Evolutionary Multiobjective Optimization to Target Social Network Influentials in Viral Marketing

Juan Francisco Robles*[,a], Manuel Chica[a,b], Oscar Cordon[a]

[a] *Andalusian Research Institute DaSCI "Data Science and Computational Intelligence", University of Granada, 18071 Granada, Spain*
[b] *School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia*

Abstract

Marketers have an important asset if they effectively target social networks' influentials. They can advertise products or services with free items or discounts to spread positive opinions to other consumers (i.e., word-of-mouth). However, main research on choosing the best influentials to target is single-objective and mainly focused on maximizing sales revenue. In this paper we propose a multiobjective approach to maximize the revenue of the marketing campaigns while reducing the costs by locating influentials using local social network metrics. Two evolutionary multiobjective optimization algorithms, NSGA-II and MOEA/D, a multiobjective adaptation of a single-objective genetic algorithm, and a greedy algorithm, are applied to the problem to generate a set of non-dominated solutions that allow marketers to consider multiple targeting options. Our proposal uses a realistic agent-based market framework to evaluate the fitness of the individuals by simulating the viral campaigns. We apply the algorithms to five network topologies and a real data-generated social network, which represent consumers' connections in the market, showing that both MOEA/D and NSGA-II outperform the single-objective and the greedy approaches. More interestingly, we show a clear correlation between the algorithms' performance and the diffusion features of the social networks.

Key words: Influentials Targeting, Social Networks, Evolutionary Multiobjective Optimization, Agent-based Modeling.

*Corresponding author
Email addresses: jfrobles@ugr.es (Juan Francisco Robles), manuelchica@ugr.es (Manuel Chica), ocordon@decsai.ugr.es (Oscar Cordon)

1. Introduction

On-line social networks such as Facebook or Instagram make people (potential consumers) more connected than ever before. With just a few actions, consumers can instantly communicate their products and brands' opinions. Social networks' influentials have thousands of friends and a word-of-mouth process can create a cascade of positive or negative information about a brand. This word-of-mouth process in social networks is crucial for marketers and advertisers (Leskovec et al., 2007; Haenlein and Libai, 2017). In fact, people place more value in friends' recommendations than in those from traditional advertisement channels such as TV. Viral marketing (VM) consists of targeting certain consumers to encourage a faster product's adoption (Haenlein and Libai, 2017). The selection of these influentials is not a random but a complex optimization process that involves the analysis of the social network of consumers to favor a positive information diffusion.

Designing the best VM strategies before running them in the real market is possible through the use of artificial social networks (Watts and Dodds, 2007) and simulations using paradigms such as agent-based modeling (ABM) (Epstein, 2006). ABM can describe and simulate network-level interactions between consumers to reproduce word-of-mouth mechanisms that provide realism into marketing models and allow modelers to study emergent behaviors (Janssen and Jager, 2003; Epstein, 2006). These models enable the aggregation of individual-level interactions through the underlying artificial social network, representative of a real social network of individuals, creating high level outcomes and incorporating individual behavioral rules without higher level assumptions (Chica et al., 2018). In competitive markets, the before commented rules describe typical consumer activities and how consumers adopt products or services by using diffusion information models (Watts and Dodds, 2007).

In this contribution, we propose the use of evolutionary multiobjective optimization (EMO) (Deb, 2001; Coello et al., 2007) together with an ABM market simulation model to locate and target those influentials which are used as promoters of VM campaigns. The ABM simulation model is used to evaluate the fitness of individuals (i.e., the effectiveness of the VM campaigns). The market model incorporates realistic consumer behavior such as an awareness filtering process (Robles et al., 2016) and is inspired by the well-known Consumat marketing model (Janssen and Jager, 2003). Our novel definition of the multiobjective optimization problem jointly optimizes the sales revenue and costs of the targeting campaign using local social network measures (Newman et al., 2006)

to make the problem more realistic. Sales revenue and campaigns' costs are natural conflicting objectives because increasing the total sales of a brand normally requires a higher budget for the marketing campaigns. Although there are single-objective optimization problem formulations for VM (Stonedahl et al., 2010; Schlereth et al., 2013), there is not any multiobjective approach to optimize these two conflicting objectives at the same time.

From the vast family of EMO algorithms, we use the NSGA-II (Deb et al., 2002) and MOEA/D (Zhang and Li, 2007) as they are two well-known algorithms, successfully applied to a large number of multiobjective optimization problems. Our experimentation comprises different marketing scenarios by using five different well-known artificial social network topologies. These five social networks represent the connections between consumers in the market and are: a real network based on the email communications between University scholars (Guimera et al., 2003), a scale-free network (Barabási and Albert, 1999), a Watts-Strogatz small world network (Watts and Strogatz, 1998), an Erdös-Rényi random network, and a regular lattice network. Additionally and to better validate the scalability of our approach, we use a real data-generated social network of 20,000 nodes, built from the users of a freemium app (Chica and Rand, 2017). We compare the quality of the non-dominated solutions found by the EMO algorithms with a single-objective genetic algorithm and a greedy algorithm. The design of our developed experimentation also allows us to analyze the influence of the social network topologies on the solutions and algorithms' performance. Finally, we explore the marketing policies obtained from the set of non-dominated solutions returned by the EMO algorithms.

2. Background

Targeting influentials on social networks was first presented as an optimization problem (i.e., influence maximization) in (Domingos and Richardson, 2001). Later, Kempe et al. showed that the influence maximization is essentially an NP-hard optimization problem and proposed a greedy algorithm to solve it although the algorithm required global knowledge of the network (Kempe et al., 2003). Other authors used a descriptive approach to VM problems (Leskovec et al., 2007). Meanwhile, marketing scholars used ABM to describe adoption processes and characterize the individuals having the greatest effect on global adoption (Schlereth et al., 2013), showing that the role of hubs or influentials are important for new product adoption processes. Some other authors have

improved the approximation to the influence maximization problem by using metaheuristics such as genetic algorithms (GAs) (Stonedahl et al., 2010; Lahiri and Cebrian, 2010) or swarm intelligence algorithms (ŞİMŞEK and Resul, 2018). Other authors have addressed the influence maximization problem from a different perspective. Their approaches consist on the use of algorithms such as Deep Learning (Keikha et al., 2019), or community detection (Banerjee et al., 2019) to explore the underlying structures of social networks and then use diffusion methods to maximize the influence across communities or between interconnected networks.

Among the different approaches to the influence maximization problem studied by other authors the most similar to ours is the presented in (Stonedahl et al., 2010), where authors proposed prescriptive policies for seeding campaigns handling with an affordable knowledge level to marketers. The latter proposal used GAs to generate VM campaigns by using local and global social network analysis measures.

Although EMO has been widely used in the social network domain, most of the efforts related to our work were carried out for other related problems such as community detection. Just recently, Gunasekara et al. considered EMO to identify key players in large social networks (Gunasekara et al., 2015). Up to our knowledge, there is not any previous multiobjective evolutionary research dealing with conflicting objectives when finding and targeting influentials in VM campaigns. Normally, there is a simplification of the inherent two-objective problem (maximizing revenue and minimizing costs) into a single objective (see (Stonedahl et al., 2010)). In our case, we address the targeting problem from a multiobjective perspective by using EMO algorithms to jointly deal with the two key campaign objectives.

3. The Agent-based Market Simulation Model

We present the ABM framework to recreate the artificial market to be used when evaluating the fitness of the VM campaigns. First, the main variables of the market are shown in Sections 3.1 and 3.2. Section 3.3 details the social network influence in the consumers. Finally, consumers' decision making heuristics are listed in Section 3.4.

3.1. Consumers and Products of the Artificial Market

Our model is an extension of the one proposed in (Robles et al., 2016), inspired by the well-known Consumat model (Janssen and Jager, 2003). We use an ABM to recreate an artificial market

with a population of I consumer agents, connected by an artificial social network, and J competing market products. The simulation has T simulation steps and follows a synchronous update (Chica and Rand, 2017). Every agent $i \in \{1, 2, ..., I\}$ has a probability $p_b \in [0, 1]$ to buy one product of the market $j \in \{1, 2, ..., J\}$ at every simulation step $t \in \{1, 2, ..., T\}$. When a drawn random value is lower than p_b then agent i makes a choice from the set of J products at step t. An array $X_i = \{x_i^1, x_i^2, ..., x_i^T\}$ with $x_i^t \in \{0, 1, ..., J\}$ stores selected products by agent i or sets 0 when the agent does not buy.

The J products of the market differ in a set of characteristics which define their features (e.g., price, quality, or design). For the sake of simplicity, we model these characteristics by a single variable $d_j \in [0, 1]$ which globally describes the characteristics of product j. Each agent i has a personal preference $\pi_i \in [0, 1]$ about its needs in the market. Equation 1 calculates utility $\mu_{ij} \in [0, 1]$ of product j for consumer agent i, based on its preferences and product's characteristics:

$$\mu_{ij} = 1 - |d_j - \pi_i|. \tag{1}$$

3.2. Products' Awareness

The model also incorporates a consumer awareness behavior for a more realistic design (Robles et al., 2016). Agents do not have a complete knowledge of the whole set of available products but certain knowledge limitations about them, based on the idea of bounded human rationality (Chica et al., 2018). Thanks to this awareness mechanism, the model restricts the agents' purchase decisions to only the products they are aware of, which varies overtime.

