Evaluating Surrogate Models for Multi-Objective Influence Maximization in Social Networks

Doina Bucur
University of Twente, EEMCS
Enschede, The Netherlands
d.bucur@utwente.nl

Giovanni Iacca
University of Trento, DISI
Povo (Trento), Italy
giovanni.iacca@unitn.it

Andrea Marcelli
Politecnico di Torino, DAUIN
Torino, Italy
andrea.marcelli@polito.it

Giovanni Squillero
Politecnico di Torino, DAUIN
Torino, Italy
giovanni.squillero@polito.it

Alberto Tonda
INRA, UMR 782 GMPA
Thiverval-Grigno, France
alberto.tonda@inra.fr

ABSTRACT

One of the most relevant problems in social networks is *influence maximization*, that is the problem of finding the set of the most influential nodes in a network, for a given influence propagation model. As the problem is NP-hard, recent works have attempted to solve it by means of computational intelligence approaches, for instance Evolutionary Algorithms. However, most of these methods are of limited applicability for real-world large-scale networks, for two reasons: on the one hand, they require a large number of candidate solution evaluations to converge; on the other hand, each evaluation is computationally expensive in that it needs a considerable number of Monte Carlo simulations to obtain reliable values. In this work, we consider a possible solution to such limitations, by evaluating a surrogate-assisted Multi-Objective Evolutionary Algorithm that uses an approximate model of influence propagation (instead of Monte Carlo simulations) to find the minimum-sized set of most influential nodes. Experiments carried out on two social networks datasets suggest that approximate models should be carefully considered before using them in influence maximization approaches, as the errors induced by these models are in some cases too big to benefit the algorithmic performance.

CCS CONCEPTS

• **Information systems → Social networks**; • **Networks → Online social networks**; • **Human-centered computing → Social networks**; • **Theory of computation → Evolutionary algorithms**; **Social networks**;

KEYWORDS

Social Networks, Influence maximization, Multi-Objective Evolutionary Algorithm, Surrogate models

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208238

ACM Reference Format:
Doina Bucur, Giovanni Iacca, Andrea Marcelli, Giovanni Squillero, and Alberto Tonda. 2018. Evaluating Surrogate Models for Multi-Objective Influence Maximization in Social Networks. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3205651.3208238

1 INTRODUCTION

Social networks are one of the most pervasive and disruptive phenomena that characterize our modern society. A recent report from Deloitte [7] has estimated that in 2014, Facebook alone enabled 227 B$ of economic impact, being a catalyst for a broad range of business opportunities in an "ecosystem" of marketers, app developers and providers of connectivity. Another study [18] has indicated that in 2015, the revenue due to social media advertising (i.e., all spending generated by online social networks such as Facebook, Twitter or LinkedIn) in the U.S. accounted for 0.06% of GDP, with an increasing trend over the next six years.

Social networks pose a number of hard-to-solve computational issues, ranging from data mining to link prediction and clustering. One of the most challenging problems is influence maximization, i.e. the problem of finding the set of the *most influential* "seed" nodes in a network, according to some influence *propagation (or diffusion) model* [14]. For instance, one may want to identify in a social network the group of "early adopters" of a new product that should be targeted by an aggressive marketing strategy in order to trigger the largest possible cascade of further adoptions, e.g. thanks to recommendations. Similar examples can be found in political campaigns, news diffusion, public opinion analysis, to name a few. In all these scenarios, being able to identify the influential nodes can have large implications. However, as shown in [14] this problem is NP-hard, and one can find the optimal set only under certain conditions and with some level of approximation.

In the last decade, a number of methods have been proposed in the literature to solve the influence maximization problem. Recent works tried to tackle this problem by means of Computational Intelligence, exploiting methods such as Simulated Annealing [12] and Evolutionary Algorithms [1–3, 15, 21, 22]. These approaches have shown promising results on a number of relatively large datasets taken from real-world networks. However, they suffer from one fundamental problem: to converge, they need to evaluate tens or

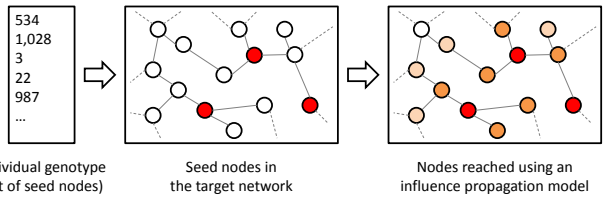


Figure 1: Schema of the proposed encoding. Seed nodes are internally represented as a list of integers. The fitness value is the average number of nodes that are influenced, following a given model of probabilistic influence propagation. In the example, the only node not influenced by the set of seeds is the white one (right frame, top left corner).

hundreds of thousands of candidates solutions, where each solution is a possible set of seed nodes. Furthermore, typically the evaluation of the influence of a set of nodes in a network relies on the computation of one or more probabilistic propagation models, whose complexity depends on the network size (the number of edges), and whose accuracy depends on the number of Monte Carlo simulations. This makes the application of such computational methods limited, especially on large networks with millions of edges, as on the one hand they require a large number of solution evaluations, and, on the other hand, each candidate solution is computationally expensive.

