

# Choice of optimization routine for multi-agent models: A case of viral video marketing campaign

Michal Kvasnička<sup>1</sup>

**Abstract.** Very few agent-based computational models are optimized because the usually used optimization routine, the genetic algorithm, is extremely time-consuming. This paper explores how much precision is lost if a simpler optimization routine, mutational hill climber, is used instead. It shows on the case of a viral-video marketing model that even though the standard genetic algorithm is slightly more precise, the mutation hill climbing could be used as an approximate optimization routine for robustness check and scenario analysis.

**Keywords:** optimization, genetic algorithm, mutation hill climbing, simulation, agent-based model, social network, viral video marketing.

**JEL classification:** C61, C63, D85, M31

**AMS classification:** 68U20, 37N40, 05C82, 90B15

## 1 Introduction

Agent-based computational models (ABM) allow us to study many real-world phenomena which are difficult to explore with the standard tools of economic theory. The models can also be used to optimize the explored processes. However, since the structure of the ABM models does not allow us to use the standard optimization routines and available techniques (e.g. genetic algorithms) are computationally extremely demanding, most researchers do not try to optimize their models, and even those few exceptions that do usually stick with one or few particular settings of their model's exogenous parameters, and do neither provide robustness checks, nor explore systematically how the optimal parameters depend on the exogenous parameter of the model.

For example, consider the ABM literature on viral marketing. (The ABM is a natural tool to study this phenomenon because the explicit description of the social network is necessary, the agents interact locally within the network, and their behavior is non-continuous.) Most of this wide literature (for a review see e.g. [1]) is only exploratory: it studies how knowledge about a new product diffuses over a social network. The understanding of the diffusion process is interesting as such, but it also creates an opportunity for marketers to actively utilize the spontaneous knowledge diffusion as an advertising tool. To do it, the marketer must decide how many and which consumers “infect” with the knowledge of the product and how to do it. This, however, requires to optimize the model. Presently, there are only few papers trying to optimize the ABM viral marketing models because the optimization is extremely time consuming even in a case of a small stylized social network. If we neglect the papers with unrealistic assumptions that marketers know the complete description of the social network and are able to optimize over it, there are only two papers trying to optimize a viral marketing campaign, [6] and [2].

Both these papers assume that the marketer has only local knowledge about the social network, such as the number of agents in the network, the agents' degree (the number of their connections), their clustering, their willingness to share the knowledge etc. The marketer's problem is then to seed the network, i.e. to choose the agents she initially “infects” with the knowledge of a product, to maximize her objective function. Both these papers also describe a seeding strategy as a vector of the number of initially seeded agents and weights placed on their desirable properties. Both the papers use genetic algorithms implemented in BehaviorSearch [4, 5] to optimize the seeding strategy too. The papers differ in their underlying assumptions about the diffusion process ([6] assumes the traditional “word-of-mouth” marketing, while [2] assumes viral-video marketing) and the marketer's objective function ([6] assumes that the marketer maximizes the discounted number of “infected” agents, while [2] assumes she maximizes the profit from the marketing campaign). The authors of [6] optimized two scenarios on five stylized networks, which took about 462 CPU-days. [2] optimized only one scenario, which took about 480 CPU-days.

---

<sup>1</sup>Masaryk University, Faculty of Economics, Department of Economics, Lipová 41a, Brno, 602 00, michal.kvasnicka@econ.muni.cz

The above mentioned case shows that using genetic algorithms (which are supposed to be the most sophisticated optimization tool available, see [4, 5]) to optimize an ABM model practically precludes a more systematic exploration of the optimal setting of the model, at least at the present state of the computing power. To do so, it is necessary to use faster (yet perhaps less sophisticated) optimization routines. The mutation hill climbing seems to be a promising substitute. However, it is not known how much precision is lost when the genetic algorithms are substituted with the mutation hill climbing. The goal of this paper is explore this question. Since the ABM does not allow us to study the problem in general, it is assessed on the particular case of the ABM model of viral-video marketing described in [2]. Our procedure is the following: 1) we implement the model, 2) we find its optimal parameters both using genetic algorithms and mutation hill climbing (the first routine is given reasonably more resources than the later one), and 3) we compare the results.