A binary variable $\theta_{ij}^t \in \{0, 1\}$ models if agent i is aware of product j ($\theta_{ij}^t = 1$) or is not ($\theta_{ij}^t = 0$) at time step t. Therefore, at each step, agent i will choose a product from a reduced set of products $J_i^t = \{j \in J : \theta_{ij}^t = 1\}$.

Values for θ_{ij}^0 are randomly initialized at the beginning of the simulation. During the simulation, an agent can also forget their products' awareness by a deactivation process to mimic reality. This process is modeled by a deactivation variable $\delta \in [0, 1]$, which is common for all the agents. At every step, agent i forgets the awareness of product j if a drawn random value is lower than δ. Otherwise, the agent will continue being aware of product j. High values of δ mean agents easily forget the awareness information they obtained in the past. The deactivation process is not applied for the product the agent is consuming at a given time-step.

3.3. Peer Influence by the Social Network

Agents are connected through an artificial social network where each node is a consumer and edges represent the connections of the agents with their direct contacts. Interactions between consumers occur during all the steps of the simulation and these interactions make agents have complete knowledge about the consuming decisions of their direct contacts. This information exchange influences the buying decisions of the agents at every step of the simulation by the next three ways.

3.3.1. Biased product utility

First, the direct contacts of agent i modify its personal utility μ_{ij} by means of a biased temporal utility u_{ij}^t:

$$u_{ij}^t = \beta\mu_{ij} + (1-\beta)\gamma_{ij}^t, \tag{2}$$

where $\gamma_{ij}^t \in [0,1]$ is the fraction of direct contacts of agent i in the social network who consumed product j in their last choice. Biased utility u_{ij}^t is also modulated by $\beta \in [0,1]$, a social peer influence parameter for the whole artificial market. High values of β mean consumers are highly influenced by their contacts in the social network. Low values of β means the personal preferences when adopting a product j are more important (e.g., a market with more innovative and unconventional consumers).

3.3.2. Uncertainty about the decision

Second, agent's contacts in the social network also influence the uncertainty of an agent when buying. An agent will be less convinced about making its decision just based on the utility value u_{ij}^t if this agent has a high number of direct contacts who consumed different products in their last buying decision (i.e., at time step $t-1$). Agents' variable $\phi_i^t \in [0,1]$ calculates this uncertainty, based on the direct contacts in the social network:

$$\phi_i^t = \beta(1-\gamma_{\neq x_i^{t-1}}^t), \tag{3}$$

where $\gamma_{\neq x_i^{t-1}}^t$ is the fraction of contacts of agent i who have bought a different product than the one bought by i in the previous time step. Social parameter β modulates how uncertain agents

are when making their decisions. If the market has a high social component (β), agents will have a higher uncertainty when buying if contacts bought a different product.

3.3.3. Awareness diffusion

Third, the model has an awareness diffusion probability $p_\theta \in [0, 1]$ that enables consumer agent i to talk about product j when (s)he is aware of it. To determine if a consumer talks about a product, a random number is drawn in $[0, 1]$. If this number does not exceed the p_θ value, consumer agent i will spread its awareness about product j to its direct contacts and these contacts will also be aware of the products. Notice that, if an agent is aware of a product, it can talk about it with its direct contacts in the social network regardless of its buying decision at the same time step.

3.4. Decision Making and Heuristic Rules

Agents can make decisions in different ways depending on their level of satisfaction and uncertainty. We consider four different heuristics, as done in (Janssen and Jager, 2003), to select a product from the set of products agents are aware of (i.e., sub-set $J_i^t \subseteq J$). Agents dynamically choose the heuristic by comparing their internal variables with respect to two thresholds, common to all the agents in the market. The first threshold is U and defines the minimum satisfaction level for the agents' utility. The second threshold is Φ and states agents' uncertainty tolerance level. These two thresholds specify the conditions of the artificial market and define the type of heuristic together with biased utility u_{ij}^t and uncertainty value ϕ_i^t. The following four heuristics are chosen by each agent at each step t:

- *Repetition*: An agent uses this heuristic when it is satisfied and certain about its selection (i.e., $u_{ij}^t \geq U$ and $\phi_i^t \leq \Phi$). This is a repeat buying phenomenon related to the loyalty and satisfaction of a consumer to a brand (Ehrenberg et al., 2004). Thus, agent i repeats buying the same product:

$$x_i^t = x_i^{t-1}. \tag{4}$$

- *Deliberation*: A consumer agent is not satisfied with its last purchase but is certain (i.e., $u_{ij}^t < U$ and $\phi_i^t \leq \Phi$). Here, agent i evaluates its utility u_{ij}^t for all the products in the market

it is aware of and make a probabilistic decision using a logit function. Using this function, product j has a probability to be chosen by agent i:

$$prob_i^t(j) = \frac{e^{u_{ij}^t}}{\sum_{k=0}^{M_i^t} e^{u_{ik}^t}}, \forall j \in M_i^t. \quad (5)$$

- *Imitation*: An agent is satisfied but uncertain about its choice (i.e., $u_{ij}^t \geq U$ and $\phi_i^t > \Phi$). Agent i evaluates the products it is aware of and the fraction of direct contacts who bought them. A logit function is again used to calculate the probabilities for each product j. The product with the largest share among the neighbors has the highest probability to be chosen:

$$prob_i^t(j) = \frac{e^{2\gamma_{ij}^t}}{\sum_{k=0}^{M_i^t} e^{2\gamma_{ik}^t}}, \forall j \in M_i^t. \quad (6)$$

- *Social comparison*: An agent is neither satisfied nor certain about its choice (i.e., $u_{ij}^t < U$ and $\phi_i^t > \Phi$). The agent only evaluates the products that were bought by its direct contacts in their last purchase. We use a similar logit function than in Equation 5 but with a more reduced set of products $J_i^{t'}$ (i.e., another screening pre-process to filter alternatives such as awareness):

$$prob_i^t(j) = \frac{e^{u_{ij}^t}}{\sum_{k=0}^{M_i^{t'}} e^{u_{ik}^t}}, \forall j \in M_i^{t'}. \quad (7)$$

4. Multiobjective VM Optimization Problem

4.1. Decision Variables

The goal of this problem is finding the optimal number of influentials for one of the brands of the market and their locations in the social network. In order to find these influentials we combine local social network metrics to evaluate all the consumers of the population and target those ones having the highest evaluation. Those consumer agents selected as influentials are targeted by free samples of the product to simulate a positive diffusion of their opinions to their direct contacts in the social network.

Therefore, a solution to the optimization problem is how to measure when a consumer is an influential and how many of them are targeted. Then, each influential will talk about the product

of the VM campaign and influence the decisions of its direct contacts [1]. We follow an adaptation of the approach defined in (Stonedahl et al., 2010) to calculate the influential nodes in the social network (i.e., consumer agents). However, we do not take into account global social network metrics as it is unrealistic to have knowledge about the structure of the global network when launching the VM campaign. Our three local social network measures are:

- Degree: Number of direct neighbors in the social network of an agent i, normalized by the maximum possible value $I-1$. Nodes with a higher degree will influence more neighbors by directly spreading the information to more possible new adopters.

- 2-steps: It is a measure corresponding to the number of nodes that are reachable within two steps from node of agent i in the social network. It is also normalized by $I-1$.

- Clustering coefficient (cc): It quantifies how much the neighbors of a node are interconnected among them (i.e., it is a local density measure) (Watts and Dodds, 2007). If the node is clustered as a clique (full graph) its cc value is maximum and equal to 1. Small cc values indicate a low clustered node in the social network. When a node has a high cc value, it will presumably generate a quicker adoption through a high diffusion speed in the social network.

We use a weighted combination of the latter social network measures to obtain a real value $q(i)$ for each node i which indicates the degree of influence of each agent within the social network (higher values mean higher influence):

$$q(i) = \omega_d \frac{degree(i)}{N-1} + \omega_{2s} \frac{\text{2-steps(i)}}{N-1} + \omega_{cc}(1 - cc(i)). \tag{8}$$

These weights ω_d, ω_{2s} and ω_{cc} are decision variables within $[0,1]$. We also incorporate a fourth decision variable $s \in \{1, 2, ..., S_{max}\}$. s is the number of influentials to target with product k of the VM campaign and S_{max} is the maximum number influentials to target from the total number of consumer agents I. The process to identify the set of influentials (S) is to first evaluate all the

[1] In our artificial market, this process is implemented when consumers who bought a given product positively influence their contacts because of the biased utility and heuristic rules defined in Sections 3.3 and 3.4, respectively. Notice also that our multiobjective problem is independent from the specific ABM design and can be adapted to use any other artificial market approach.

agents by calculating their $q(i)$ values. Later, we sort them in descending order and pick the first s agents to be targeted in the campaign.

