This is a pre-copyedited, author-produced version of an article accepted for publication in **Logic Journal of the IGPL** following peer review. The version of record J. David Nuñez-Gonzalez, Borja Ayerdi, Manuel Graña, Michał Wozniak, *A new heuristic for influence maximization in social networks*, **Logic Journal of the IGPL**, Volume 24, Issue 6, December 2016, Pages 996–1014, is available online at: https://doi.org/10.1093/jigpal/jzw048

A new heuristic for influence maximization in social networks

J. DAVID NUNEZ-GONZALEZ (1), BORJA AYERDI (1), MANUEL GRAÑA (1), MICHAL WOZNIAK (2)

(1) University of Basque Country, (2) Wroclaw University of Technology

Abstract

Influence Maximization (IM) is defined as the problem of finding the minimal IM-seed set of nodes maximally influential in a network. IM solution is formulated in the context of an influence spread model describing how the influence is propagated through the network. IM is relevant for applications such as viral marketing, and the analysis of infection diffusion in a community. Such communities are described by graphs model which have some kind of probabilistic description of how influence is propagated from one node to its neighbours. The cascade and threshold propagation models are the most popular in the literature. In this article, a new global heuristic search method for IM is proposed. We provide comparison over a collection of synthetic and real life graphs against other state-of-the-art heuristic search methods, namely Simulated Annealing, Genetic Algorithms, Harmony Search and the classical Greedy Search (GS) algorithm. Our new method (IMH) competes with the GS algorithm getting the minimal IM-seed set whose influence spreads the largest amount of nodes. Our method improves Greedy algorithm's time execution.

1 Introduction

The analysis of influence propagation through social media started from the consideration of phenomena such as mobs, riots or strikes [18] as pure physical phenomena, stripped out of psychological considerations. That is, the quantitative model considers that individual decisions are taken as the fruit of social pressure defined by social interactions. The same model applies to propagation of innovations, rumours and advertising [14], so the topic become naturally part of the marketing research area. The research question was to determine the appropriate balance between marketing efforts and word-of-mouth propagation through personal social networks defined by strong and weak links.

Cellular automata formalism allowed to build computational models to explore such questions. The two basic spread models of influence propagation are the Independent Cascade model (ICM) [14] and the Linear Threshold model (LTM) [18].

Social networks are represented by a weighted directed graph $G(V,E,W)$ where nodes $v \in V$ represent individuals of the community, edges $(v,v') \in E$ represent social relationships between them, and W are the weights of either nodes or edges. In the LTM diffusion model, nodes are weighted by a decision threshold $wv \in R$, while in the ICM the weights are placed on the edges, they are propagation probabilities. Nodes can be active or inactive, i.e. they have been influenced or not. When a node becomes active it is possible to spread influence to an inactive node from its neighbourhood, and the influence propagation is modelled by an iterative process. In the LTM propagation model, a node becomes active when the percentage of active neighbours is above the threshold, i.e. $1/|V| \text{ sum}=v av,v\_\sigma v\_ (S) \geq wv$, where $av,v\_ \in [0,1]$ is the entry in the adjacency matrix A such that $av,v' = 1$ iff $(v,v') \in E$, and $\sigma v (S)$ is an indicator function that values 1 iff node v belongs to the influence spread of a IM-seed set S. When a node v becomes active it is added to the actual influence spread, i.e., $\sigma (S) \leftarrow \sigma (S) \cup \{v\}$. In the ICM, the weights $wv,v' \in R$ are measures of the strength of the relation, i.e a probabilistic measure of the influence capability of one node over another. Obviously, we have $wv,v' = 0$ iff $av,v' = 0$. Each node activates its neighbours by

carrying out a stochastic decision by Monte Carlo sampling the Bernoulli distribution defined by edge probability $w_{v,v'}$. A variation of the LTM allows an inactive to become active when the summation of influence degrees on the incoming links is greater than the node threshold. There are works such as [30] analysing the computational complexity of influence maximization problem in the deterministic LTM. In both ICM and LTM, there is no reversibility of states so that the influence propagation process will end when no more nodes may become active. In the literature, ICM and LTM propagation models are usually applied separately but some works subsume both models [37], in some cases new evidences such as trust are added to the propagation model [32]. In other words, we do not consider viral propagation models [39] which contemplate *infection* and *recovery* of individuals. Study of such models by mean field analysis [29] show that it is possible to determine the diffusion rate that ensures that the system reaches a steady state where the infection persists. Specifically, in this article we use ICM as the propagation model in the experiments.

1.1 Influence maximization

Influence Maximization (IM) [5, 22] is stated as the problem of finding the minimal subset of influential nodes (IM-seed nodes) with maximal influence, i.e. that affect the largest number of nodes in the network, where influence is computed by propagation in the network according to a spread model. The IM-seed nodes are the initially active nodes spreading their influence according to any of the propagation models discussed above. The most straightforward application is in marketing, when a new product hits the market, a company may want to select the smallest group of (most influential) seed customers to provide them the product for free in order to boost its popularity by propagating it in their social network by word-of-mouth [14]. Additionally, IM has been applied to design negotiation strategies addressing persuasion to the most influent agents [33], and to worm propagation containment in ad networks of smartphones [38].

1.2 Article contribution

IM under both ICM and LTM propagation models is a NP-complete problem [22]. Therefore, exhaustive search and exact solutions are unfeasible in general, and research efforts are driven towards the proposition of efficient heuristics. The greedy selection of nodes (i.e. selecting the one contributing the largest increase of influence spread) is ensured to reach a solution which is at least a 67% of the global optima, due to the submodular nature of the influence spread function [22]. This article formulates a new method (IMH) for the selection of the minimal IM-seed set of nodes producing the maximal influence, which is several orders of magnitude faster than GS but provides the same influence results.

1.3 Article organization

The rest of the article is organized as follows: Section 2 presents some related works of the State of the Art. Section 3 presents the computational methods applied. Section 4 presents some experimental results. Section 5 concludes this article.

2 Related works

IM was proven to have NP-complete computational complexity in [22], for both LTM and ICM propagation models. In fact, the computational cost of spreading influence is #P hard, while the combinatorial search for the mini-max set of nodes in equivalent to NP hard problems. They also show that the greedy search solution is guaranteed to be at worst within $(1-1/e)$% of the optima for these propagation models, on the basis of previous results for submodular functions. In general, the estimation of the influence $\sigma(S)$ of IM-seed node set $S \subset V$ must be carried out by simulation,i.e. repeating the random process of influence propagation a number of times. Influence $\sigma(S)$ was proven to be a submodular function for both LTM and ICM propagation models, as well as for their generalizations. The critical computational load is, therefore, in the estimation of influence $\sigma(S)$, [19] propose a fast computation of $\sigma(S)$ that stores some of the previously computed influence sets, so that they do not need to be recomputed each time. Its disadvantage is the large memory requirements to store the precomputed influences. Another approach to reduce the time complexity of influence computation is a a divide and conquer method [43] applied to IM on large-scale mobile social networks in two steps. First, the large-

scale social network is divided into communities selected according to information diffusion, assuming ICM propagation model. Second communities are selected to look for influential nodes by dynamic programming. Further, a model of parallel computation of the influence spread in each community is proposed. Similar community decomposition is proposed in [40] where IM-seed nodes are then selected from the communities. Another kind of acceleration is preprocessing the graph to obtain the spread trees which allow efficient computation of influence probabilities. This approach makes [27] in the context of targeted IM, where some nodes are the target of the viral marketing, while others are susceptible or immune. Dealing with immune nodes requires some care, but it is not a source of computational complexity. Besides, graph communities are useful to reduce influence computation problem complexity also in [27]. The greedy algorithm has quadratic complexity on the number of nodes (which can be large in real life social networks). There have been a number of optimizations trying to reduce the cost of spread computation. The Cost-Effective Lazy Forward selection) (CELF) method has been proposed [28], which consists in maintaining an ordered table of nodes and their marginal gain, so that candidates to be included in the IM-seed set are taken from the top of this list. An enhancement to CELF, the CELF++ [15] fully exploits influence spread function submodularity. Further, the Simpath algorithm [16] is based on the idea that under LTM propagation it is possible to compute an estimation of the spread by enumeration of the simple paths emanating of a node. Another optimization comes by the hand of a new propagation model called *credit distribution* [17], which avoids Monte Carlo simulation to achieve estimation of the spread on the basis of propagation traces. Another optimization considers the number of simple paths departing from a node as the indicator of spread potential [9] (ASIM), achieving a scalable algorithm for IM under the ICM.