In this work we introduce a surrogate-assisted Evolutionary Algorithm (EA), formulating the influence maximization problem in a multi-objective fashion, similarly to [2, 3]. To select a possible surrogate model, we consider two approximations presented in the literature, specifically the *Expected Diffusion Value* (EDV) [12] and the *Probability Sorting* (PS) [17]. Given that these approximate models by definition introduce an approximation error (that, in general, is not constant across the search space), we then ask the question: *Can an EA actually use these approximations as surrogate models, even if sometimes the comparisons between two sets of seed nodes based on such such approximations can be wrong?* We answer this question by testing the proposed approach on two graph datasets taken from real-world social networks, and comparing the results obtained by the surrogate models with those obtained by Monte Carlo simulations.

The remainder of this paper is structured as following. The next section introduces the background concepts and briefly surveys the related work. Section 3 describes the proposed surrogate-assisted multi-objective evolutionary algorithm approach, and the numerical results are shown in section 4. Finally, section 5 draws the conclusions of this study.

2 BACKGROUND AND RELATED WORK

The influence of a set of nodes. Given a social network graph $G = (V, E)$, a probabilistic *influence propagation model* takes a subset $A \in V$ of k seed nodes and returns the *expected number of nodes* eventually influenced by the seeds. Given a seed set A, an exact computation of its overall influence, expressed in terms of nodes reached in the graph, $\sigma(A) \in V$, is #P-hard[1]. Monte-Carlo simulations of the given propagation model(s) are used to obtain an estimation with an arbitrarily small error. The time complexity of an estimation with Monte-Carlo simulations is $O(|E| \cdot R)$, where R is the number of simulation repetitions ([14] advises $R = 10^4$).

Maximizing influence. The problem of *influence maximization* is an instance of discrete optimization: given an influence propagation model and a numerical budget k, the goal is to identify the set A of k seeds that will maximize the output of the influence propagation model $\sigma(A)$ [14]. More formally:

$$\max_{A \in V} \sigma(A)$$
$$\text{s.t.} |A| = k$$

where the operator $|\cdot|$ indicates the cardinality of the set.

As shown in [14], influence maximization is NP-hard for the widely studied influence propagation models Linear Threshold (with random independent thresholds) and Independent Cascade (the one we use in this paper), with a hill-climbing heuristic proven to always reach a fraction arbitrarily close to $1 - 1/e$ of the optimal [14], where e is Euler's number.

In [2, 3], the problem of influence maximization has been further extended in a *multi-objective* form, such that there are two conflicting goals, namely (a) minimize the number of seed notes k (instead of fixing it a priori, as in the original influence maximization problem) and (b) maximize their influence $\sigma(A)$:

$$\begin{cases} \max_{A \in V} \sigma(A) \\ \min k \end{cases}$$
$$\text{s.t.} |A| = k$$

In those studies, such bi-objective formulation has been tackled by means of a Multi-Objective Evolutionary Algorithm (MOEA). However, while the MOEA was shown to outperform influence-maximization heuristics with the $1 - 1/e$ approximation guarantee, it was also noted that multi-objective influence maximization is particularly expensive computationally, since the fitness evaluation of each set of seed nodes A consists of a large number R of Monte-Carlo simulation repetitions for $\sigma(A)$, and a large number of evaluations is needed to uniformly cover the Pareto front.

Approximations for $\sigma(A)$. Graph-aware analytics were proposed as means to approximate the calculation of $\sigma(A)$, given A. [12] proposes the *Expected Diffusion Value* (EDV) to replace the expensive repeated simulations of diffusion, in the cases where the influence diffuses in the network according to the Independent Cascade model [14] with a small probability p of edge activation. The expected number of nodes influenced in the graph G by the seed set A of size k is estimated by:

$$k + \sum_{v \in N(A) \setminus A} \left(1 - (1-p)^{r(v)}\right)$$

where $N(A)$ is the one-hop area around the nodes in A, i.e., $N(A) = A \cup N^1(A)$, with $N^1(A) = \{v \mid \exists u \in A : (u, v) \in E\}$, and $r(v)$ a measure of the direct influence from A to node v, $r(v) = |\{u \mid u \in A : (u, v) \in E\}|$. The time complexity is in this case $O(kd)$, where d is the average degree in G. A use of EDV as surrogate was reported in [12],

[1] "Number P", or "Sharp P", is the class of function problems of the form "count the accepting paths of a nondeterministic Turing machine running in polynomial time".

where it was applied to Simulated Annealing (SA), and in [10], where it was applied over discrete Particle Swarm Optimization (PSO). In comparison with simulation-based fitness evaluations, [12] finds that the adoption of EDV reduced the accuracy of the SA by up to 10%, while [10] found solutions comparable to those of the best heuristics.

Another algorithmic approach to remove some of the time complexity from the $\sigma(A)$ estimation is to simplify the graph structure of the social network under study. The propagation models (such as Independent Cascade) see every graph edge as probabilistic (with p the probability of activation in the diffusion process); [17] studies the fundamental problem of extracting *a single representative instance* (i.e., a deterministic graph) from a probabilistic graph. A single diffusion simulation can then be executed on this representative to approximate the diffusion process on the original graph.