4.2. Objective Functions

The ABM framework defined in Section 3 is used to simulate the market and evaluate the effectiveness of the VM campaigns encoded in the individuals after targeting s influentials with free samples of product $k \in J$ (i.e., the one promoted by the VM campaign). This effectiveness is measured by two objectives: a) maximizing the number of sales of product k (i.e., the highest revenue), and b) minimizing the costs of the campaign. Notice that influentials of set S cannot purchase products during the simulation as they obtain free product samples when they need to buy according to probability p_b. The second objective is the cost of the campaign, associated to the cost of the free samples of the product provided to the selected influentials over time.

We use the net present value (NPV) to calculate both objectives. NPV is a measure of a company's revenue when consumers buy the product the moment in which the product has been commercialized rather than several months after product's commercialization. Mathematically, NPV is defined as $\sum_{t=1}^{\infty} |A^t| p \lambda^t$ where $|A^t|$ is the number of adopters or buyers of product k at time t, p is the profit for adoption of a product, and λ is the discount factor. This discount factor means that an earlier purchase is preferable because of the cumulative effect of several factors such as opportunity costs and the potential need of setting lower prices over time to be more competitive.

The first objective $f^0(x)$ is directly the NPV obtained accumulating the revenue of the purchases of product k at each time step t. The number of purchases obtained from the set of adopters $|A_k^t|$ is multiplied by discount factor λ related to the time of adoption as we set $p = 1$ for each sold product:

$$f^0(x) = NPV(A_k) = \sum_{t=1}^{T} |A_k^t| \lambda^t. \tag{9}$$

The second objective $f^1(x)$ is the cost of targeting the influentials of set S with free product k samples when they need to buy according to probability p_b. Note that this objective is clearly in conflict with $f^0(x)$ because increasing the number of influentials s consequently decreases the number of possible adopters of the product (A_k^t). We define $S_{x^t=k}$ as the set of influentials who are targeted at time step t. Campaign costs are computed multiplying the number of provided samples by a manufacturing cost of $\frac{1}{10}$ and the discount factor:

$$f^1(x) = Cost(S) = \frac{1}{10} \sum_{t=1}^{T} |S_{x^t=k}| \lambda^t. \quad (10)$$

5. Evolutionary Multiobjective Optimization Methods for Influentials Selection

We first present the common methods' design issues in Section 5.1. From the vast EMO family, we choose two of the best-known algorithms, NSGA-II (Deb et al., 2002) and MOEA/D (Zhang and Li, 2007), which respectively follow two different EMO approaches (i.e., Pareto-based selection and objective function decomposition). We also use a single-objective GA guided by a linear combination of the two objectives and a greedy approach as a baseline. These methods are briefly described in Sections 5.3, 5.4, and 5.5.

5.1. Common Design Issues of the Methods

5.1.1. Representation

Each individual of the population has four genes which correspond to the four decision variables defined in Section 4.1. The first three genes are real-coded and represent the weights of the social network measures $(\omega_d, \omega_{2s}, \omega_{cc})$. The fourth gene is an integer value which states the number of influentials s to target in the campaign. The first population is generated at random.

5.1.2. Selection

The three evolutionary algorithms follow a steady-state approach and use a binary tournament selection method. This tournament selection picks two individuals at random from the population and compares their objective values to keep the best. We repeat this operation twice until two individuals are selected at each generation and apply the genetic operators to them.

5.1.3. Crossover operator

The algorithms use a BLX-α crossover operator to cross over offspring with a probability $p_c \in [0, 1]$. The operator works as follows. Given two parents $C^1 = (C_1^1, \ldots, C_n^1)$ and $C^2 = (C_1^2, \ldots, C_n^2)$, BLX-$\alpha$ generates two offspring $D^1 = (d_1^1, \ldots, d_i^1, \ldots, d_n^1)$ and $D^2 = (d_1^2, \ldots, d_i^2, \ldots, d_n^2)$, where each gene d_i is randomly generated in the range $[C_i^{min} - \alpha I, C_i^{max} + \alpha I]$, with $C_i^{max} = max\{C_i^1, C_i^2\}$, $C_i^{min} = min\{C_i^1, C_i^2\}$, and $I = C_i^{max} - C_i^{min}$.

5.1.4. Mutation operator

It is applied to each individual with a probability $p_m \in [0, 1]$ to reset one of its genes at random within the interval of the specific gene. This mutation operator is a real-coded reset for the first three genes and an integer-coded reset for the fourth gene.

5.2. Pareto dominance

The set of solutions composing the Pareto fronts found by algorithms follows a Pareto dominance schema. The concept of Pareto domination establishes that an individual A Pareto dominates an individual B if and only if A is at least as good as B in all objectives, and is superior to B in at least one objective. There are three main cases where A might not Pareto-dominate B:

- B Pareto-dominates A
- A and B have exactly the same objective values for all objectives.
- A is superior to B in some objectives, but B is superior to A in other objectives.

The Pareto front showing algorithms' solutions only contains the solutions which are not Pareto dominated. Every Pareto dominated solution is discarded.

5.3. NSGA-II Description

NSGA-II (Deb et al., 2002) is the most well-known EMO algorithm. In this algorithm, the offspring population Q_t is created from the parents population P_t (also composed of N individuals) by applying the considered selection mechanism and genetic operators. Afterwards, the two populations are joined to form a single intermediate population R_t with two times the population size ($2N$). Then, the R_t population is partitioned into several fronts according to their dominance degrees by using a non-dominated sorting process.

Once the non-dominated sorting process is completed, the new population is generated from the configurations of the non-dominated fronts. This new population is first built with the best non-dominated front, continues with the solutions of the second front, the third, and so on until the N available positions are occupied. Every member of each dominance group is inserted into the new population until there is not any free place in the whole front. For the case of the first front that cannot be fully allocated in the new population, a diversity preservation mechanism is

applied. A crowding distance calculation is made for every individual in that last considered front and the most diverse individual are those included in the new population.

5.4. MOEA/D Description

MOEA/D (Zhang and Li, 2007) is an EMO algorithm based on explicitly decomposing a multiobjective optimization problem into M scalar optimization sub-problems. The algorithm solves these sub-problems simultaneously by evolving a population of solutions. At each generation, the population is comprised by the best solution found so far for each sub-problem. The neighborhood relations among these sub-problems are defined based on the distances between their aggregation coefficient vectors. Each sub-problem is optimized in MOEA/D by only using information from its neighboring sub-problems.

There are several approaches for converting the problem of approximating a Pareto front into a number of scalar optimization problems. We use the Tchebycheff approach (g^{te}) for our experiments due to the good results in terms of feasibility and efficiency obtained in (Zhang and Li, 2007). Mathematically, let $\lambda = (\lambda_1, \ldots, \lambda_m)^W$ be a weight vector and m be the number of sub-problems, i.e., $\lambda_i >= 0$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^{m} \lambda_i = 1$. The Tchebycheff approach considers a scalar optimization problem in the form:

$$\begin{aligned} \min g^{te}(x|\lambda, z^*) = \min_{1 \leqslant i \leqslant m} \{\lambda_i | f_i(x) - z_i^*|\} \\ \text{subject to } x \in \Omega, \end{aligned} \tag{11}$$

where $z^* = (z_1^*, \ldots, z_m^*)^W$ is the reference point. This point is initialized as the lowest value of the objective function f_i found in the initial population. For each Pareto optimal point x^* there is a weight vector λ such that x^* is the optimal solution of Equation 11 if it is a Pareto optimal solution of the objective function. Therefore, we can obtain different Pareto optimal solutions by altering the weight vector.

5.5. Weighted Single-objective Genetic Algorithm

We have implemented a single-objective approach to solve the multiobjective VM problem by linearly combining the two objectives, $f^0(x)$ and $f^1(x)$, in a single-objective fitness value. We use a parameter $\rho \in [0, 1]$ to weight both problem objectives by $f(x) = \rho f^0(x) + (1 - \rho) f^1(x)$.

An aggregated Pareto front approximation for this GA is obtained by joining the non-dominated set of solutions found by the algorithm with 11 values for the linear combination parameter ρ (from 0 to 1 with a step of 0.1). These values are able to cover a diverse solutions range and allow us to obtain a Pareto set approximation to be compared with those obtained by NSGA-II and MOEA/D, allowing a fair comparison.

5.6. Greedy Algorithm

Finally, we have extended our experimentation with a greedy algorithm for the multiobjective VM problem. For this greedy approach we define four weight combinations of values for the *degree*, $2-$*steps* and *cc* network measures. The used weight combinations are the following ones:

- $[1.0, 0.0, 0.0]$: Influentials are only sorted and targeted by their degree measure values.
- $[0.0, 1.0, 0.0]$: Influentials are only sorted and targeted by their 2-steps measure values.
- $[0.0, 0.0, 1.0]$: Influentials are only sorted and targeted by their clustering coefficient values.
- $[0.33, 0.33, 0.33]$: Influentials are equally sorted and targeted by considering their degree, 2-step, and clustering coefficient.

The algorithm generates sets of s influentials in a social network, from 1 to S_{max}, for each of the latter weight combinations. After the generation of all the solutions, we generate an aggregated Pareto front with the non-dominated solutions from those generated by the algorithm, for each set of weights and number of targeted influentials s.