A different approach is the use of heuristics, such as Ant Colony Optimization (ACO) [6], to search for an almost optimal solution. One approach maps the IM problem into the problem of finding a cycle of prescribed length with maximum influence spread. ACO are well suited to find cycles in graphs, however the approach only contemplates the selection of IM-seed nodes, it does not reduce the complexity due to influence spread computation. Another heuristic search tested is Simulated Annealing (SA) [21], where the minimal set solution was trivially encoded as a binary vector and the influence spread was computed by Markov random simulation.

To avoid the computation of the influence spread, some authors use node features such as betweenness, diversity of community belonging, or k-shell decomposition value as indirect measures of influence [7]. In this same line [41] proposes 'supermediators' using as indirect measure of influence the information spread, which decreases if the supermediators are removed from the network.

3 Computational methods

The straightforward approach is to perform an exhaustive search, where all possible combinations of IM-seed node sets are evaluated. In a systematic procedure, a search tree would be traveled where each node correspond to a solution. For NP-complete problems, such as IM, this and other exact global search methods are unfeasible. Therefore, we must resort to heuristic approaches, which provide suboptimal but good results in a reasonable time, such as the greedy search [22], which has quadratic complexity on the number of nodes. In this article, we explore a new application method comparing its results against other well-known heuristic algorithms: Simulated Annealing (SA), Genetic Algorithm (GA), Harmony Search (HS) and Greedy Search (GS) algorithm. In this section we provide an overview of these computational methods. All methods have been programmed and run in Matlab. The objective function to maximize is the influence spread, i.e. max S $\{\sigma(S)\}$. For a given candidate solution, $\sigma(S)$ is computed by a ICM Monte Carlo approximation specified in the introduction section. Often the heuristic approaches need some codification of the candidate solution. The common codification is a vector of binary valued components such that Sv=1 iff node v has been included in the IM-seed solution S. For convenience, the components are not explicit in most algorithm presentations. Note that the vector space dimension is the size of the node set V.

3.1 Simulated annealing

Simulated Annealing (SA) was proposed [23, 24, 36] as a general purpose probabilistic metaheuristic for the global optimization of non-convex functions (often non-differentiable) in large search spaces.

It is a nature inspired technique, mimicking the heating and cooling process followed to obtain some materials, for example high quality steel. SA was shown to provide the global optima under very strict conditions, and provides good approximations in a reasonable computational time. It generates a sequence of solutions whose objective function values converge to the global optimum value.

A temperature parameter allows to control the search. The temperature parameter typically starts off high and is slowly 'cooled' or lowered in every iteration. At high temperatures, the process accepts state (solution) changes that deteriorate the objective function to a limited extent. This prevents the search from getting trapped in local optima at early stages. At decreasing temperatures, it becomes a hill climbing algorithm that only accepts improvements of the objective function.

Algorithm 1 Simulated Annealing algorithm pseudocode
1. $s=s_0$
2. For k=0 : k_{max} (exclusive):
(a) $T \leftarrow temperature(k/k_{max})$
(b) Pick a random neighbor, $S_{new} \leftarrow neighbour(S)$
(c) If $P_a(\sigma S, \sigma(S_{new}), T) > r$, $r \sim U(0,1)$
(d) $S \leftarrow S_{new}$
3. Return s

Algorithm 1 presents a pseudocode of the Simulated Annealing, where S is the current solution, which is a binary codification of the nodes, one bit per node which is on if the node belongs to the IM-seed influence set. *Snew* is a candidate new solution generated by the *neighbour*() function from the current solution, by randomly changing one of the components to its opposite value. The acceptance probability $Pa(E(S), E(Snew), T)$ is a function of the temperature T and the difference of the objective function that is the influence spread of the IM-seed set $\sigma(S)$ of the current and new candidate states.

3.2 Genetic algorithm

Genetic Algorithms (GA) [13, 20] were proposed as a general method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm iteratively generates a population $\mathbf{S}t$ of individual candidate IM solutions Si by the application of genetic operators, crossover and mutation, to the chromosomes in the previous generation $\mathbf{S}t-1$. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Successive population generations 'evolve' toward an optimal solution.1 In GAs, the number of generations and the population size are critical parameters, in the experiments they are set proportional to problem complexity, i.e. number of nodes in the social network, so that effective computation time grows linearly.
A typical GA is as follows: Start with a randomly generated population $\mathbf{S}0=\{S1,...,SN\}$, each chromosome is a candidate solution encoded as a binary valued vector. Calculate the fitness $\sigma(Si)$ of each chromosome Si in the population St , the fitness allows to compute the selection probability. At each generation, create N offspring by crossover and mutation. For crossover, randomly draw a pair of parent chromosomes from the current population, according to an empirical probability distribution which is a increasing function of chromosome fitness. Selection is done 'with replacement', meaning that the same chromosome can be selected more than once to become a parent. According to crossover probability pc exchange the bits of the pair randomly to form two offsprings. If no crossover takes place, form two offspring that are exact copies of their respective parents. For mutation: mutate each bit the two offspring at each locus with probability pm (the mutation probability), and place the resulting chromosomes in the new population. Finally, replace the current population with the new population applying an elitist rule that preserves the best 20% chromosomes of the previous population.

3.3 Harmony search

Harmony Search (HS) [12] is a heuristic for global optimization of non-convex functions inspired in the musical improvisation process. It has been successfully applied to a variety of problems alone or hybridized with other methods. Some recent applications follow. The muzzle velocity of an electromagnetic railgun was optimized applying HS on the variables ranked by a previous orthogonal design method [3]. The optimization of an electrical transformer design [2] was achieved by a multiobjective HS endowed with crowding distance ranking and control parameter tuning by a Ricker map.

Parameter tuning by HS of the proportional integral controllers of a distributed power generation system overcomes genetic algorithms and gradient descent approaches in [1]. A differential HS compares favorably with particle filter approaches for face tracking in video imagery [8]. A quasioppositional HS improves over fuzzified internal control models [42, 44] for the control of several standard power generation systems. The design of ensemble classifiers using HS for classifier-asfeature selection is discussed in [4]. An improved HS is successfully applied or thrust optimization of dynamic positioning of off-shore oil drilling platforms [45]. The variety of these applications show the versatility of the HS approach.

Harmony Search (HS) [12, 26] is a global search heuristic algorithm inspired by the musical improvisation process proposed. In the HS algorithm, each musician (= decision variable) plays (=generates) a note (= a value) looking for the best harmony (= global optimum) all together.