A simple algorithm for computing the representative instance for G is *Probability Sorting* (PS) [17], which outputs a deterministic graph $G^* = (V, E^*)$ with the same nodes as G, but only containing a subset of (non-probabilistic) edges which approximate the original node degrees in G. PS iteratively considers every edge in E for inclusion in E^* (Algorithm 1). The edge is included only if this edge addition decreases the discrepancy in node degrees between G and G^*, where $dis_2(u)$ denotes the difference (discrepancy) in the degree of node u between G and G^*. The complexity of this algorithm is linear, $O(E)$.

Algorithm 1: Probability Sorting (PS)

Input : Graph $G = (V, E)$, edge probability p
Output: Representative graph $G^* = (V, E^*)$

$E^* \leftarrow \emptyset$
foreach $(u, v) \in E$ **do**
  if $|dis_2(u) + 1| + |dis_2(v) + 1| < |dis_2(u)| + |dis_2(v)|$ **then**
    $E^* \leftarrow E^* \cup \{(u, v)\}$
  end
end

3 PROPOSED APPROACH

The proposed approach used to test the two approximations relies upon an MOEA for influence maximization in social networks, previously presented in [2, 3] and briefly summarized here.

The encoding of an individual. When using an EA to find good solutions to the influence maximization problem, the genome of an individual can be expressed as a subset of size k of all nodes in the target graph. Following the approach proposed in [1–3], a candidate solution A is then encoded as a k-sized list of node indices:

$$A = [n_1, n_2, \ldots n_k]$$

where $n_1, n_2, \ldots n_k$ are node indexes in $\{1, 2, \cdots N\}$, with N being the total number of nodes in the graph G.

Fitness functions and operators. The fitness function in single-objective optimization is typically an assessment of the average number of nodes influenced, as given by an influence propagation model such as IC. This evaluation usually requires multiple Monte-Carlo simulations, as influence propagation models are stochastic in nature, and several runs are needed to obtain reliable values [2, 15]. The genetic operators can change nodes in a seed set, or perform crossovers between two candidate seed sets. In the multi-objective problem formulation, the second fitness value is the number of nodes in the seed set (k). Minimizing this second fitness will result in Pareto fronts where each point is the most influential seed set for a given size k [2], with the added benefit of highlighting possible compromises between the number of seed nodes and the influence reached. To handle variable-sized genomes, in this case it is necessary to use operators that are able to add or remove nodes from a seed set, as illustrated in [2]. As only two objectives are evaluated, classic crowding-distance-based MOEAs such as NSGA-II [6] can be effectively applied to the task.

Population initialization. As the influence diffusion evaluation is always based on Monte-Carlo simulations, both the single- and multi-objective approaches share the issue of high computational costs. A possible solution is to initialize the initial population of the EA (or MOEA) in part with individuals already known to be good. For the selection of good seed nodes, inexpensive degree-based heuristics can be used: they greedily add nodes from the graph G to a set of seeds A, in order of decreasing node degrees. This implements the intuitive assumption that degree centrality translates into high influence. *Degree discount* heuristics are a refinement of the simplest degree heuristic, and have been shown to find good seed sets [4]. Here, we use the *Generalized degree discount* heuristic (GDD) [20] to compute seed sets for the MOEA; it improves the basic degree heuristic by excluding from the degree count of a candidate node v those edges which connect v to nodes already added to the seed set. Initializing the population with the results of a fast heuristic (in this case GDD) to quickly reach good solutions follows the idea presented in [3].

4 EXPERIMENTAL EVALUATION

The selected case studies used in the experimental evaluations are reported in Table 1. The **ego-Facebook** and **ca-GrQc** are taken from the SNAP repository [16]. In the experiments using Monte Carlo simulations, we use the Independent Cascade model with $p = 0.01$ or $p = 0.05$, and 100 repetitions.

Table 1: Network case studies

Social network	ego-Facebook	ca-GrQc
Nodes	4039	5242
Edges	88234	14496
Type of graph	undirected	undirected
Nodes in largest WCC	4039	4158
Nodes in largest SCC	4039	4158
Avg. clustering coeff.	0.6055	0.5296
Diameter	8	17

In the following subsections, we first analyze the accuracy of the two approximate methods, in terms of errors on pairwise comparisons between two candidate solutions. Then, we show the results obtained by the surrogate-assisted MOEA on the two datasets detailed in Table 1.

Table 2: Errors measured during pairwise comparisons obtained with both EDV and PS for randomly generated seed sets on ego-Facebook and ca-GrQc, with $p = 0.01$ and $p = 0.05$, and number of nodes ranging from 25 to 150.

	IC, p=0.01		IC, p=0.05	
	EDV	PS	EDV	PS
CA-GrQc (25 nodes)	12.62%	65.25%	18.62%	24.62%
CA-GrQc (50 nodes)	17.31%	59.87%	24.04%	22.32%
CA-GrQc (100 nodes)	23.74%	42.73%	21.88%	21.18%
CA-GrQc (150 nodes)	27.38%	34.10%	18.69%	24.44%
CA-GrQc (avg, 1-200 nodes)	**21.68%**	**45.66%**	**21.46%**	**26.23%**
Facebook (25 nodes)	28.86%	28.54%	54.76%	34.10%
Facebook (50 nodes)	21.26%	31.79%	52.10%	34.54%
Facebook (100 nodes)	30.27%	43.21%	49.56%	40.86%
Facebook (150 nodes)	24.56%	36.84%	59.72%	48.40%
Facebook (avg, 1-200 nodes)	**26.82%**	**37.67%**	**56.85%**	**36.96%**

4.1 Pairwise comparison of random seed sets

In this first set of experiments, the objective is to obtain an empirical evaluation of the accuracy of the selected influence diffusion approximations, EDV and PS.

