützle (Eds.). ACM, 2052–2055.

[18] Zoltán Dezső and Albert-László Barabási. 2002. Halting viruses in scale-free networks. *Physical Review E* 65, 5 (2002), 055103.

[19] Stefan Ehard and Dieter Rautenbach. 2019. On some tractable and hard instances for partial incentives and target set selection. *Discrete Optimization* 34 (2019), 100547.

[20] Stefan Ehard and Dieter Rautenbach. 2019. On some tractable and hard instances for partial incentives and target set selection. *Discrete Optimization* 34 (2019), 100547. https://doi.org/10.1016/j.disopt.2019.05.004

[21] Paola Flocchini, Rastislav Královič, Peter Ružička, Alessandro Roncato, and Nicola Santoro. 2003. On time versus size for monotone dynamic monopolies in regular topologies. *Journal of Discrete Algorithms* 1, 2 (2003), 129–150.

[22] Angela K Fournier, Erin Hall, Patricia Ricke, and Brittany Storey. 2013. Alcohol and the social network: Online social networking sites and college students' perceived drinking norms. *Psychology of Popular Media Culture* 2, 2 (2013), 86.

[23] Piotr Fronczak. 2018. *Scale-Free Nature of Social Networks*. Springer New York, New York, NY, 2300–2309. https://doi.org/10.1007/978-1-4939-7131-2_248

[24] Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review* 9, 3 (2001), 1–18.

[25] José Fernando Gonçalves and Mauricio GC Resende. 2011. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics* 17, 5 (2011), 487–525.

[26] Mark Granovetter. 1978. Threshold models of collective behavior. *American journal of sociology* 83, 6 (1978), 1420–1443.

[27] Tim A Hartmann. 2018. Target set selection parameterized by clique-width and maximum threshold. In *International Conference on Current Trends in Theory and Practice of Informatics*. Springer, 137–149.

[28] E. Muller J.Goldenberg, B.Libai. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters* 12, 3 (2001), 211–223.

[29] David Kempe, Jon Kleinberg, and Éva Tardos. 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* (Washington, D.C.) *(KDD '03)*. Association for Computing Machinery, New York, NY, USA, 137–146. https://doi.org/10.1145/956750.956769

[30] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[31] Xianliang Liu, Zishen Yang, and Wei Wang. 2016. Exact solutions for Latency-Bounded Target Set Selection Problem on some special families of graphs. *Discrete Applied Mathematics* 203 (2016), 111–116.

[32] Cheng Long and Raymond Chi-Wing Wong. 2011. Minimizing Seed Set for Viral Marketing. In *2011 IEEE 11th International Conference on Data Mining*. 427–436. https://doi.org/10.1109/ICDM.2011.99

[33] Manuel López-Ibáñez, Jeremie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. 2016. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3 (2016), 43–58.

[34] Vijay Mahajan, Eitan Muller, and Frank M Bass. 1990. New product diffusion models in marketing: A review and directions for research. *Journal of marketing* 54, 1 (1990), 1–26.

[35] Rafael T Mikolajczyk and Mirjam Kretzschmar. 2008. Collecting social contact data in the context of disease transmission: prospective and retrospective study designs. *Social Networks* 30, 2 (2008), 127–135.

[36] Stanley Milgram. 1967. The small world problem. *Psychology today* 2, 1 (1967), 60–67.

[37] André Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller. 2013. On tractable cases of target set selection. *Social Network Analysis and Mining* 3, 2 (2013), 233–256.

[38] André Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller. 2013. On tractable cases of target set selection. *Social Network Analysis and Mining* 3, 2 (2013), 233–256.

[39] Rodrigo Olivares, Francisco Muñoz, and Fabián Riquelme. 2021. A multi-objective linear threshold influence spread model solved by swarm intelligence-based methods. *Knowledge-Based Systems* 212 (2021), 106623.

[40] Joel Penney. 2016. Motivations for participating in 'viral politics' A qualitative case study of Twitter users and the 2012 US presidential election. *Convergence* 22, 1 (2016), 71–87.

[41] Bruno Q. Pinto, Celso C. Ribeiro, Isabel Rosseti, and Alexandre Plastino. 2018. A biased random-key genetic algorithm for the maximum quasi-clique problem. *European Journal of Operational Research* 271, 3 (2018), 849–865.

[42] Santiago V. Ravelo, Cláudio N. Meneses, and Eduardo A.J. Anacleto. 2020. NP-hardness and evolutionary algorithm over new formulation for a Target Set Selection problem. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. https://doi.org/10.1109/CEC48606.2020.9185558

[43] TV Thirumala Reddy, D Sai Krishna, and C Pandu Rangan. 2010. Variants of spreading messages. In *International Workshop on Algorithms and Computation*. Springer, 240–251.

[44] Jean-Baptiste Rival and Joey Walach. 2009. The use of viral marketing in politics: a case study of the 2007 French presidential election.

[45] WS Rossi. 2015. *Analysis and control of the Linear Threshold Model of cascades in large-scale networks. A local mean-field approach*. Ph. D. Dissertation. Ph. D. thesis, Politecnico di Torino, 2015. Available online at http://porto ….

[46] Efrain Ruiz, Maria Albareda-Sambola, Elena Fernández, and Mauricio G.C. Resende. 2015. A biased random-key genetic algorithm for the capacitated minimum spanning tree problem. *Computers & Operations Research* 57 (2015), 95–108.

[47] Thomas C Schelling. 2006. *Micromotives and macrobehavior*. WW Norton & Company.

[48] Paulo Shakarian, Abhivav Bhatnagar, Ashkan Aleali, Elham Shaabani, and Ruocheng Guo. 2015. *Diffusion in social networks*. Springer.

[49] Paulo Shakarian, Sean Eyre, and Damon Paulo. 2013. A scalable heuristic for viral marketing under the tipping model. *Social Network Analysis and Mining* 3, 4 (2013), 1225–1248.

[50] Cheng Wang, Lili Deng, Gengui Zhou, and Meixian Jiang. 2014. A global optimization algorithm for target set selection problems. *Information Sciences* 267

(2014), 101–118. https://doi.org/10.1016/j.ins.2013.09.033

[51] Feng Wang, Erika Camacho, and Kuai Xu. 2009. Positive Influence Dominating Set in Online Social Networks. In *Proceedings of the 3rd International Conference on Combinatorial Optimization and Applications* (Huangshan, China) *(COCOA '09)*. Springer-Verlag, Berlin, Heidelberg, 313–321. https://doi.org/10.1007/978-3-642-02026-1_29