Harmony is defined by some objective function which we try to optimize (minimize or maximize). Since its initial proposal, there have been variations and improvements of HS in the literature to be applied in different contexts, for instance: discrete design variables, and global optimization through competition [10, 11, 25, 31, 34]. the optimization problem is specified as follows:

$$_{I}\ \min_{x}\{f(\mathbf{x})|\mathbf{x}=[xi\in Xi;i=1,...,N]\}$$

where $f(\mathbf{x})$ is the objective function that corresponds to the musical harmony, Xi is the range set of design variable xi (we consider continuous design variables), and N is the number of design variables. The 'harmony memory'(HM) matrix, equation (2), is the central data structure of the algorithm containing the current state of the search, given by the preserved harmony vectors plus their harmony value $f(\mathbf{x})$,

$$HM=\begin{bmatrix}\mathbf{x}^{1} & \cdots & \mathbf{x}^{HMS}\\ f(\mathbf{x}^{1}) & \cdots & f(\mathbf{x}^{HMS})\end{bmatrix}$$

ordered so that $f(\mathbf{x}j)\geq f(\mathbf{x}j+1)$, so that $f(\mathbf{x}HMS)$ is the worst harmony value. Algorithm 2 shows a pseudo-code of the HS optimization procedure. In the first step problem data is read and algorithm parameters are initialized. First, we select the graph to be explored to solve IM. It can be a randomly generated graph or a real social network graph. Next, HS algorithm parameters controlling the optimization process are specified: the harmony memory size (HMS) specifying the number of solution vectors stored in the harmony memory, the harmony memory considering rate (HMCR) specifying if a variable improvisation is extracted from the memory, the pitch adjusting.

Algorithm 2 Harmony Search algorithm adapted to Influence Maximization

1. Given probabilistic social graph $G=(V,E,W)$
2. Initialize HS parameters and HM_0
3. while $t \leq NI$
   (a) for $i=1\ldots N$ //Improvise new harmony $\mathbf{x}'$,
      i. if $r \leq HMCR$
         A. $x_i' \overset{\in}{\leftarrow} \{x_i^1,\ldots,x_i^{HMS}\}$,
         B. if $r<PAR$ then $x_i' \leftarrow x_i'+\alpha_i$; $\alpha_i \sim U[-BW,BW]$;
      ii. otherwise $x_i' \overset{\in}{\leftarrow} X_i$
   (b) Evaluate harmony $f(\mathbf{x}')$
   (c) If $f(\mathbf{x}')>f(\mathbf{x}^{HMS})$
      i. replace $\mathbf{x}^{HMS}$ in HM, and sort HM.
4. Return best Harmony

rate (PAR), and termination criterion (maximum number of searches). In the next step the HM is initialized. Next we carry the improvisation of a new harmony. A new harmony vector, $\mathbf{x}'=(x'_1,\ldots,x'_N)$ is generated based on memory considerations, pitch adjustments, and randomization. With probability HMCR the value of the design variable x'_I is selected from the collection of values in the HM, i.e. $x'_i \in \leftarrow \{x^1_i,\ldots, x^{HMS}_i\}$ where $\in\leftarrow$ denotes random selection from a set of values. In Algorithm 2, $r\sim U(0,1)$ denotes a random number with uniform distribution in the interval (0,1). If $r>HMCR$, in other words with probability $1-HMCR$, the value of the variable is extracted from its range set Xi . The new value can be fine tuned with probability PAR after a positive test with HMCR. i.e. $x_i = v_{i,k\pm m}$ where $v_{i,k\pm m}$ is either the next value of the range set of a discrete variable or a random mutation in continuous variables. In some implementation, the first variable is always assigned a value from the history. If there is pitch adjustment for x_i , the pitch-adjusted value of xi (k) is $x_i \leftarrow x_i+\alpha_i$ where α_i is a sample of a random variable following a uniform distribution $U(-BW,NW)$, where BW is an arbitrary distance bandwidth for the continuous design variable. In the next step HM is updated.

If the new harmony vector is better than the worst harmony in the HM in terms of the objective function value, the new harmony is included in the HM and the existing worst harmony is excluded from the HM. The HM is then sorted by the objective function value.

Mapping IM into HS: For IM in social networks graphs, harmonies are binary vectors encoding the IM-seed set, i.e., a vector component value is 1 if the corresponding graph node belongs to the seed set, otherwise it is zero. IM is a multi-objective problem, because we want to achieve two goals: (i) maximize spread, and (ii) minimize IM-seed size. Given an harmony $\mathbf{x}$ and a probabilistic graph G_i, the evaluation returns:

$$f(\mathbf{x},G_i')=\sigma(S_\mathbf{x})+10^{-log_{10}V}*(V-S_\mathbf{x}),$$

where V is the set of nodes in the network, σ $(S\mathbf{x})$ is the number of nodes that have been visited through the spread model (Independent Cascade Model) and $S\mathbf{x}$ the number of actives nodes in the harmony $\mathbf{x}$. In this way, the harmony which visits the largest amount of nodes in the network with the minimum active nodes in the harmony will be reported as local optimum at the end of the computational process.

Algorithm 3 Greedy Search algorithm

1. Given weighted social graph $G=(V,E,W)$
2. $S_0=\varnothing$
3. Iterate until $\Delta\sigma(v^*)=0$
   (a) Compute $\Delta\sigma(v)=\{\sigma(S_t\cup\{v\})-\sigma(S_t)\}$ for all $v\in V-S_t$
   (b) $v^*=\arg\max_{v\in V}\{\Delta\sigma(v)\}$
   (c) $S_{t+1}=S_t\cup\{v^*\}$
   (d) $t=t+1$
4. Return S_t

3.4 Greedy Search

A straightforward greedy search (GS) algorithm achieves a good deterministic approximation to the optimum solution of IM due to non-negativity, monotonicity and submodularity of $\sigma(\cdot)$ [22].

The influence spread is a set function $\sigma:2^V\rightarrow R$, which is non-negative, i.e. $\sigma(S)\geq 0$ for all $S\subseteq V$, monotone, i.e. $\sigma(S)\leq\sigma(T)$ for all $S\subseteq T$, and submodular, i.e. $\sigma(S\cup\{v\})-\sigma(S)\geq\sigma(T\cup\{v\})-\sigma(T)$ for all $S\subseteq V$ and $v\in V$. For this kind of objective functions, it is shown that the greedy search would achieve a solution which is at least 63% of the global optimum [22].

There are two main steps in the GS specified. First, we compute the influence spread $\sigma(vi)$ for all nodes in the graph. Second, main step is solution generation specified in Algorithm 3. At the beginning the solution IM-seed node set S is empty. Then we iterate the greedy search, consisting in looking for the node $v*$ that provides the greatest increase in the influence $\_\sigma(v)$, until there is no increase in influence whatever the node chosen, i.e. $\_\sigma(v*)=0$. For the ease of notation, we assume that $\sigma(\varnothing)=0$.

3.5 New heuristic for IM

Our proposed new heuristic method starting step is to identify nodes with zero in-degree, i.e. no incoming edge ending to them, ($S0$ in step 2). The justification is that any IM-seed set whose influence covers all the graph must include them because they can not be influenced by any other node. The set of remaining nodes $R0$ consists of all nodes not in the initial solution $S0$ nor in its influence spread $\sigma(S0,Ab)$ computed using the base adjacency matrix. The set $R0$ contains all candidate nodes to enlarge the solution, because the removed nodes add nothing to the influence spread of the actual solution $S0$. Next, the adjacency matrix is simplified removing edges ending into or departing from nodes removed from $R0$, because these edges will not play any role in ensuing influence computations.

The algorithm proceeds by iterating the following steps until the set of remaining nodes Rt is empty. The first step is to find the node $v*$ with maximal influence in one step $\sigma1(\{v\},At)$, i.e. paths of length 1, using the simplified adjacency matrix At . The IM-seed solution is increased adding $v*$, while the set of remaining candidate nodes is decreased removing $v*$ and its one step influences. The adjacency matrix is updated accordingly. The heuristic of assuming that maximal one step influences would correspond to maximal influence spreads is a extreme form of greediness, but that appears to be effective from the experimental results. At the same time, removing only one step influences may leave candidate nodes which in fact do not improve the influence spread when added to the solution, however from the experimental results, it seems to have little effect. We will denote this heuristic as IMH in the following.