While generally speaking it would be desirable for an approximation to return values in the same order of magnitude of the influence diffusion simulations (such that simulated and approximate influence values can be compared, and the approximation error can be computed), for the purpose of having a surrogate-assisted MOEA that can use *only* approximations (so to avoid computationally expensive simulations) the essential property is that the approximation provides the same result as the simulations, in terms of pairwise comparisons between seed sets. In other words, given two seed sets A and B, the difference between the influence diffusion approximate values $\sigma_a(A)$ and $\sigma_a(B)$ should have *the same sign* as the difference between simulated the corresponding diffusion values $\sigma_S(A)$ and $\sigma_S(B)$. This property is obviously important as pairwise comparisons are at the basis of the selection process performed by the MOEA to identify non-dominated solutions and thus converge to the Pareto front.

In order to empirically evaluate the approximations considered, EDV and PS, we generated 10000 random seed sets for each of the two graphs, **ego-Facebook** and **ca-GrQc**. We then compared all possible pairs of seed sets of the same size[2] by using simulations, and by using the two approximations; when the comparison of an approximation has a different sign with respect to the comparison performed using simulations, it is considered an error, otherwise the approximation is considered to be correct.

From the results reported in Table 2, it is noticeable how both approximations incur a considerable number of errors. More interestingly, the performance of EDV and PS seems to be dependent not only on the number of nodes in the seed set, but also on the graph, and on the probability of diffusion p. Furthermore, it seems that the two approximate models are able to compare correctly the expected influence diffusion of two seed sets when the difference between the influence diffusion (in simulation) is higher. Figure 2 shows the average difference of simulated influence diffusion between equally-sized seed sets (with size ranging from 2 to 200), both in cases of correct pairwise comparisons and errors, for CA-GrQc, with $p = 0.05$. It is apparent how both approximations are able to discriminate more efficiently when the difference in simulated influence diffusion is larger; or, in other terms, both approximations are less reliable when the difference becomes smaller.

As a side note, results from this first batch of experiments can also be exploited to assess the relative speed of computation of the approximate models, compared with complete simulations of influence diffusion. EDV is on average the quickest, being between 30 and 150 times faster than a simulation, depending on the target graph and probability of diffusion p. Surprisingly, PS is always *slower* than a complete simulation, and even at its fastest, it just gets closer to the speed of the simulation itself. This puzzling result might be due to the number of edges E in the considered graphs being still too small for PS to take advantage of its lower time complexity, as different overheads impact performance. In order to test this intuition, an additional evaluation of 1,000 random seed sets was carried on **ego-Twitter**, a graph with over 1.7 million edges and 81,000 nodes, also taken from the SNAP repository. Results, however (not reported here for the sake of brevity), showed PS to be slower than simulations also in this case. For these reasons, PS was not used in the experiments with the MOEA, only EDV.

4.2 Multi-objective evolutionary optimization

An empirical evaluation on random seed sets can only provide limited information. Even if EAs have a stochastic component, these algorithms explore the search space with a bias towards good solutions. While the approximations do not seem reliable, they might be good enough as a surrogate model to direct an EA towards good parts of the search space. This second set of experiments aims at assessing the efficiency of the approximations as fitness functions during an optimization run, comparing them to runs performed using simulations. All experiments were run with the parameters reported in Table 3. These parameters were chosen empirically, following the parameter setup previously used in [1–3].

[2]For the sake of simplicity, in this analysis we only consider pairwise comparisons between equally-sized sets. While during the MOEA optimization process sets of different size are also compared, we noted that in general larger seed sets are highly correlated with larger influence, both in simulation and with surrogate models. Therefore, dominance checks between two differently-sized seed sets using surrogates will be more consistent with the results of the simulations.

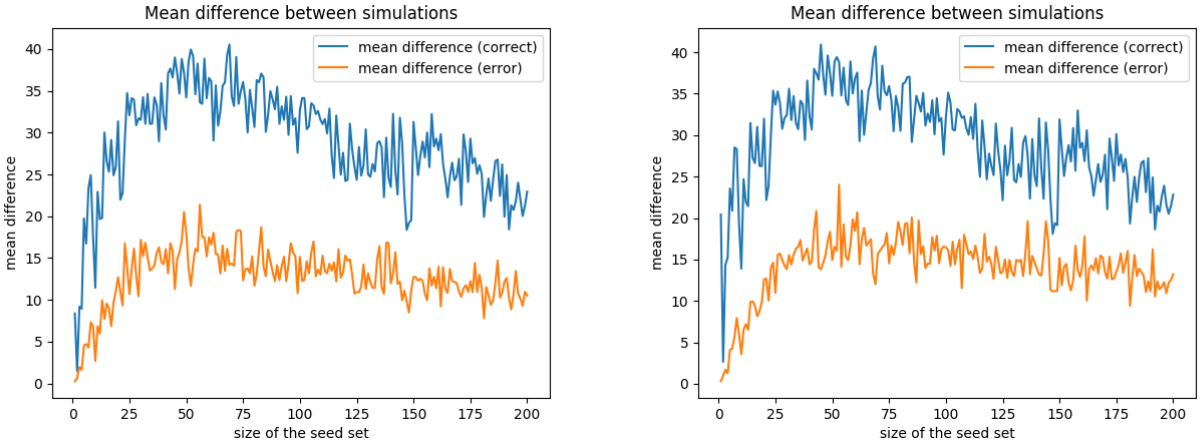


Figure 2: Average difference of simulated influence corresponding to pairwise comparisons approximated by EDV (left) and PS (right), on graph CA-GrQc, with $p = 0.05$, in both cases of correct comparisons (consistent with simulations) and errors (different from simulations). Correct comparisons by means of approximations correlate to larger influence differences.