6. Experiments and Analysis of Results

Section 6.1 reports the experimental setup. Section 6.2 reviews the EMO indicators. Sections 6.3 and 6.4 analyze the results and marketing insights from the EMO solutions.

6.1. Experimental Setup

We set a 10% discount factor ($\lambda = 0.9$) as used in (Stonedahl et al., 2010) for the objective functions (Equations 9 and 10). All the EMO algorithms are run 30 times with independent random seeds. Their population size P is of 200 individuals. The algorithms' populations evolve during 100 generations. Crossover and mutations probabilities, p_c and p_m, are equal to 0.6 and

0.2, respectively, and $\alpha = 0.5$ for the BLX crossover operator. A neighborhood of 40 individuals is used for MOEA/D because of the high population size ($P = 200$). The maximum number of agents to target S is defined as as $0.1I$ so the maximum number of agents acting as influentials in a simulation will be as maximum a 40% of the total consumer agents I.

The EMO algorithms use the ABM for artificial markets to evaluate the revenue and costs of the target VM campaigns. We set the parameters of the model as follows. There are $J = 10$ products available in the market including the product k of the VM campaign. The simulations runs for $T = 365$ time steps and the total number of agents I is set according to the number of nodes of each of the five artificial social networks used in the experimentation.

Table 1 shows the features of the five artificial social network topologies and the case study. The first artificial social network topology, EMAIL, is a real email network of the Rovira i Virgili University in Spain (Guimera et al., 2003). The rest of the artificial networks were generated using similar features than those of the EMAIL network (i.e., an average degree of 10 and a density of 0.01). For the scale-free network (SF) we used the Barabasi-Albert preferential attachment algorithm (Barabási and Albert, 1999) with $m = 2$ (number of new connections for each added node). For the random network (RAND), we connected each node at random to other nodes in the network by maintaining the average degree of 10 ($p = 0.01$). For the small world network (SW), we used 0.2 as the rewiring probability in the Watts-Strogatz generation algorithm (Watts and Strogatz, 1998). We use 4 edges per node with periodic boundary conditions for the regular lattice network (REG). Finally, the data-generated network of the case study (APP) presents a heavily bi-modal degree distribution (Chica and Rand, 2017). It means there is a numerous group of users with one or two connections and another numerous group with a high number of connections (i.e., around 100).

The social influence parameter β for the ABM simulation is set to 0.5. We set thresholds U and Φ to 0.5, which define agents' uncertainty and the minimum utility in the market. Agents' probabilities to buy a product and talk about them are $p_b = 0.8$ and $p_\theta = 0.5$, respectively, while the awareness deactivation δ equals 0.2. Agents' product awareness θ_{ij} are randomly initialized at the beginning of the simulation. Finally, the values of the objectives are averaged for the 30 Monte Carlo independent runs of the ABM simulation developed for each individual' evaluation. Please notice that we tested a higher number of values for the latter model's parameters in a pre-

liminary experimentation and no significant differences were found in the evolutionary algorithms' performance.

6.2. Multiobjective Performance Indicators

We consider two unary and two binary multiobjective performance indicators (Zitzler et al., 2003; Coello et al., 2007; Deb, 2001) to evaluate the performance of the algorithms. The first unary indicator is the cardinality of each Pareto set approximation (i.e., the number of solutions composing it). The second unary indicator is the hyper-volume ratio (HVR) (Coello et al., 2007) which jointly measures the distribution and convergence of a Pareto front approximation. The HVR is defined as follows:

$$HVR = \frac{HV(P)}{HV(P^*)}, \tag{12}$$

where $HV(P)$ and $HV(P^*)$ are the volume of the approximate Pareto front and the true Pareto front, respectively. When HVR is 1, the Pareto front approximation and the true Pareto front are equal. HVR values lower than 1 indicate a generated Pareto front that is not as good as the true Pareto front.

Since we are working with a real-world problems, one difficulty is the lack of information about the true Pareto fronts which makes the indicator computation difficult. To overcome the latter drawback and be able to calculate the HVR indicator we consider a pseudo-optimal Pareto front instead. A pseudo-optimal Pareto front is an approximation to the true Pareto front obtained by merging all the Pareto front approximations generated for each problem instance by each algorithm in every independent run.

Binary indicators compare the performance of two different EMO algorithms by comparing the Pareto front approximations generated by each of them. We consider two of them, the I_ϵ indicator and the set coverage indicator (C). The I_ϵ indicator (Zitzler et al., 2003) is a quality assessment method for multiobjective optimization that avoids particular difficulties of unary and classical methods. We chose the multiplicative variant of the I_ϵ indicator. Given two Pareto front approximations, P and Q, $I_\epsilon(P, Q)$ is calculated as follows:

$$I_\epsilon(P, Q) = \inf_{\epsilon \in \Re} \forall z^2 \in Q, \exists z^1 \in P : z^1 \leq_\epsilon z^2, \tag{13}$$

where $z^1 \leq_\epsilon z^2$ if $z_i^1 \leq_\epsilon z_i^2, \forall i \in 1, \ldots, o$, with o being the number of objectives. Assuming a minimization problem, $I_\epsilon(P,Q) < I_\epsilon(Q,P)$ indicates that the P set is better than the Q set because the minimum ϵ value needed so that approximation set P $\epsilon - dominates$ Q is smaller than the ϵ value needed for Q to ϵ-dominate P.

The second binary indicator, the set coverage indicator C (Zitzler et al., 2003), is computed as follows:

$$C(P,Q) = \frac{|q \in Q; \exists p \in P : p \leq q|}{|Q|}, \tag{14}$$

where $p \leq q$ indicates solution p of Pareto front approximation P weakly dominates solution q of Pareto front Q, in a minimization problem. Hence, $C(P,Q) = 1$ means that all the solutions in Q are either dominated by or equal to the solutions in P. $C(P,Q) = 0$ means no solution in Q is covered by any one in P. Both $C(P,Q)$ and $C(Q,P)$ have to be considered since $C(P,Q)$ is not necessarily equal to $1 - C(Q,P)$.

6.3. Performance analysis on artificial social networks

We show the results obtained by NSGA-II, MOEA/D, the single-objective GA, and the greedy algorithm for the five artificial social network topologies: EMAIL, SF, RAND, SW and REG. We first discuss the values obtained by the EMO algorithms for each multiobjective performance indicator. Then, we show how each algorithm covers the search space and its contribution to the best set of solutions found for each scenario by analyzing their Pareto front approximations. We will discuss the results starting from those network topologies with higher diffusion speed.

Table 2 shows the values for the HVR and the cardinality unary performance indicators. Table 3 reports the values for the C indicator. Finally, values for I_ϵ performance indicator are shown in Table 4. In all the tables, we show the average and standard deviation values (in brackets) for the 30 runs of the algorithms. The best values for each indicator are in bold.

MOEA/D obtains the highest HVR values in three of the five social network topologies (see Table 2). On the contrary, NSGA-II shows the best results in the other two topologies. The single-objective GA shows poor results compared to multiobjective algorithms but performs correctly despite not being a pure multiobjective method. Finally, the greedy algorithm is the worst algorithm in all the scenarios and its far form the rest of the EMO algorithm performances. Results show that differences among MOEA/D, NSGA-II, and the single-objective GA are clear depending

on network topologies with different diffusion speeds. MOEA/D performs between a 1.5% and 2% better than the other two algorithms in the higher information diffusion speed network topologies (EMAIL and SF). Meanwhile, the single-objective GA obtains the worst values for those networks. In the RAND network, where information diffusion is slower than in SF, MOEA/D is still the best algorithm but with a low difference with respect to NSGA-II (around 0.03%) and the GA (0.8%). Finally, NSGA-II obtains the best HVR values for SW and REG networks, the networks with the lowest diffusion speed.

NSGA-II is the EMO algorithm that returns the highest number of non-dominated solutions in the Pareto fronts (i.e., cardinality indicator). MOEA/D is the second one in cardinality values, returning around the half of the solutions obtained by NSGA-II but in the REG network scenario (in this case, similar cardinality values are obtained). The greedy approach obtains a good number of non-dominated solutions into its Pareto fronts but is far from NSGA-II cardinality values. The single-objective GA has the lowest cardinality value as it obtains less than ten non-dominated solutions in every scenario.

C indicator values of Table 3 draw the same conclusions than those observed in Table 2. MOEA/D obtains the best C values in the EMAIL, SF, and RAND networks. On the contrary, MOEA/D solutions are mainly covered by NSGA-II solutions for the SW and REG networks, the two scenarios with lower information diffusion speed. The single-objective GA shows a low coverage value when comparing with pure EMO methods. Only when considering the SW network, the single-objective GA cover around the 20% of the non-dominated solutions found by MOEA/D and NSGA-II. The greedy algorithm is the worst performing algorithm in every scenario, being covered for all the other EMO algorithms Pareto fronts in all scenarios.

I_ϵ values are shown in Table 4. Again, the algorithms have the same behavior than with the previous indicators of Tables 2 and 3. MOEA/D Pareto front approximations need to be multiplied by lower values (with 1.03 being the highest one) to achieve the set of NSGA-II solutions for the first three scenarios (i.e., EMAIL, SF, and RAND). Meanwhile, NSGA-II needs a greater ϵ value to approximate the MOEA/D Pareto fronts. The opposite conclusion is derived for the case of SW and REG networks, where NSGA-II reports lower I_ϵ values. The single-objective GA presents the high I_ϵ values but is far from the greedy algorithm which is the worst performing algorithm again.