Algorithm 4 Proposed IM heuristic (IMH) algorithm

1. Given social graph $G=(V,E,W)$, with adjacency matrix $A_b=\left[a^b_{v,v'}\right]$
2. $S_0=\left\{v\middle|\sum_{v'\neq v}a^b_{v',v}=0\right\}$
3. $R_0=V-\{S_0\cup\sigma(S_0,A_b)\}$
4. $A_0=\left[a^0_{v,v'}\right]$ s.t. $a^0_{v,v'}=0$ if $v\notin R_0\vee v'\notin R_0$; otherwise $a^0_{v,v'}=a^b_{v,v'}$
5. $t=0$
6. iterate until $R_t=\emptyset$
   (a) $v^*=\arg\max_{v\in R_t}\{\sigma_1(\{v\},A_t)\}$
   (b) $S_{t+1}=S_t\cup\{v^*\}$
   (c) $R_{t+1}=R_t-\{\{v\}\cup\sigma_1(\{v\},A_t)\}$
   (d) $A_{t+1}=\left[a^{t+1}_{v,v'}\right]$ s.t. $a^{t+1}_{v,v'}=0$ if $v\notin R_t\vee v'\notin R_t$; otherwise $a^{t+1}_{v,v'}=a^t_{v,v'}$
   (e) $t\leftarrow t+1$
7. Return S_t

4 Experimental results

In this section, we report experimental results comparing the HS, SA, GA, GS algorithms and the proposed IMH. The code and the data for these experiments can be found in the university group site.2 First we describe the construction of the experimental graphs. Next, we evaluate the IM solution methods comparatively on a large collection of synthetic graphs of increasing size, and on subgraphs of increasing size of a real life social network. Finally, we report response time results comparing GS and IMH.

4.1 Graph construction

A social network is defined as a directed graph *G*(*V,E*) where *V* is the set of nodes that represents the set of users and *E* the set of edges that represents the set of relationships among users. Given a directed graph *G*(*V,E*) edges are weighted by *wij*=1/*degreein*(*j*), thus the graph becomes a weighted graph *G*(*V,E,W*) where *W* is the set of weights that are values in the interval [0,1]. Some of the experiments reported here are done on synthetic graphs. To build such graphs we generate the random weights of a complete graph, determining the in-degree of each node. Thus, from the a probabilistic adjacency matrix we generate graph instances *Gi*(*V,E*') where edges will appear according with the probabilistic weight. An weighted edge with a high weight has more probability to appear in the sampled graph *G'i*. We define *G*'=*G*'$_1$,…,G'$_n$ as the set of sample graphs used to test the diverse IM heuristics. For the experiments referred below, the ICM influence propagation estimation consists in the average of the propagations over the set of sample graphs. Experiments involve the repetition of the search over a 100 synthetic graphs *per* size parameter value.

4.2 Experimentation with synthetic graphs

Figures 1, 2 and 6 show a comparison among HS, SA, GA, GS algorithms, and the proposed IMH. Figure 1 plots the average IM-seed node set size found by the algorithms for simulated social network graphs of increasing sizes. Figure 2 plots the average influence spread size. Figure 3 plots the ratio of the influence spread size versus the size of the IM-seed set in order to visualize the relative success of each node in the IM-seed set. Figures 1a, 2a, and 6a give results for small size graphs. Figures 1b, 2b, and 6b give results for big size graphs. We have not computed SA for the bigger graphs due to its very slow convergence and large response times. If we consider the size of the IM-seed set obtained by each method, we find that for small size graphs (Figure 1a) IMH and GS compete with GA to provide the smaller size IM-seed sets, while for large graphs (Figure 1b) GA and HS are always giving smaller IM-seed sets. This situation is reversed if we consider the size of the influence spread in Figure 2. Notice that GS and IMH give almost always the same influence spread solution, though in some cases GS is slightly different from IMH in Figure 2a, as confirmation that IMH is in fact an approximation of GS. The size of the influence spread found by the GA, HS and SA heuristics is much smaller, near half of the size of the spread

found by GS and IMH. The ratio of the influence spread to the IM-seed set size plotted in Figure 6 is a measure of the quality of the solutions, which shows that the new heuristic is far better than the GA. HS and SA heuristics.

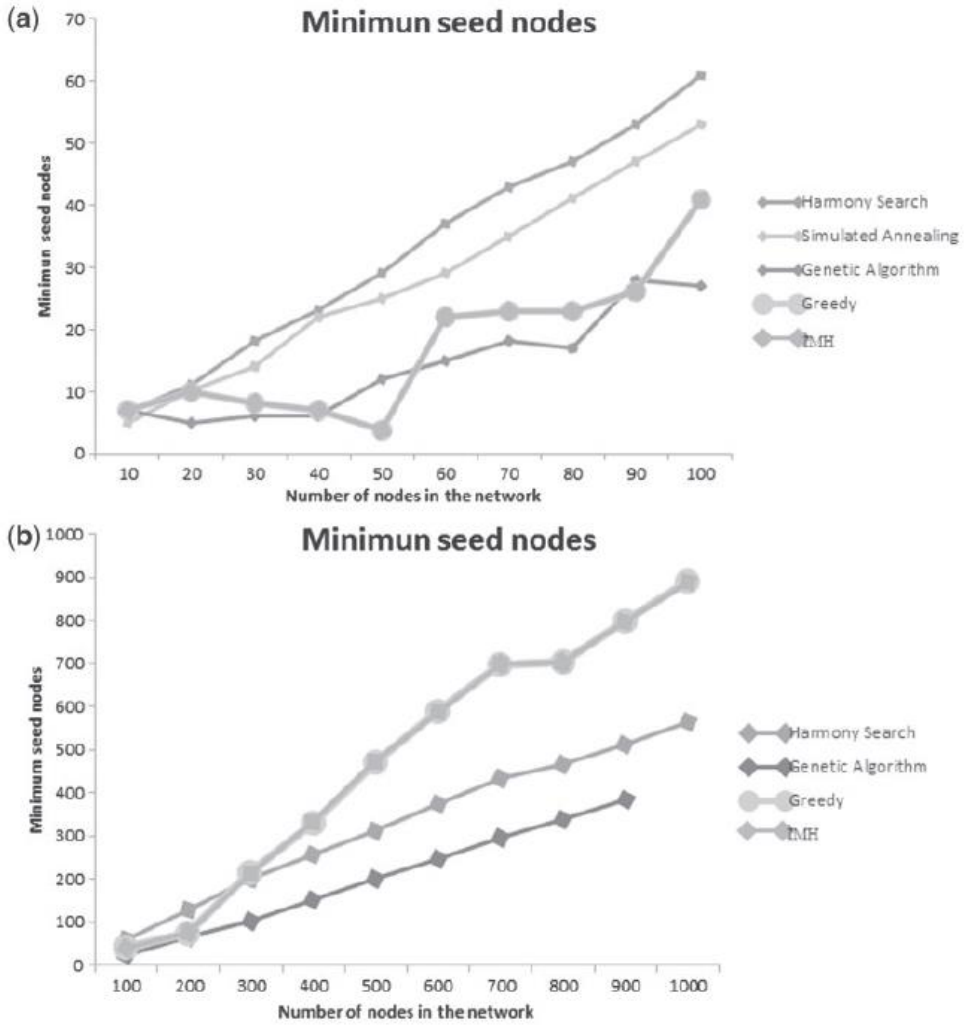

FIG. 1. Results for increasing size random graphs of GA, SA, HS, Greedy search and IMH measured as the size of the IM-seed found.