Table 3: Parameters of the MOEA used in the experiments.

Parameter	Value	Operator	Probability
μ (pop. size)	2000	Add node	0.25
λ (offspring size)	2000	Remove node	0.25
τ (tournament selection)	2	Change node	0.25
MOEA (used as a base)	NSGA-II	One-point crossover	0.25

As the aim is to compare the approximations in a situation as close as possible to a real-world application, the starting population was initialized with values computed by the GDD heuristic, following the same experimental setup as in [3].

All the necessary code has been implemented using Python open-source modules `networkx` module [11] for computations on graphs, and `inspyred`[3] for the EA. Only one evolutionary operator (either crossover or mutation) was applied when generating a single individual. Further details on the algorithm can be found in [3].

Figure 3 shows the results of evolutionary runs on **ego-Facebook** with $p = 0.01$, using simulations or EDV as fitness values, respectively. In order to be comparable, objective values for all points have been recomputed using the simulations, given the original evolved individuals. It is immediately noticeable how the MOEA guided by the simulation-based fitness function is able to improve over the initial solutions used to initialize the population; on the other hand, EDV does not seem to move the Pareto front at all. As in this case EDV is about 50 times faster than the simulations, the figure also shows two Pareto fronts selected at comparable generations: while the EDV-based runs evaluated individuals for 1600 generations, and the simulation-based runs only for 32, the simulation-based results are still better.

Similar conclusions can be surmised from Figure 4, showing evolutionary runs on **ego-Facebook**, this time with $p = 0.05$. With these settings, EDV is about 150 times faster than the simulations, so the Pareto fronts from generations 2700 and 18 are compared. Interestingly, the search space defined by EDV seems to be highly uncorrelated w.r.t. the simulations, with the consequence that the performance of the surrogate-assisted EA is heavily impaired by the approximation; after a few generations, the results are even worse than the starting initialization. For completeness, for Figures 3-4 we report the videos of the generational trends as supplementary material online (links in the captions of these figures).

For graph **ca-GrQc**, both evolutionary runs (with and without surrogate models) are able to obtain only minor improvements over the starting initialization of the population, when $p = 0.01$ (see Figure 5). Still, simulations seem to perform marginally better for comparable wall-clock time, with EDV being about 100 times faster.

Finally, Figure 6 reports results for graph **ca-GrQc**, $p = 0.05$. In this scenario, EDV is about 30 times faster than the simulations. It is clearly noticeable how EDV actually impairs the evolutionary algorithm, forcing it to converge on results that are Pareto-dominated even by the original population, initialized with heuristic results.

5 CONCLUSIONS

In this paper, a comparison between different approximations for influence diffusion was presented. The comparison was focused on the applicability of approximations as surrogate models for evolutionary optimization, thus evaluating approximations' potential to correctly discriminate the relative influence diffusion values of two seed sets. First, the approximations' performance on pairwise comparison on random seed sets was assessed; then, approximations were compared with full simulations, as fitness functions on evolutionary optimization runs.

While the approximations evaluated in this work, EDV and PS, seem too unreliable to be exploited as surrogate models for evolutionary optimization of influence diffusion (at least in the way we used them in this work), the presented results open several research lines, namely:

[3] http://pythonhosted.org/inspyred/overview.html

(1) If the performance of an approximation is dependent on the peculiarities of a specific graph, it might be possible to find correlations between graph features and approximation effectiveness, even resorting to machine learning to perform predictions on how well a surrogate model works. However, this solution would require datasets of considerable size, and consequently a large number of experiments.
(2) As the ability of an approximation to compare the influence of two seed sets seems to be connected to their relative difference in predicted influence diffusion, it might be possible to still use approximations as surrogate models for an evolutionary optimization; it would suffice to call a complete simulation of two seed sets when the difference between their approximation values would fall under a given threshold. In this case, the main issue would become finding a sensible value for such a threshold, as this might also be graph-dependent. Another option would be using state-of-the-art strategies to schedule surrogate evaluations with EAs [13], such as the EGO framework [5] or pre-selection [8], or apply surrogate multi-fidelity fidelity surrogates[9] or a fidelity adjustment mechanism [19].
(3) Another possibility is that approximations might indeed be useful for the early exploratory stages of the optimization process, when it is not so important to discriminate between points with close influence diffusion values; this might explain the good results reported in literature with EDV [12]. In our case, initializing the MOEA with results that are already good might have blocked (or at least introduced a bias into) the evolutionary runs using approximations as fitness functions, as the optimization starts directly from an exploitation stage, where approximations become less effective. Further analyses are needed in this sense, for instance comparing the results presented in this paper with a surrogate-assisted EA started from a random population, or measuring the diversity of the initial population when it is started with a seeding strategy like the one we used here.
(4) Another alternative might be represented by using other approximations described in influence modeling literature. While EDV and PS are among the most popular, other approximations might be better suited to evolutionary optimization. Machine learning could also be used to find better (more realistic) propagation/diffusion models, from real data, which may possibly also remove the need for repeated simulations of the probabilistic propagation model.
(5) Finally, another possibility would be to simply extend the present work combining the same approximations with alternative optimization algorithms. It is possible that the performance of the MOEA used in this work is inherently hampered by the use of two chosen surrogate models, while these may work better in a single-objective context such as the Simulated Annealing proposed in [12], or with optimization algorithms that make use of graph-aware operators. Additionally, running experiments on multiple surrogate-assisted algorithms would allow to perform anytime comparisons at different times (i.e. at different values of generations, or fitness evaluations) during the optimization process.