We now focus our analysis on the attainment surfaces of the Pareto front approximations

obtained by the algorithms. Figures 1, 2, and 3 show the aggregated Pareto front approximations for three of the five social network topologies (i.e., EMAIL, SF and REG). We do not show the other two attainment surfaces because of the lack of space. In the attainment surfaces each algorithm is represented by a color and a shape in the figures: green squares for MOEA/D solutions, blue triangles for NSGA-II solutions, red crosses for single-objective GA, and orange diamonds for greedy algorithm solutions. The solutions of the pseudo-optimal Pareto front built from all the runs are represented by black circles. Notice that the previous indicators' analysis used the independent Pareto front approximations obtained by the algorithms in the 30 runs. However, these attainment surfaces show the aggregated Pareto fronts of the different algorithms in the 30 runs for a better visualization.

MOEA/D provided the best solutions for the EMAIL network (see Figure 1). Only one NSGA-II solution dominates the MOEA/D ones in the upper part of the pseudo-optimal Pareto front. The aggregated Pareto front approximation of the single-objective GA only contributes with two solutions to the pseudo-optimal Pareto front (both in its bottom part). The remaining solutions obtained by the single-objective GA are dominated by MOEA/D and NSGA-II solutions. All the greedy solutions are Pareto dominated by MOEA/D, NSGA-II and single-objective GA algorithms.

MOEA/D is again the best algorithm for the SF network (Figure 2). 41 of the 49 solutions of the pseudo-optimal Pareto front come from the runs of the MOEA/D. Only a few solutions obtained by NSGA-II dominate MOEA/D ones and are those with revenue in $[1, 500]$ and costs in $[1, 20]$, located in the bottom left part of the Pareto front. The majority of the single-objective GA solutions are dominated by NSGA-II and MOEA/D. We clearly see that MOEA/D achieves better results and provides with a higher number of solutions to the pseudo-optimal Pareto front. The performance of MOEA/D in these first two scenarios shows its potential when having network topologies with a power-law degree distribution having "hubs" that increase the information diffusion speed.

Finally, Figure 3 shows the attainment surfaces for the VM campaigns when using the REG network, the one with the lowest information diffusion speed. In this scenario, we can clearly see that NSGA-II outperforms MOEA/D by generating a significantly higher number of pseudo-optimal Pareto front solutions. The fronts obtained by the two EMO algorithms largely dominate the single-objective GA Pareto front approximation. Only two single-objective GA solutions form the pseudo-optimal Pareto front in its central part. The greedy algorithm does not have solutions into

pseudo-optimal Pareto front. Its Pareto front is totally dominated by the Pareto fronts obtained by the other algorithms.

6.4. Analysis of the Returned Marketing Campaigns

We analyzed in the previous Section 6.3 the behavior of the algorithms for the five network scenarios. In this section we validate the multiobjective VM proposal from a marketing perspective. Therefore, we analyze here the quality of the marketing campaigns designed by the three evolutionary algorithms. To ease the analysis we only consider the EMAIL and REG network scenarios as they represent the most different social network topologies.

Table 5 shows the composition of three different solutions selected from the Pareto front approximation generated by each algorithm for each scenario. Two of them are those with the best value for each individual objective (revenue and cost of the VM campaign). The third one presents a trade-off between these two objectives. The trade-off solution is chosen as follows. We compute 1000 random values for weight $\rho \in [0, 1]$ and the aggregated value of the weighted combination of the two objectives (campaign revenue and costs) is taken as $f(x) = \rho NPV(A) + (1 - \rho)Cost(S)$ for each solution. The solution with the highest aggregated value $f(x)$ is selected. Table 5 shows the genotypes' values of the individuals. Remind that the first three values correspond to ω weights associated to the local social network analysis measures while the fourth is the number of influentials s.

We can find interesting differences in the composition of the solutions in Table 5. First, we can see that the algorithms return solutions with different weights for the social network metrics to define the influentials targeting strategy (e.g., see the composition of the highest revenue individuals found by MOEA/D and NSGA-II). These weights are adapted to the characteristics of the social network topology to achieve a higher diffusion of the campaign. For example, in the EMAIL network, we see that the weights of measures related to the degree of the target nodes have a higher value to use the "hubs" of the network as influentials. We can also see that there are differences between individuals depending on the objectives. As expected, solutions will have a higher or lower number of influentials depending on the revenue and cost of the campaign, as well as depending on the network topology (e.g., a significantly larger number is required to get good revenue in the smallest information diffusion speed REG network).

One can also see that, for the EMAIL network, solutions have higher weights' values, both for

high revenue and low cost solutions. We also see two main differences when comparing the highest revenue and lowest cost solutions for the EMAIL network. The first is related to the number of consumers selected as influentials, which is higher for the solutions with the highest revenue as it enables to quickly spread the product through the social network. The second difference is that, for the highest revenue solution, the degree of the contacts reachable through two steps (ω_{2s}) is weighted after the degree (ω_d) or the clustering coefficient. This is a direct consequence of the power law-distribution of the EMAIL network topology: by using a combination of a low number of targeted consumers with a high degree and a high ω_{2s} value, the VM campaign targets those consumers which are "hubs" in the social network (i.e., those who can quickly spread positive information about the product).

The composition of the solutions for the REG network has less variability than those for the EMAIL network. Solutions for REG show similar ω_d, ω_{2s}, and ω_{cc} values, and high s values, both for the highest revenue and lowest cost solutions. In this case, the main difference between these solutions resides in the number of targeted consumers s and its distribution in the social network. The characteristics of the REG network make the algorithms find the best distribution for the influentials through the network topology by also selecting a high number of seeds to obtain solutions with high revenue. Of course, there is a low number of targeted nodes when having a low cost campaign solution. In fact, it is the lowest possible number (1) as a consequence of the very slow information diffusion associated to the network topology.

6.5. Case study on a data-generated social network

As already mentioned, we extend the set of social networks of the experimentation by adding a real case study. This case study allows us to show that the EMO algorithms can be extended to large real social networks and to analyze if they behave similarly than with the five artificial social networks. We use the same indicators to measure the algorithms' performance (Table 6) and we show the attainment surfaces which contain the set of non-dominated solutions found by MOEA/D, NSGA-II and single-objective GA EMO algorithms (Figure 4). Here the greedy algorithm is not used as we consider to discard it considering the poor performance it showed in Section 6.3.

By taking into account the results of Table 6 we can easily see the superiority of MOEA/D over NSGA-II and GA. MOEA/D obtains the best values for all the multiobjective performance indicators except for cardinality, where NSGA-II is the best performing algorithm. In this network,

MOEA/D has much better I_ϵ and C values than NSGA-II and the single-objective GA (for instance, we see that MOEA/D covers around the 75% of the solutions obtained by NSGA-II). Figure 4 shows the attainment surfaces of the EMO algorithms and we see that MOEA/D has the highest number of solutions of the pseudo-optimal Pareto front. Only three NSGA-II solutions belong to the pseudo-optimal Pareto front but these solutions do not dominate any of the solutions of the MOEA/D Pareto front. The single-objective GA shows low results and its far from MOEA/D and NSGA-II performances.

7. Conclusions

This study proposed a novel multiobjective optimization problem formulation to maximize the revenue while minimizing the costs of a VM campaign that targets influentials with free product samples. We applied two EMO algorithms, NSGA-II and MOEA/D, a greedy algorithm and a single-objective GA which evaluate the fitness of the VM campaigns using an artificial market framework based on ABM. The ABM simulation closely represents consumer agents' reality when choosing among existing products and spreading consumers' peer influence in the social network. The two evolutionary algorithms obtained sets of non-dominated solutions with different trade-offs between high revenue and low costs.

We found that EMO algorithms showed different performance depending on the characteristics of the social network. Our experimentation comprised social networks from high information diffusion speed (e.g., SF networks) to low information diffusion speed (e.g., RAND and REG networks) and applied statistical tests to confirm the results. We concluded that the properties of the social networks affected the best performing EMO algorithm as well as the type of non-dominated solutions obtained by the algorithms to optimize the VM campaign. For instance, MOEA/D obtained more extended fronts and had the best performance for the artificial SF and EMAIL networks. In contrast, NSGA-II outperformed MOEA/D for social networks having slow diffusion speed such as the REG network. Experiments also showed that there are differences among the features of the Pareto front approximations. For instance, Pareto fronts found by NSGA-II had, on average, a higher number of solutions than those obtained by MOEA/D. The single-objective GA had low values for unary and binary multiobjective indicators and returned sets with a low number of non-dominated solutions which did not normally belong to the pseudo-optimal

Pareto front. A traditional greedy algorithm obtained the worst results for all the multiobjective indicators.