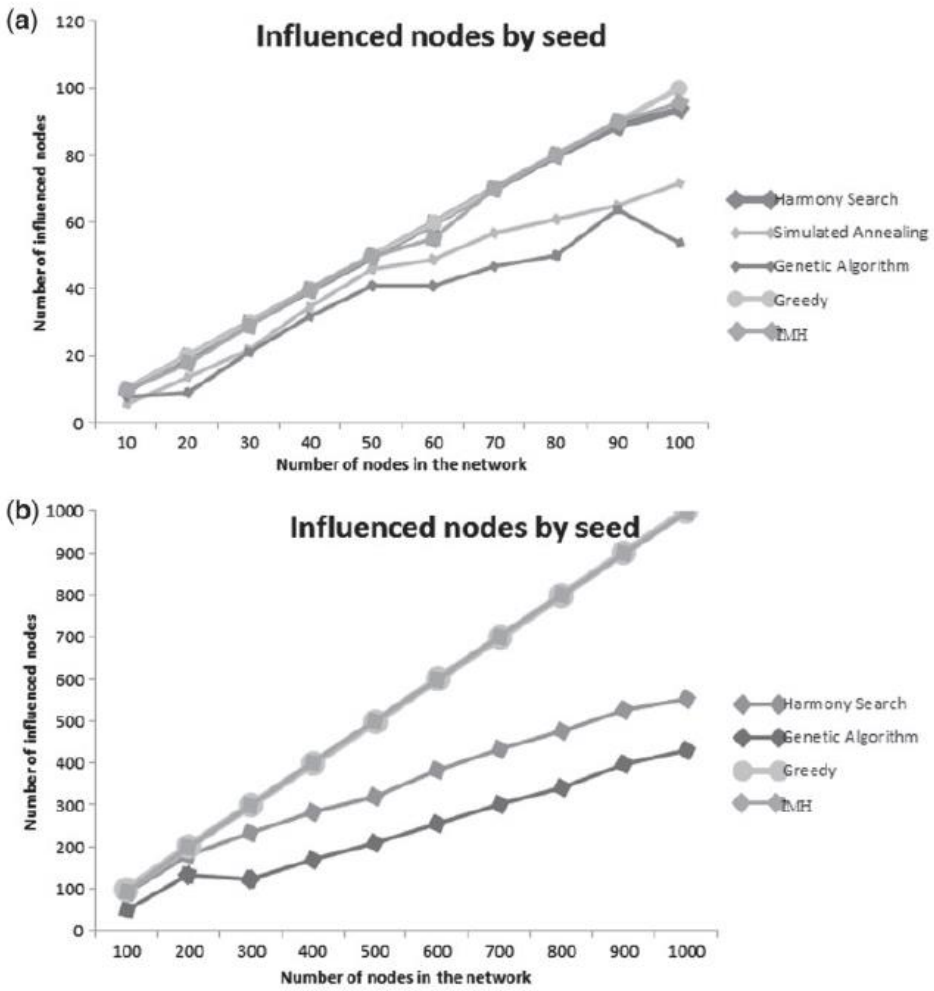

FIG. 2. Results for increasing size random graphs of GA, SA, HS, Greedy search and IMH measured as the size of influence spread.

4.3 Experimentation with Epinions database

In order to evaluate the algorithms on a realistic problem, we apply them to the trust graph from the Epinions site,3 which is a social web service where users provide reviews of products of any kind, ranging from music up to perfumes or construction hardware. These reviews are the base for the establishment of trust relations between users. Trust is a binary variable taking values in {−1,1}: a truster user can choose to trust (1) or distrust (-1) another, the trustee. Negative trust values are not published in the web service, but the anonymized dataset provided for experimentation, which is available to the public,4 contains also negative Trust values. This dataset has 841,372 data samples. Each sample is a triplet (*A,B,tAB*) composed of two user indexing numbers (no personal data of any form is included) and the binary Trust value of the first user on the second user. Therefore, Trust relations define a directed graph, with weighted edges.

The influence propagation can be interpreted as the propagation of trust in the network. Experimentation is done on subnetworks of sizes that are in the ranges from 10 to 100, and from 100 to 700. Figures 4, 5 and 6 show results of experimentation over these Epinions subnetworks. In all cases, the proposed IMH and GS algorithm provide the greatest influence spread, while GA and SA provide more compact IM-seed of lesser influence spread. Attending to the ratio plots in Figure 6 we find that there is significative superiority of the GS and IMH to the other methods. The GA and SA contain implicit mechanisms for the minimization of the IM-seed size, because they seek minimal representations through their feasible solution coding; however, they do not ensure the influence spread covering the complete social graph. This lack of completeness can be more relevant in some applications such advertisement.

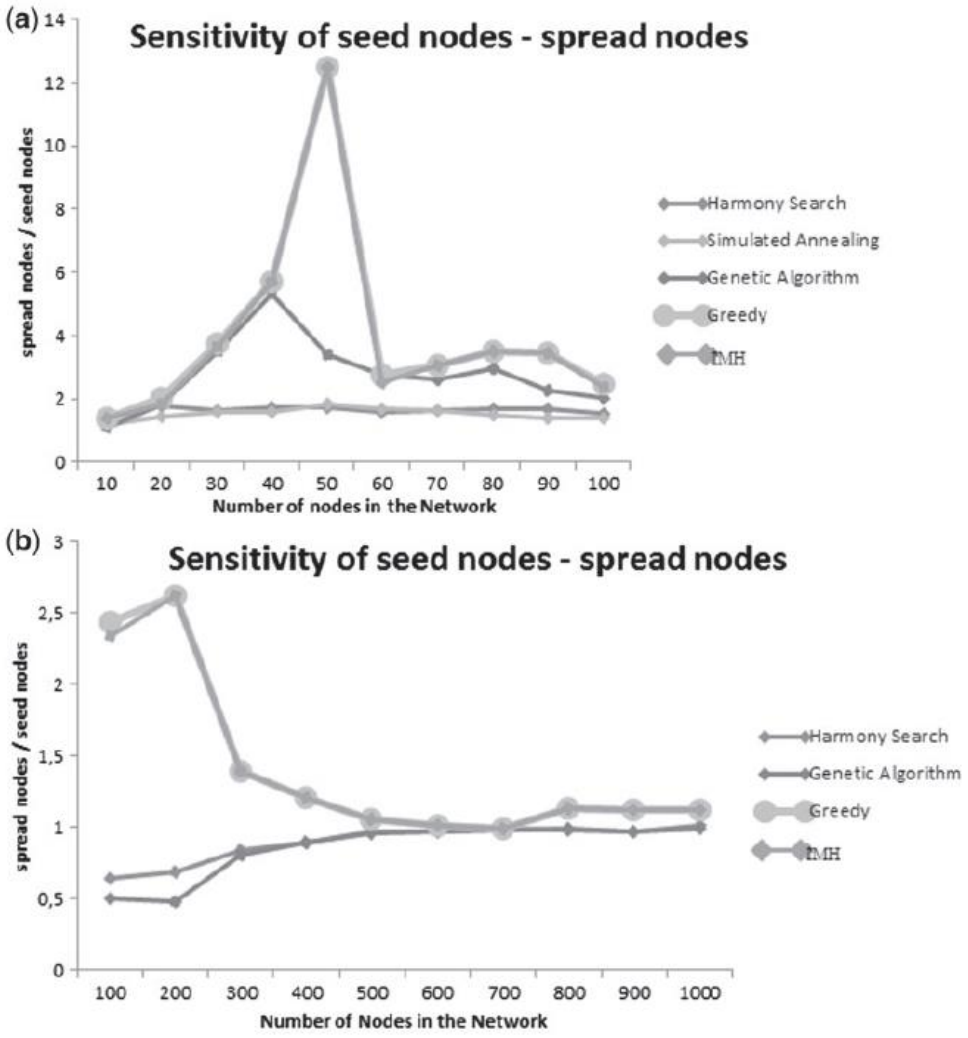

FIG. 3. Results for increasing size random graphs of GA, SA, HS, Greedy search and IMH measured as the ratio of influence spread to the IM-seed size.