To conclude, the preliminary results presented in this work are not enough to provide a final word on the effectiveness of approximations as surrogate models for influence diffusion optimization. However, they clearly show that the subject should be approached carefully, as a naive use of EDV or PS might lead to incorrect conclusions. Also, the present studies opens up a broad range of research directions that may be worth exploring in the future.

REFERENCES

[1] Doina Bucur and Giovanni Iacca. 2016. Influence Maximization in Social Networks with Genetic Algorithms. In *European Conference on the Applications of Evolutionary Computation*. Springer, 379–392.
[2] Doina Bucur, Giovanni Iacca, Andrea Marcelli, Giovanni Squillero, and Alberto Tonda. 2017. Multi-Objective Evolutionary Algorithms for Influence Maximization in Social Networks. In *European Conference on the Applications of Evolutionary Computation*. Springer, 221–233.
[3] Doina Bucur, Giovanni Iacca, Andrea Marcelli, Giovanni Squillero, and Alberto Tonda. 2018. Improving Multi-objective Evolutionary Influence Maximization in Social Networks. In *European Conference on the Applications of Evolutionary Computation*. Springer, 117–124.
[4] 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 (KDD '09)*. ACM, New York, NY, USA, 199–208.
[5] Ivo Couckuyt, Filip De Turck, Tom Dhaene, and Dirk Gorissen. 2011. Automatic surrogate model type selection during the optimization of expensive black-box problems. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*. 4269–4279.
[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
[7] Deloitte. 2015. Facebook's global economic impact. (2015).
[8] Michael Emmerich, Alexios Giotis, Mutlu Özdemir, Thomas Bäck, and Kyriakos Giannakoglou. 2002. In *Parallel Problem Solving from Nature — PPSN VII*. Springer Berlin Heidelberg, Berlin, Heidelberg, 361–370.
[9] Alexander I.J Forrester, András Sóbester, and Andy J. Keane. 2007. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 463, 2088 (2007), 3251–3269.
[10] Maoguo Gong, Jianan Yan, Bo Shen, Lijia Ma, and Qing Cai. 2016. Influence maximization in social networks based on discrete particle swarm optimization. *Information Sciences* 367-368 (2016), 600 – 614.
[11] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, CA USA, 11–15.
[12] Qingye Jiang, Guojie Song, Gao Cong, Yu Wang, Wenjun Si, and Kunqing Xie. 2011. Simulated Annealing Based Influence Maximization in Social Networks. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11)*. AAAI Press, 127–132.
[13] Yaochu Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 2 (2011), 61 – 70.
[14] David Kempe, Jon Kleinberg, and Éva Tardos. 2015. Maximizing the Spread of Influence through a Social Network. *Theory of Computing* 11, 4 (2015), 105–147.
[15] Pavel Krömer and Jana Nowaková. 2017. Guided Genetic Algorithm for the Influence Maximization Problem. In *Computing and Combinatorics: 23rd International Conference, COCOON 2017, Hong Kong, China, August 3-5, 2017, Proceedings*, Yixin Cao and Jianer Chen (Eds.). Springer International Publishing, Cham, 630–641.
[16] Jure Leskovec and Andrej Krevl. 2017. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (2017).
[17] Panos Parchas, Francesco Gullo, Dimitris Papadias, and Francesco Bonchi. 2015. Uncertain Graph Processing Through Representative Instances. *ACM Transactions on Database Systems* 40, 3, Article 20 (Oct. 2015), 39 pages.
[18] Statista. 2016. Digital Market Outlook. (2016).
[19] Handing Wang, Yaochu Jin, and Jan O Jansen. 2016. Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. *IEEE Transactions on Evolutionary Computation* 20, 6 (2016), 939–952.
[20] Xiaojie Wang, Xue Zhang, Chengli Zhao, and Dongyun Yi. 2016. Maximizing the Spread of Influence via Generalized Degree Discount. In *PLOS ONE*.
[21] Michał Weskida and Radosław Michalski. 2016. Evolutionary algorithm for seed selection in social influence process. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 1189–1196.
[22] Kaiqi Zhang, Haifeng Du, and Marcus W. Feldman. 2017. Maximizing influence in a social network: Improved results using a genetic algorithm. *Physica A: Statistical Mechanics and its Applications* 478 (2017), 20 – 30.