The validation of our EMO approach was enhanced by a real case study with a social network of 20,000 nodes and a bi-modal degree distribution. The algorithms behave similarly, having MOEA/D the best performance. Therefore, using an EMO algorithm is an appropriate way to solve the targeting VM problem, although modelers need to evaluate the social network topology to choose the best EMO method. During our experimentation we also showed that the set of non-dominated solutions can help marketers' decisions on the most appropriate targeting campaigns to be applied.

8. Acknowledgments

This work is supported by the Spanish Ministry of Science, Innovation and Universities, and ERDF under grant EXASOCO (PGC2018-101216-B-I00). M. Chica is supported by the Ramón y Cajal program (RYC-2016-19800).

References

Banerjee, S., Jenamani, M., Pratihar, D. K., 2019. Combim: A community-based solution approach for the budgeted influence maximization problem. Expert Systems with Applications 125, 1–13.

Barabási, A. L., Albert, R., 1999. Emergence of scaling in random networks. Science 286 (5439), 509–512.

Chica, M., Chiong, R., Kirley, M., Ishibuchi, H., 2018. A networked N-player trust game and its evolutionary dynamics. IEEE Transactions on Evolutionary Computation 22 (6), 866–878.

Chica, M., Rand, W., October 2017. Building agent-based decision support systems for word-of-mouth programs. a freemium application. Journal of Marketing Research 54, 752–767.

Coello, C. A., Lamont, G. B., Van Veldhuizen, D. A., 2007. Evolutionary Algorithms for Solving Multi-objective Problems (2nd edition). Springer.

Deb, K., 2001. Multi-objective Optimization Using Evolutionary Algorithms. Wiley.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6 (2), 182–197.

Domingos, P., Richardson, M., 2001. Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 57–66.

Ehrenberg, A. S., Uncles, M. D., Goodhardt, G. J., 2004. Understanding brand performance measures: using dirichlet benchmarks. Journal of Business Research 57 (12), 1307–1325.

Epstein, J. M., 2006. Generative social science: Studies in agent-based computational modeling. Princeton University Press.

Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., Arenas, A., 2003. Self-similar community structure in a network of human interactions. Physical Review E 68 (6), 065103.

Gunasekara, R. C., Mehrotra, K., Mohan, C. K., 2015. Multi-objective optimization to identify key players in large social networks. Social Network Analysis and Mining 5 (1), 1–20.

Haenlein, M., Libai, B., 2017. Seeding, referral, and recommendation: Creating profitable word-of-mouth programs. California Management Review 59 (2), 68–91.

Janssen, M. A., Jager, W., 2003. Simulating market dynamics: Interactions between consumer psychology and social networks. Artificial Life 9 (4), 343–356.

Keikha, M. M., Rahgozar, M., Asadpour, M., Abdollahi, M. F., 2019. Influence maximization across heterogeneous interconnected networks based on deep learning. Expert Systems with Applications, 112905.

Kempe, D., Kleinberg, J., Tardos, É., 2003. Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 137–146.

Lahiri, M., Cebrian, M., 2010. The genetic algorithm as a general diffusion model for social networks. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. pp. 494–499.

Leskovec, J., Adamic, L. A., Huberman, B. A., 2007. The dynamics of viral marketing. ACM Transactions on the Web (TWEB) 1 (1), 5.

Newman, M., Barabási, A.-L., Watts, D. J., 2006. The structure and dynamics of networks. Princeton University Press.

Robles, J. F., Chica, M., Cordón, Ó., July 2016. Incorporating awareness and genetic-based viral marketing strategies to a consumer behavior model. In: Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC). pp. 5178–5185.

Schlereth, C., Barrot, C., Skiera, B., Takac, C., 2013. Optimal product-sampling strategies in social networks: How many and whom to target? International Journal of Electronic Commerce 18 (1), 45–72.

ŞİMŞEK, A., Resul, K., 2018. Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks. Expert Systems with Applications 114, 224–236.

Stonedahl, F., Rand, W., Wilensky, U., 2010. Evolving viral marketing strategies. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation. GECCO '10. ACM, New York, NY, USA, pp. 1195–1202.

Watts, D. J., Dodds, P. S., 2007. Influentials, networks, and public opinion formation. Journal of Consumer Research 34 (4), 441–458.

Watts, D. J., Strogatz, S. H., 1998. Collective dynamics of 'small-world' networks. Nature 393 (6684), 440–442.

Zhang, Q., Li, H., 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11 (6), 712–731.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., da Fonseca, V. G., 2003. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 7 (2), 117–132.

9. Tables

Table 1: Main features of the used social network topologies

	I	Edges	Avg. path length	Avg. degree	Clust. coeff.	Density
EMAIL	1133	5451	3.60	9.622	0.254	0.009
SF	1000	5007	3.04	10.014	0.056	0.01
RAND	1000	4980	3.27	9.852	0.008	0.01
SW	1000	5000	4.43	10	0.495	0.01
REG	1024	2048	16.02	4	0	0.04
APP	20000	481907	3.02	48.19	0.006	0.002

Table 2: Average and Standard Deviation Values of HVR and cardinality (best results in bold)

	HVR				Cardinality			
	MOEA/D	NSGA-II	GA	Greedy	MOEA/D	NSGA-II	GA	Greedy
EMAIL	**0.96**	0.947	0.913	0.864	41.33	**65.47**	10	32
	(0.017)	(0.007)	(0.01)	(0.008)	(3.32)	**(4.88)**	(0.83)	(3.1)
SF	**0.957**	0.938	0.928	0.862	35	**59.86**	9.03	32
	(0.017)	(0.007)	(0.01)	(0.003)	(3.05)	**(6.21)**	(1.03)	(2.54)
RAND	**0.943**	0.94	0.937	0.857	41	**65.5**	9.56	29
	0.023	(0.009)	0.01	(0.02)	(3.85)	**(6.25)**	(0.86)	(5.8)
SW	0.932	**0.934**	0.878	0.789	39	**63.6**	8.73	21
	(0.028)	**(0.01)**	(0.01)	(0.023)	(3.76)	**(5.35)**	(1.01)	(2.7)
REG	0.865	**0.897**	0.687	0.653	28	**35**	6.1	15
	(0.045)	**(0.022)**	(0.037)	(0.034)	(3.68)	**(3.74)**	(0.8)	(4.21)

Table 3: Average and Standard Deviation Values of Coverage C (best results in bold)

	C(MOEA/D,NSGA-II) / C(NSGA-II,MOEA/D)	C(MOEA/D,GA) / C(GA,MOEA/D)	C(MOEA/D,Greedy) / C(Greedy,MOEA/D)	C(NSGA-II,GA) / C(GA,NSGA-II)	C(NSGA-II,Greedy) / C(Greedy,NSGA-II)	C(GA,Greedy) / C(Greedy, GA)
EMAIL	**0.461 (0.019)** / 0.358 (0.0246)	**0.604 (0.0152)** / 0.068 (0.05)	**1.0 (0.0)** / 0.0 (0.0)	**0.526 (0.022)** / 0.082 (0.008)	**1.0 (0.0)** / 0.0 (0.0)	**1.0 (0.0)** / 0.0 (0.0)
SF	**0.4803 (0.0218)** / 0.333 (0.0212)	**0.439 (0.0252)** / 0.09 (0.075)	**1.0 (0.0)** / 0.0 (0.0)	**0.421 (0.0142)** / 0.145 (0.0076)	**1.0 (0.0)** / 0.0 (0.0)	**1.0 (0.0)** / 0.0 (0.0)
RAND	**0.4529 (0.026)** / 0.4072 (0.0277)	**0.448 (0.0121)** / 0.139 (0.065)	**1.0 (0.0)** / 0.0 (0.0)	**0.385 (0.0201)** / 0.14 (0.093)	**1.0 (0.0)** / 0.0 (0.0)	**1.0 (0.0)** / 0.0 (0.0)
SW	0.3261 (0.0238) / **0.5537 (0.0263)**	**0.3 (0.0151)** / 0.249 (0.0136)	**1.0 (0.0)** / 0.0 (0.0)	**0.414 (0.0116)** / 0.175 (0.059)	**1.0 (0.0)** / 0.0 (0.0)	**1.0 (0.0)** / 0.0 (0.0)
REG	0.2289 (0.0164) / **0.715 (0.0178)**	**0.647 (0.0235)** / 0.113 (0.0132)	**1.0 (0.0)** / 0.0 (0.0)	**0.733 (0.0164)** / 0.045 (0.013)	**1.0 (0.0)** / 0.0 (0.0)	**1.0 (0.0)** / 0.0 (0.0)

Table 4: Average and Standard Deviation Values of multiplicative I_ϵ (best results in bold)