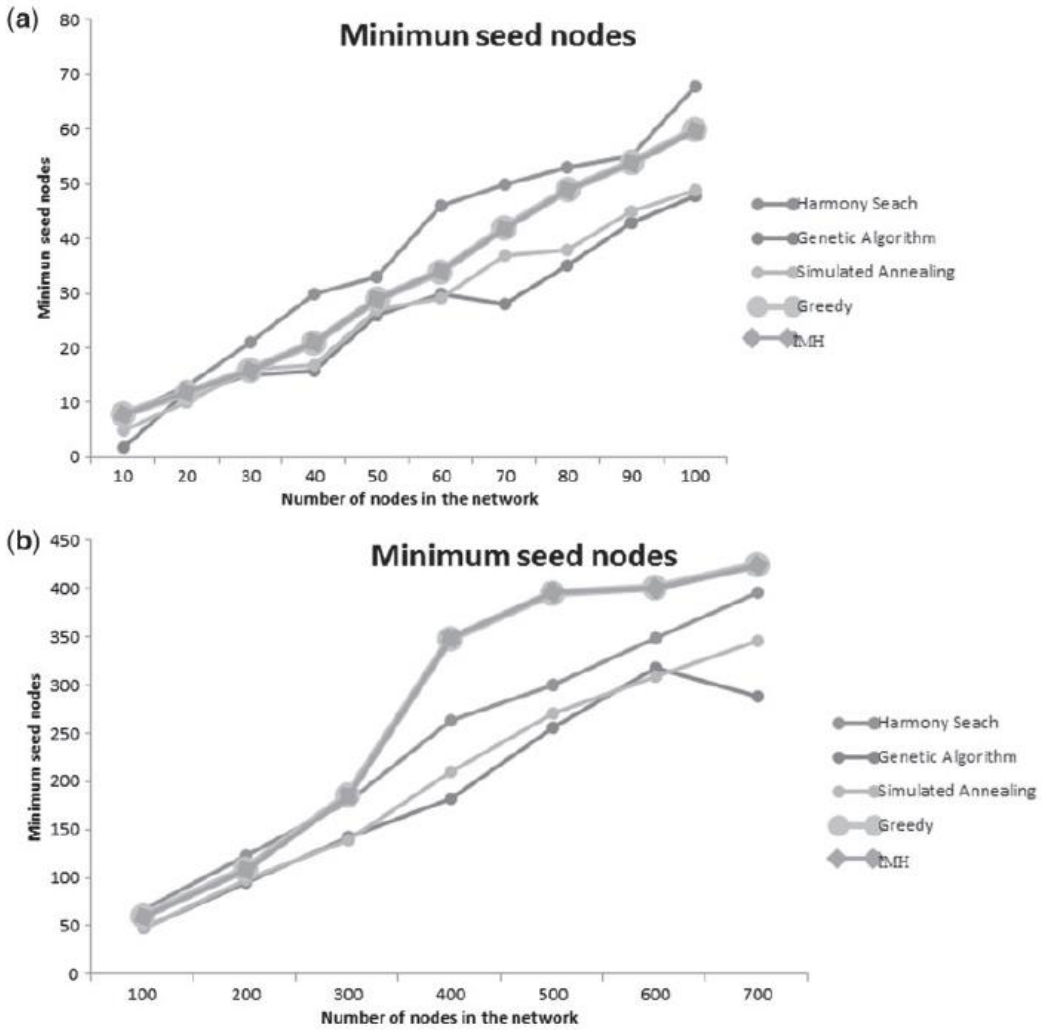

FIG. 4. Results for increasing size Epinions subgraphs of GA, SA, HS, Greedy search and IMH measured as the IM-seed size.

4.4 Experiments measuring response time

We carried out two experiments in order to compare the computational time cost of the GS algorithm and the proposed IMH method. The experiments were run on a 3 GHz 4 core Intel desktop computer with 34 Gb RAM, using Matlab implementations. The first experiment consists on solving IM on four different kinds graphs of 1000 nodes with increasing edge density. Table 1 shows the average time spent to obtain their approximations to the optimal IM-seed set of influential nodes. The response time of the GS approach grows with the edge density. In contrast, IMH method always achieves a solution under one second. The reason of this phenomenal speed up is that the IMH does not compute the complete spread, but only considers the nearest neighbours of each node. Though this is a rather risky approach, results in the previous sections confirm that the approximation is good enough to compare with GS and other heuristics. We have made our own implementation of CELF whose time results are also show on in Table 1. CELF improves over the raw GS implementation, but it is still slower than IMH on average.

For the second experiment we keep a small value of edge density in the graph but we increase the number of nodes of the experimental graphs. We test the algorithms on 100 graphs with sizes ranging from 1000 nodes to 10000 nodes, increasing a thousand nodes each time. Table 2 shows the time spent to get the minimal IM-seed nodes. Even for the smaller graph (1000 nodes) the time spent for execution of GS is already is some orders of magnitude of the IMH time. The CELF time is not so big, but, nevertheless far from IMH. When the graph size increases, the GS

algorithm response time increases exponentially, due to the cost of computing the influence spread of each candidate node.

In contrast, the proposed IMH method response time grow linearly, remaining below 2.5 seconds with the biggest graph. The optimization carried out by CELF trading space for computational time does alleviate the difference, but still managing the tables requires time and the response of CELF also grows exponentially, though more slowly than raw GS.