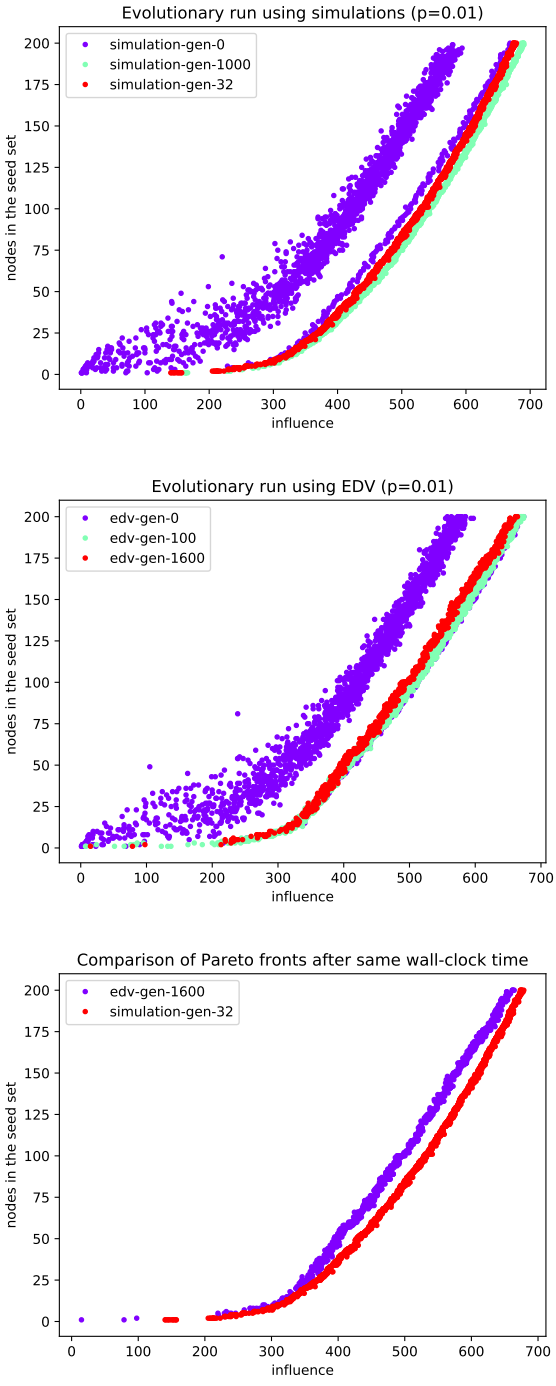


Figure 3: Results of evolutionary optimization runs on the ego-Facebook graph, using the IC model and $p = 0.01$. The two runs are presented as videos at https://youtu.be/tXOv40CzzRs and https://youtu.be/rt0fqimBXI8.

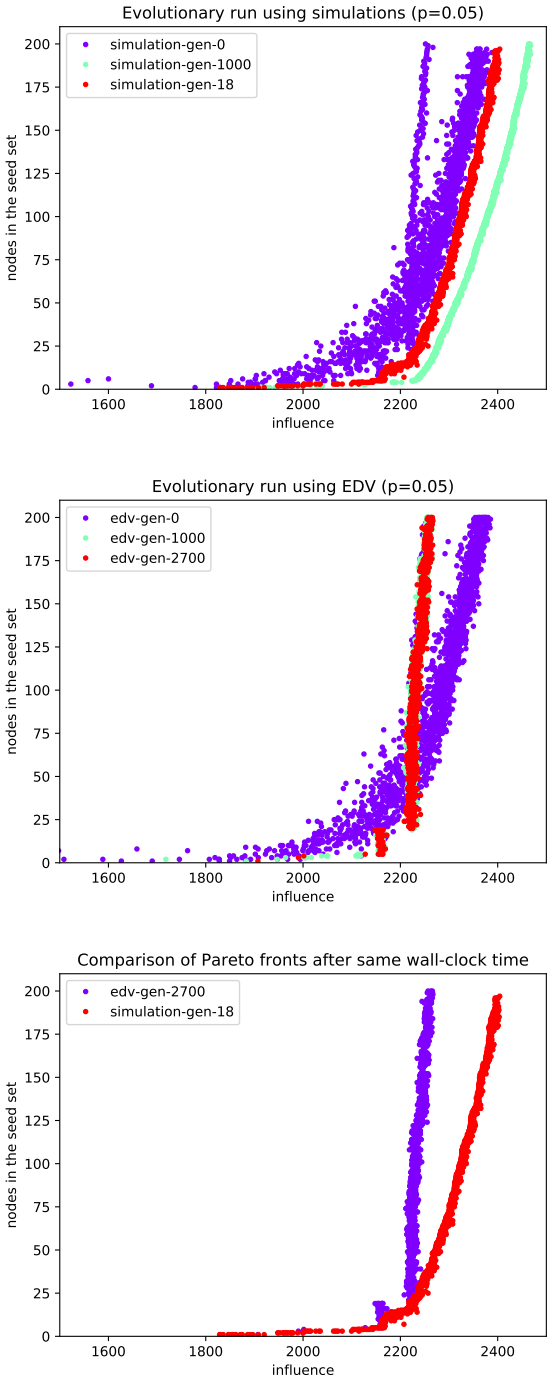
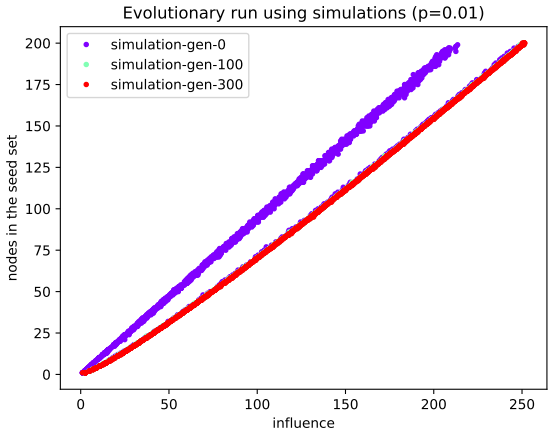
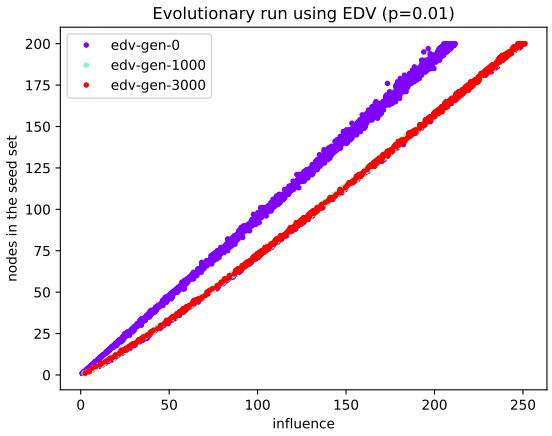