	I_ϵ(MOEA/D,NSGA-II) / I_ϵ(NSGA-II,MOEA/D)	I_ϵ (MOEA/D,GA) / I_ϵ(GA,MOEA/D)	I_ϵ(MOEA/D,Greedy) / I_ϵ(Greedy,MOEA/D)	I_ϵ(NSGA-II,GA) / I_ϵ(GA,NSGA-II)	I_ϵ(NSGA-II,Greedy) / I_ϵ(Greedy,NSGA-II)	I_ϵ(GA,Greedy) / I_ϵ(Greedy, GA)
EMAIL	**1.023 (0.009)** / 1.04 (0.018)	**1.014 (0.007)** / 1.092 (0.023)	**0.928 (0.001)** / 1.22 (0.048)	**1.016 (0.009)** / 1.093 (0.023)	**0.953 (0.002)** / 1.21 (0.02)	**0.988 (0.002)** / 1.12 (0.037)
SF	**1.022 (0.006)** / 1.042 (0.017)	**1.019 (0.009)** / 1.086 (0.023)	**0.91 (0.001)** / 1.33 (0.028)	**1.025 (0.009)** / 1.09 (0.022)	**0.952 (0.025)** / 1.171 (0.035)	**0.973 (0.032)** / 1.22 (0.04)
RAND	**1.03 (0.018)** / 1.034 (0.018)	**1.024 (0.013)** / 1.108 (0.031)	**0.921 (0.004)** / 1.22 (0.03)	**1.023 (0.008)** / 1.103 (0.024)	**0.946 (0.002)** / 1.19 (0.018)	**0.983 (0.005)** / 1.12 (0.02)
SW	1.04 (0.019) / **1.031 (0.02)**	**1.028 (0.014)** / 1.094 (0.031)	**0.91 (0.001)** / 1.325 (0.032)	**1.022 (0.008)** / 1.09 (0.022)	**0.938 (0.003)** / 1.268 (0.003)	**0.964 (0.012)** / 1.25 (0.012)
REG	1.083 (0.026) / **1.03 (0.027)**	**1.023 (0.024)** / 1.205 (0.059)	**0.94 (0.002)** / 1.31 (0.03)	**1.01 (0.011)** / 1.205 (0.045)	**0.928 (0.001)** / 1.703 (0.002)	**0.993 (0.002)** / 1.19 (0.023)

Table 5: Decision space values obtained by solutions of MOEA/D, NSGA-II and GA in EMAIL and REG social networks

	MOEA/D				NSGA-II				GA				Greedy			
	W_d	W_{2s}	W_{cc}	F_S	W_d	W_{2s}	W_{cc}	F_S	W_d	W_{2s}	W_{cc}	F_S	W_d	W_{2s}	W_{cc}	F_S
EMAIL	Highest revenue solution ($f^0(x)$=1398.17, $f^1(x)$=191.25)				Highest revenue solution ($f^0(x)$=1350.2, $f^1(x)$=174.38)				Highest revenue solution ($f^0(x)$=1354.2, $f^1(x)$=182.25)				Highest revenue solution ($f^0(x)$=1180.0, $f^1(x)$=201.3)			
	0.58	0.9	0.23	170	0.48	0.98	0.29	155	0.34	0.78	0.22	162	1.0	0.0	0.0	198
	Best trade-off solution ($f^0(x)$=972.12, $f^1(x)$=76.3)				Best trade-off solution ($f^0(x)$=958.4, $f^1(x)$=79.88)				Best trade-off solution ($f^0(x)$=950.8, $f^1(x)$=84.6)				Best trade-off solution ($f^0(x)$=912.7, $f^1(x)$=92.2)			
	0.7	0.84	0.27	69	0.64	0.96	0.32	71	0.58	0.75	0.27	75	1.0	0.0	0.0	88
	Lowest cost solution ($f^0(x)$=204.92, $f^1(x)$=15.75)				Lowest cost solution ($f^0(x)$=238.8, $f^1(x)$=22.5)				Lowest cost solution ($f^0(x)$=70.03, $f^1(x)$=14.625)				Lowest cost solution ($f^0(x)$=52.12, $f^1(x)$=26.8)			
	0.91	0.87	0.4	14	0.87	0.85	0.49	20	0.86	0.72	0.37	13	1.0	0.0	0.0	24
REG	Highest revenue solution ($f^0(x)$=155.4, $f^1(x)$=398.25)				Highest revenue solution ($f^0(x)$=148.1, $f^1(x)$=282.38)				Highest revenue solution ($f^0(x)$=123,43, $f^1(x)$=148.5)				Highest revenue solution ($f^0(x)$=120.5, $f^1(x)$=152.3)			
	0.31	0.52	0.47	354	0.72	0.85	0.68	251	0.5	0.64	0.53	132	0.33	0.33	0.33	138
	Best trade-off solution ($f^0(x)$=112.4, $f^1(x)$=132.4)				Best trade-off solution ($f^0(x)$=122.38, $f^1(x)$=125.03)				Best trade-off solution ($f^0(x)$=108.6, $f^1(x)$=111.58)				Best trade-off solution ($f^0(x)$=89.31, $f^1(x)$=120.45)			
	0.34	0.56	0.32	117	0.47	0.63	0.0.25	111	0.74	0.58	0.49	99	0.33	0.33	0.33	114
	Lowest cost solution ($f^0(x)$=27.13, $f^1(x)$=1.13)				Lowest cost solution ($f^0(x)$=22.24, $f^1(x)$=1.13)				Lowest cost solution ($f^0(x)$=15.41, $f^1(x)$=1.13)				Lowest cost solution ($f^0(x)$=10.58, $f^1(x)$=1.13)			
	0.27	0.62	0.27	1	0.41	0.26	0.23	1	0.82	0.47	0.53	1	0.33	0.33	0.33	1

Table 6: Average and Standard Deviation Values of multiobjective performance indicators for the real case study

		MOEA/D	NSGA-II	GA
HVR		**0.999 (0.0008)**	0.997 (0.0002)	0.94 (0.0031)
Cardinality		22.6 (3.51)	**34.2 (5.18)**	12.5 (1.2)
I_ϵ	MOEA/D		**1.002 (0.002)**	**1.005 (0.001)**
	NSGA-II	1.008 (0.001)		**1.003 (0.002)**
	GA	43.08 (0)	42.7 (0.02)	
C	MOEA/D		**0.742 (0.121)**	**1.0 (0)**
	NSGA-II	0 (0.001)		**1.0 (0.001)**
	GA	0 (0)	0.05 (0.001)	

10. Figures

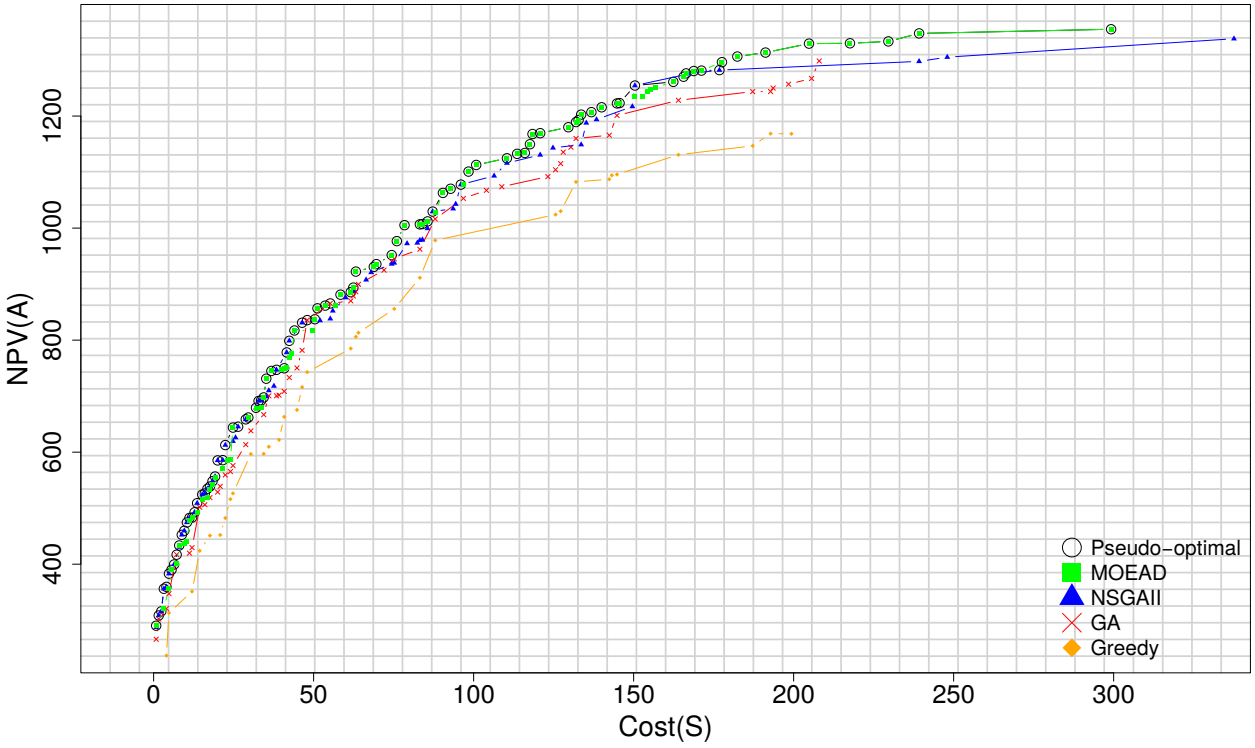

Figure 1: Pareto front approximations obtained for the EMAIL network.