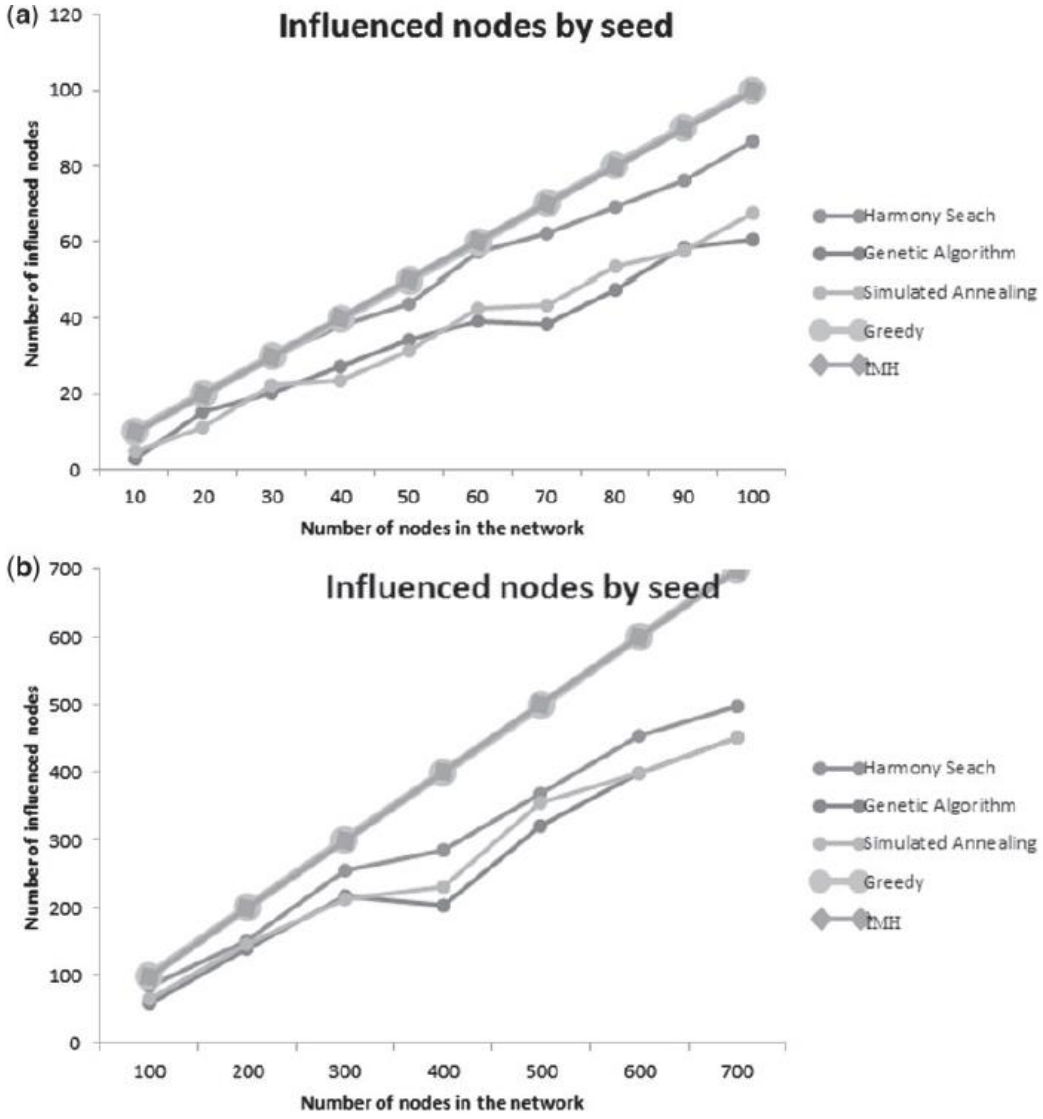

FIG. 5. Results for increasing size Epinions subgraphs of GA, SA, HS, Greedy search, and IMH measured as the size of influence spread.

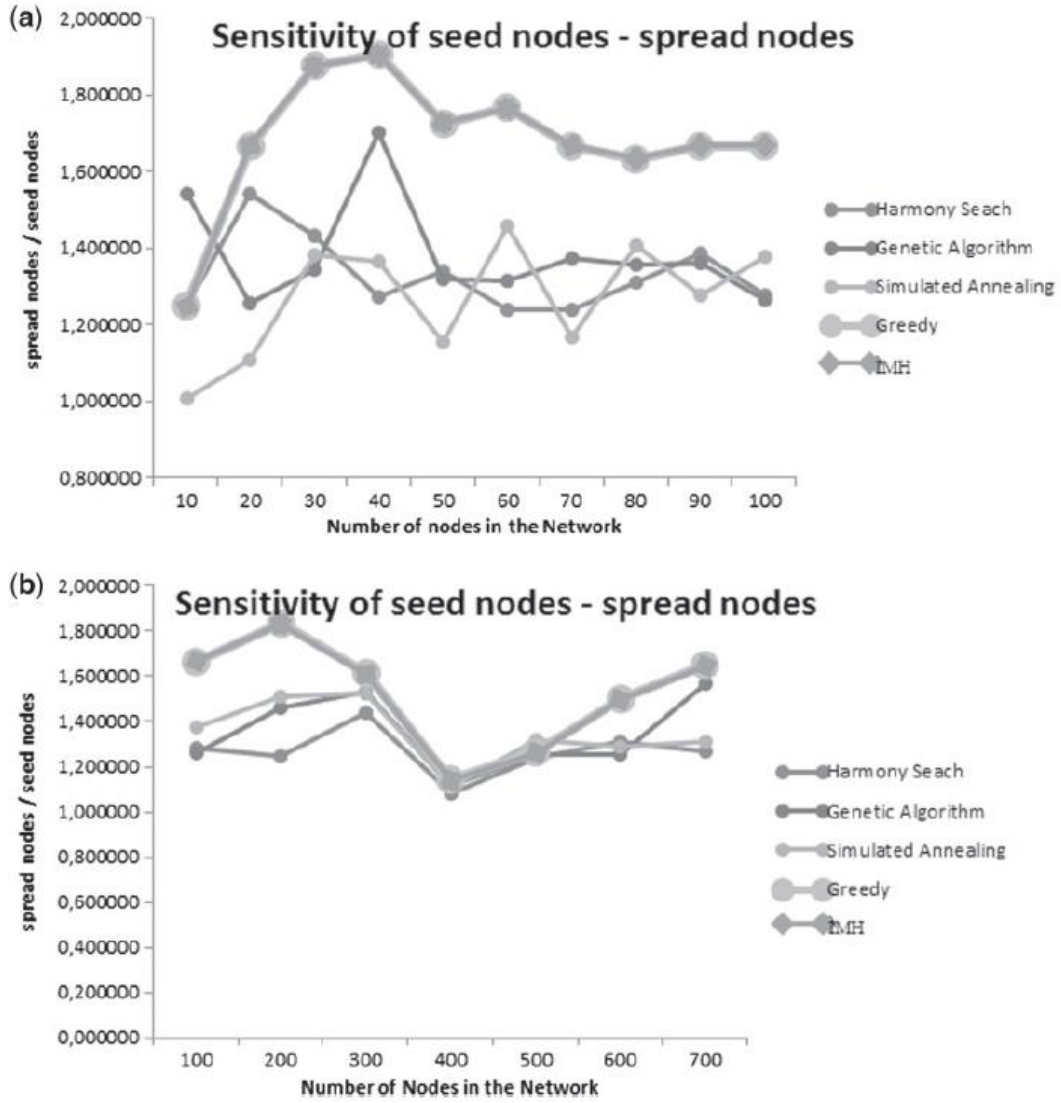

FIG. 6. Results for increasing size Epinions graphs of GA, SA, HS, Greedy search and IMH measured as the ratio of influence spread to the IM-seed size.

TABLE 1. Response time for increasing density graphs of 100 nodes

Density	IMH	CELF	GS
0.000011	0.018 sec.	1.233 sec.	5.587 sec.
0.00011	0.024 sec.	1.423 sec.	5.817 sec.
0.0011	0.031 sec.	15.32 sec.	> 5 min.
0.011	0.050 sec.	20.45 sec.	> 5 min.

Comparison of speed using graphs of the same size and different edge density.

$\|V\|$	IMH	CELF	GS
1000	0.065 sec	1.115 sec	3.586 sec
2000	0.119 sec	1.121 sec	35.938 sec
3000	0.182 sec	1.325 sec	> 5 min
4000	0.363 sec	2.124 sec	> 5 min
5000	0.446 sec	2.675 sec	> 5 min
6000	0.706 sec	3.306 sec	> 5 min
7000	1.702 sec	41.412 sec	> 5 min
8000	1.914 sec	53.413 sec	> 5 min
9000	2.012 sec	125.129 sec	> 5 min
10000	2.435 sec	182.357 sec	> 5 min

TABLE 2. Response time for graphs of constant edge density but increasing number of nodes $|V|$

5 Conclusions

A new heuristic search method for Influence Maximization (IMH) is proposed in this paper. It relaxes the influence spread search by considering only paths of length 1 however, it is guaranteed to terminate covering the entire graph. We provide comparison with other well-known heuristics, namely Simulated Annealing, Genetic Algorithm, HS and GS algorithm over a collection of synthetic and real social network graphs. The proposed heuristic method compares with the GS algorithm, providing the largest influence spread results with minimum IM-seed nodes, improving other heuristics.

Moreover, the proposed IMH method is always faster than Greedy algorithm. Future works will address the formulation of methods able to deal with very large social networks, with node sites escalating in the order of hundreds of million nodes, which is closer to current real life social networks.

Our proposed heuristic IMH is very suitable for parallel implementation because candidate nodes to be included in the seed set can have their influences computed in parallel. Future works may address the implementation in CUDA or similar parallel programming environments in order to experiment with general purpose GPUs [35]. Parallel implementations would allow to deal with very large social networks, such as facebook in a modest computing environments.

Acknowledgements

The Computational Intelligence Group is funded by the Basque Government with grant IT874-13, and participates at UIF 11/07 of UPV/EHU. Manuel Grana and Michal Wozniak were supported by EC under FP7, Coordination and Support Action, Grant Agreement Number 316097, ENGINE European Research Centre of Network Intelligence for Innovation Enhancement.