Figure 4: Results of evolutionary optimization runs on the ego-Facebook graph, using the IC model and $p = 0.05$. The two runs are presented as videos at https://youtu.be/U76sBifPQiY and https://youtu.be/a3EJ_2kOI1A.





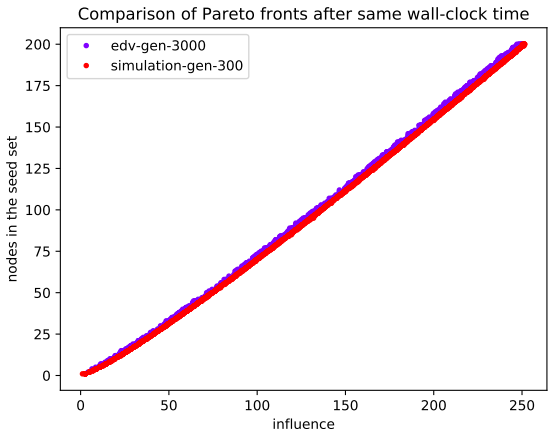
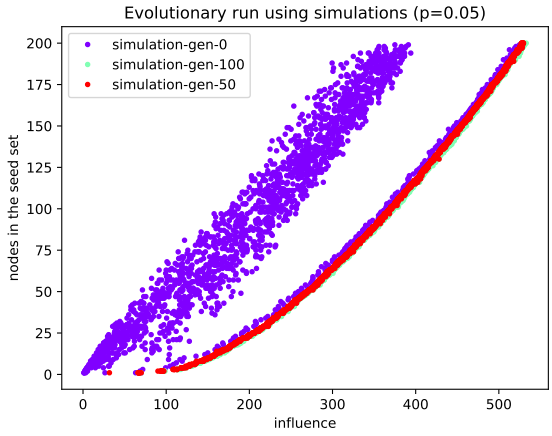
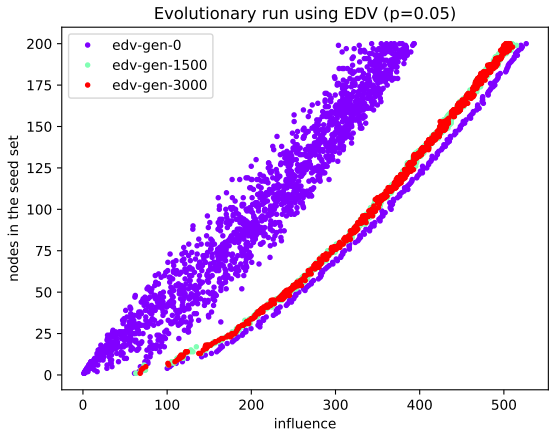


Figure 5: Results of evolutionary optimization runs on the ca-GrQc graph, using the IC model and $p = 0.01$.





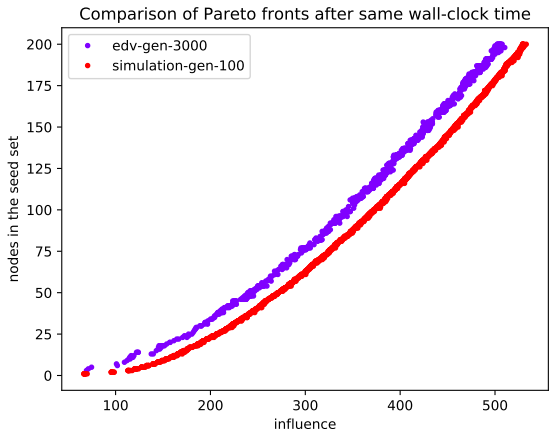


Figure 6: Results of evolutionary optimization runs on the ca-GrQc graph, using the IC model and $p = 0.05$.