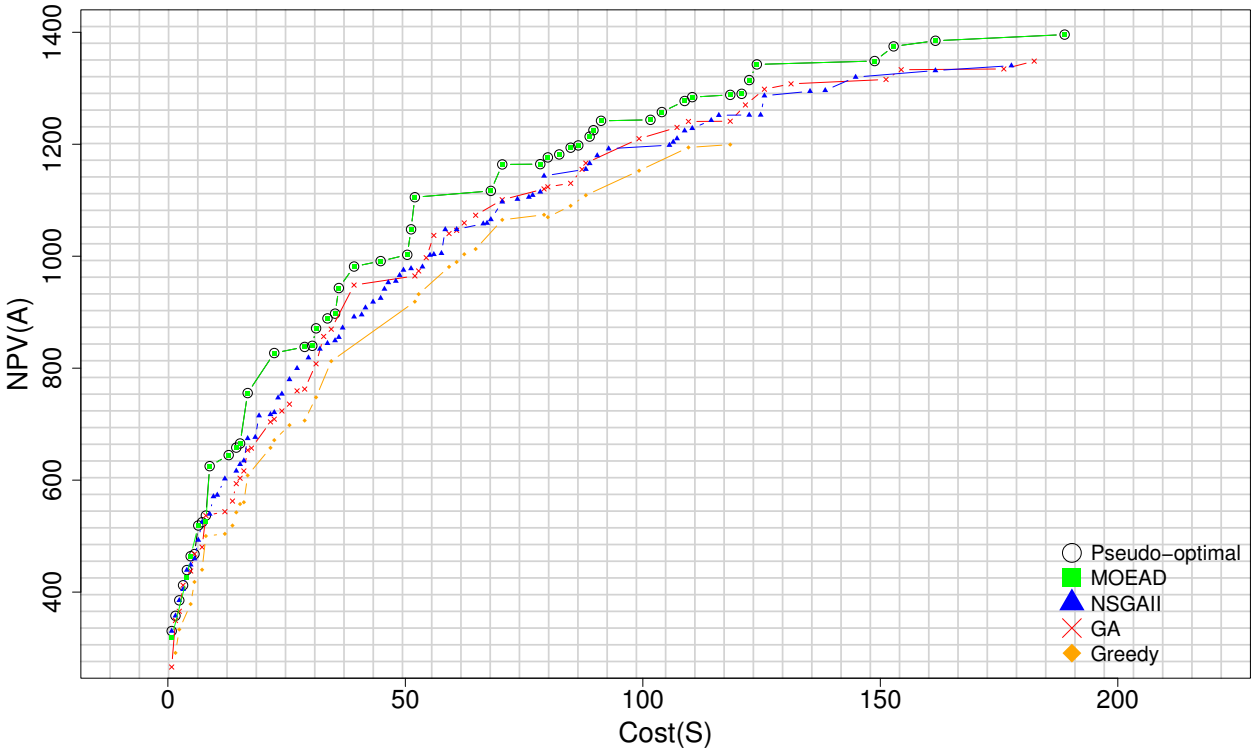

Figure 2: Pareto front approximations obtained for the SF network.

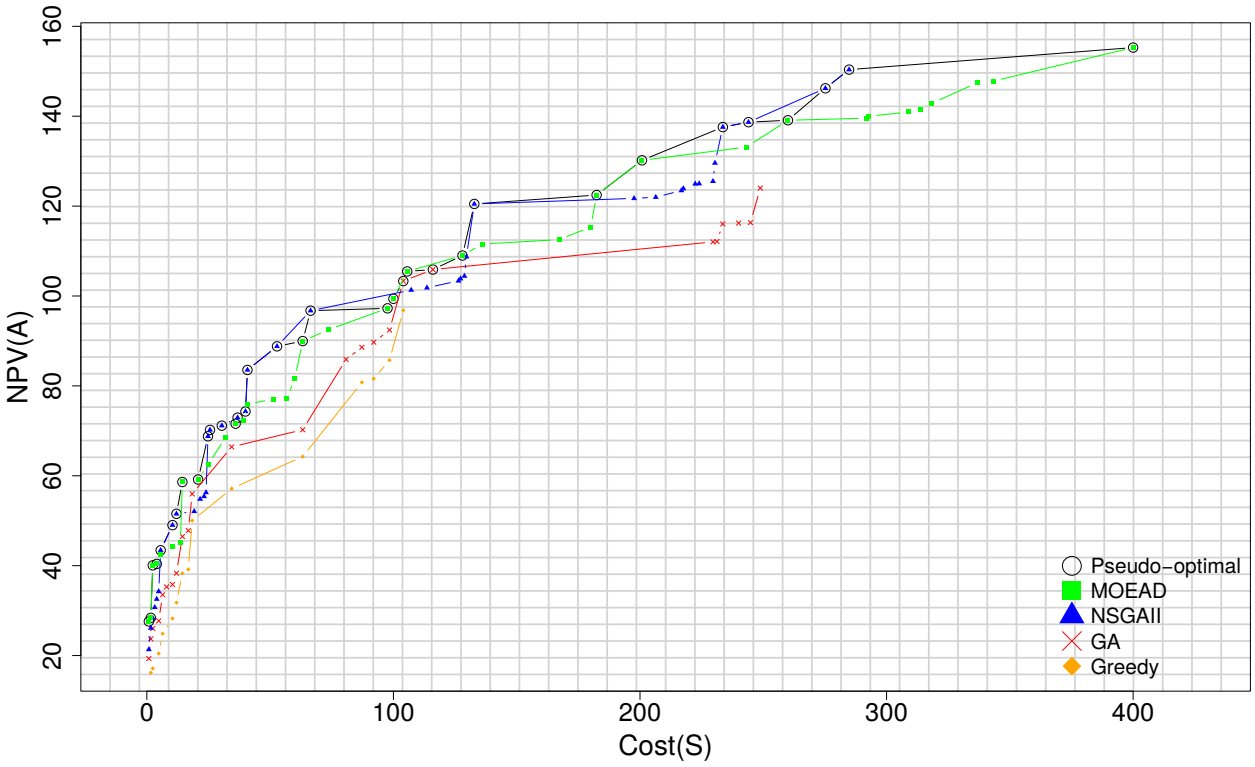

Figure 3: Pareto front approximations obtained for the REG network.

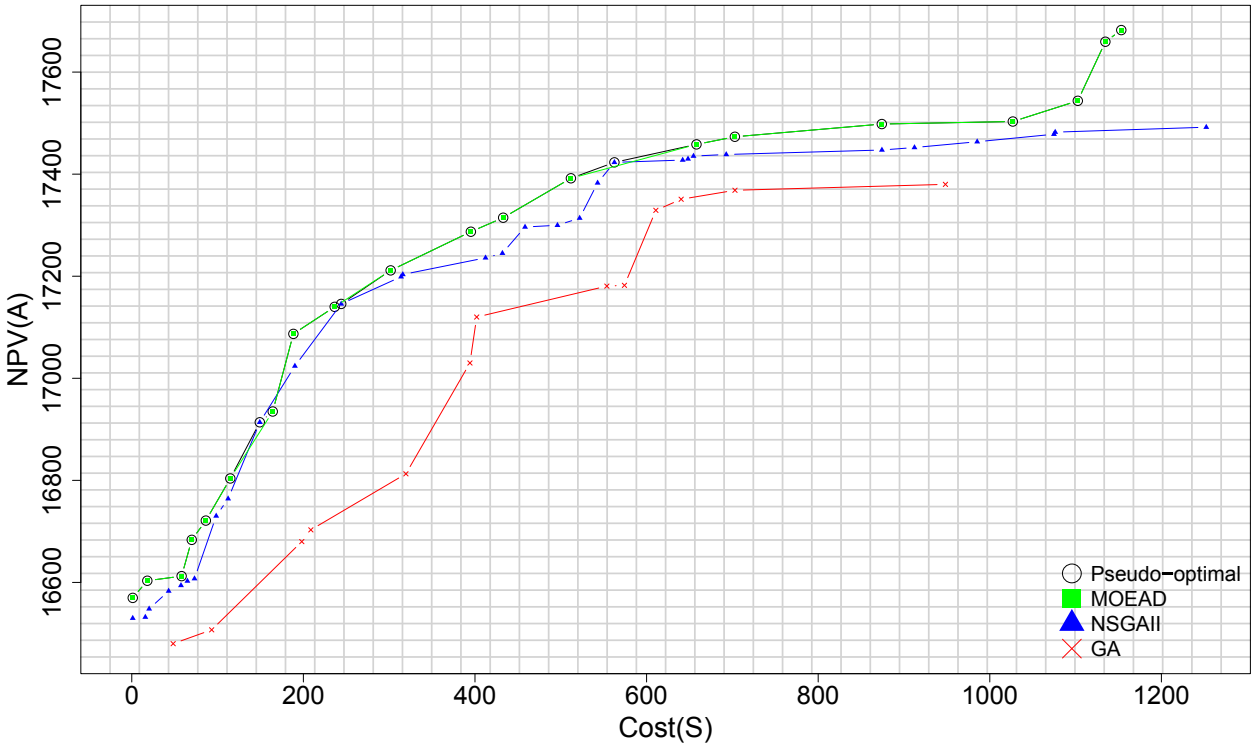

Figure 4: Pareto front approximations obtained for the APP network.