References

[1] M. N. Ambia, H. M. Hasanien, A. Al-Durra and S. M. Muyeen. Harmony search algorithm-based controller parameters optimization for a distributed-generation system, *IEEE Transactions on Power Delivery*, **30**, 246–255, 2015.
[2] H. V. H. Ayala, L. Dos Santos Coelho, V. C. Mariani, M. V. Ferreira Da Luz, and J. V. Leite. Harmony search approach based on ricker map for multi-objective transformer design optimization. *IEEE Transactions on Magnetics*, **51**, 1–4, 2015.
[3] T. Chao, Y. Yan, P. Ma, M. Yang and Y. W. Hu. Optimization of electromagnetic railgun based on orthogonal design method and harmony search algorithm, *IEEE Transactions on Plasma Science*, **43**, 1546–1554, 2015.
[4] R. Diao, F. Chao, T. Peng, N. Snooke and Q. Shen Feature selection inspired classifier ensemble reduction, *IEEE Transactions on Cybernetics*, **44**, 1259–1268, 2014.

[5] P. Domingos and M. Richardson. Mining the network value of customers. In *Seventh International Conference on Knowledge Discovery and Data Mining*, 2001.
[6] M. Dorigo. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*, 1992.
[7] Y.-H. Fu, C.-Y. Huang and C.-T. Sun. Using global diversity and local topology features to identify influential network spreaders. *Physica A: Statistical Mechanics and its Applications*, **433**, 344–355, 2015.
[8] M.-L. Gao, L.-L. Li, X.-M. Sun and D.-S. Luo. Face tracking based on differential harmony search. *Computer Vision, IET*, **9**, 98–109, 2015.
[9] S. Galhotra, A. Arora, S. Virinchi and S. Roy. Asim: A scalable algorithm for influence maximization under the independent cascade model. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pp. 35–36. International World Wide Web Conferences Steering Committee, 2015.
[10] Z. W. Geem, J.H. Kim and G.V. Loganathan. Harmony search optimization: application to pipe network design, *International Journal of Modelling & Simulation*, **22**, 125–133, 2002.
[11] Z. W. Geem. Novel derivative of harmony search algorithm for discrete design variables. *Applied Mathematics and Computation*, **199**, 223–230, 2008.
[12] Z. W. Geem, J. H. Kim and G. V. Loganathan.Anew heuristic optimization algorithm: harmony search. *Simulation*, **76**, 60–68, 2001.
[13] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine learning*, **3**, 95–99, 1988.
[14] J. Goldenberg. Talk of the network : a complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 211–223, 2001.
[15] A. Goyal, W. Lu and L. V. S. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th International Conference Companion on World Wide Web*, pp. 47–48. ACM, 2011.
[16] A. Goyal, W. Lu and L. V. S. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *2011 IEEE 11th International Conference on Data Mining (ICDM)*, pp. 211–220. IEEE, 2011.
[17] A. Goyal, F. Bonchi and L. V. S. Lakshmanan. A data-based approach to social influence maximization. In *Proceedings of the VLDB Endowment*, **5**, 73–84, 2011.
[18] M. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, **83**, 1420–1443, 1978.
[19] M. Heidari, M. Asadpour and H. Faili. Smg: Fast scalable greedy algorithm for influence maximization in social networks. *Physica A: Statistical Mechanics and its Applications*, **420**, 124–133, 2015.
[20] J. H. Holland. *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
[21] Q. Jiang, G. Song, C. Gao, Y. Wang, W. Si and K. Xie. Simulated annealing based influence maximization in social networks. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
[22] D. Kempe, J. Kleinberg and E. 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*, New York, NY, USA, 2003, KDD '03, pp. 137–146, ACM.
[23] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. Optimization by simulated annealing. *Science*, **220**, 671–680, 1983.
[24] S. Kirkpatrick. Optimization by simulated annealing: quantitative studies. *Journal of Statistical Physics*, **34**, 975–986, 1984.
[25] K. S. Lee and Z. W. Geem. A new structural optimization method based on the harmony search algorithm. *Computers & Structures*, **82**, 781–798, 2004.
[26] K. S. Lee and Z. W. Geem. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, **194**, 3902–3933, 2005.
[27] J.-R. Lee and C.-W. Chung. A query approach for influence maximization on specific users in social networks. *IEEE Transactions on Knowledge and Data Engineering*, **27**, 340–353, 2015.
[28] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen and N. Glance. Cost effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 420–429. ACM, 2007.
[29] D. Lopez-Pintado. Diffusion in complex social networks. *Games and Economic Behavior*, **62**, 573–590, 2008.

[30] Z. Lu, W. Zhang, W. Wu, J. Kim and B. Fu. The complexity of influence maximization problem in the deterministic linear threshold model. *Journal of Combinatorial Optimization*, **24**, 374–378, 2012.
[31] M. Mahdavi, M. Fesanghary and E. Damangir. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, **188**, 1567–1579, 2007.
[32] R. Mohamadi-Baghmolaei, N. Mozafari and A. Hamzeh. Trust based latency aware influence maximization in social networks. *Engineering Applications of Artificial Intelligence*, **41**, 195–206, 2015.
[33] A. Monteserin and A. Amandi. Whom should i persuade during a negotiation? an approach based on social influence maximization. *Decision Support Systems*, **77**, 1–20, 2015.
[34] M. G. H. Omran and M. Mahdavi. Global-best harmony search. *Applied Mathematics and Computation*, **198**, 643–656, 2008.
[35] S. Orts-Escolano, V. Morell, J. Garcia-Rodriguez, M. Cazorla and R. B. Fisher. Real-time 3d semi-local surface patch extraction using gpgpu: Application to 3d object recognition. *Journal of Real-Time Image Processing*, **10**, 647–666, 2015.
[36] G. Palubeckis. Fast simulated annealing for single-row equidistant facility layout, *Applied Mathematics and Computation*, **263**, 287–301, 2015.
[37] N. Pathak, A. Banerjee and J. Srivastava. A generalized linear threshold model for multiplecascades. In *2010 IEEE 10th International Conference on Data Mining (ICDM)*, pp. 965–970. IEEE, 2010.
[38] S. Peng, M. Wu, G. Wang and S. Yu. Containing smartphone worm propagation with an influence maximization algorithm. *Computer Networks*, **74**, 103–113, 2014.
[39] B. A. Prakash, D. Chakrabarti, N. C. Valler, M. Faloutsos and C. Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowledge and Information Systems*, **33**, 549–575, 2012.
[40] K. Rahimkhani, A. Aleahmad, M. Rahgozar and A. Moeini. A fast algorithm for finding most influential people based on the linear threshold model. *Expert Systems with Applications*, **42**, 1353–1361, 2015.
[41] K. Saito, M. Kimura, K. Ohara and H. Motoda. Super mediator, a new centrality measure of node importance for information diffusion over social network. *Information Sciences*, **329**, 985–1000, 2015.
[42] C. K. Shiva and V. Mukherjee. Comparative performance assessment of a novel quasioppositional harmony search algorithm and internal model control method for automatic generation control of power systems. *Generation, Transmission Distribution, IET*, **9**, 1137–1150, 2015.
[43] G. Song, X. Zhou, Y. Wang and K. Xie. Influence maximization on large-scale mobile social network: A divide-and-conquer method. *IEEE Transactions on Parallel and Distributed Systems*, **26**, 1379–1392, 2015.
[44] M. Tarkeshwar and V. Mukherjee. Quasi-oppositional harmony search algorithm and fuzzy logic controller for load frequency stabilisation of an isolated hybrid power system. *Generation, Transmission Distribution, IET*, **9**, 427–444, 2015.
[45] P. Yadav, R. Kumar, S. K. Panda and C. S. Chang. Optimal thrust allocation for semisubmersible oil rig platforms using improved harmony search algorithm. *IEEE Journal of Oceanic Engineering*, **39**, 526–539, 2014.