## 2 Model

The model used to test the relative performance of the standard genetic algorithm (GA) and the mutation hill climbing (MHC) is a slightly modified version of [2]. It differs from the other ABM models of viral marketing in its agents' activation: it assumes that people share videos over Internet and that their decision to share the video is independent from their decision to adopt the product advertised by the video (if any). It differs from the model described in [2] in three respects: 1) there are fewer agents (250 instead of 1 000), 2) the distribution of followers' links is independent on the distribution of the friends' links, and 3) the agents' properties seen by the marketer were slightly modified too. The model consists of three parts: 1) the activation mechanism (i.e. how the agent share the video and get "infected"), 2) an explicit description of the used social network, and 3) the seeding strategy (i.e. which and how many agents get initially infected by the marketer).

### 2.1 Activation mechanism

I call every agent that has viewed the video *infected* with no regard whether she has adapted the advertised product or not. An infected agent's decision whether and with whom to share a video and the receiver's decision to view the video are modeled as a simple probabilistic act: an infected person shares the video with each of her neighbors with some probability. If she shares it with a person, the person gets infected. Each infected person shares the video only once.

The precise mechanism of the activation is following: At the initialization, every agent  $i$  draws a probability  $p_i$  that she shares the video;  $p_i$  is drawn from the continuous uniform distribution  $U(0, v)$  where  $v$  is the maximal *virality* of the video. Then she creates a list  $l_i$  of agents to share the video with: she adds each of her neighbors to the list  $l_i$  with probability  $p_i$ . When agent  $i$  is first infected at time  $t$ , she shares the video with all agents in her list  $l_i$  at time  $t + 1$ , and each of these agents get instantly infected. Agent  $i$  shares the video with no one after time  $t + 1$ .

### 2.2 Network structure

An agent's neighborhood is defined by a social network in which the agent is located. There are two kinds of relations between the agents in the model: friendship and following. Friendship is a symmetric relationship between two agents that can share videos with each other. Following is an asymmetric relationship between a followed person and her follower, e.g. between a celebrity and her fan. The followed person can share videos with the follower but not vice versa. I assume that no person can be at the same time one's friend and follower.

Each model network consists of 250 agents and its parameters are selected to resemble the properties of the empirical networks. Each network is created in two steps. First, the symmetric small world network which represents the friendship is created by algorithm described by [8]; the code has been adapted from [10]. The agents are arranged into a circle. Each agent initially has 10 friends, 5 agents to the left of her and 5 agents to the right of her. Each link representing friendship is then rewired with probability 10 %, which creates the initial small world network. Second, the asymmetric power network of directed links which represents the following is created over the friends' network. The algorithm has been adapted from [11]: one follower connection is added at a time, each agent is selected randomly as the follower with an equal probability and one other agent is selected randomly as the followed one with the probability proportional to each agent's number of the previously created followers. Two followers' links per agent are added. The set of agent  $i$ 's neighbors used in the activation mechanism described above is the union of the set of her friends and the set of her followers.

### 2.3 Marketer's seeding strategy and profit

The marketer's seeding strategy is based on the approach developed by [6]. The agents are included into the seed because they have some desirable properties. Since there are multiple desirable properties, they are weighted. A seeding strategy  $(S, w)$  thus consists of two parts: the seed size  $S$ , i.e. how many agents the marketer initially infects, and weights  $w_j$  placed on measures of some desirable properties  $f_j$  that determine which agents are selected into the seed. An index  $\sum w_j f_j$  is calculated for each agent and  $S$  agents with the highest value of the index are included into the seed. I use six measures  $f_j$ : 1)  $f_1$  = the number of agent's friends divided by the maximal number of friends in the population, 2)  $f_2$  = the number of agent's followers divided by the maximal number of followers in the population, 3)  $f_3$  = the number of agent's friends and followers divided by the maximal number of friends and followers in the population, 4)  $f_4$  = the agent's sharing probability  $p_i$  divided by the maximal sharing probability in the population, 5)  $f_5 = 1 - (\text{agent's clustering ratio} / \text{the maximal clustering ratio in the population})$ , and 6)  $f_6$  is a random number drawn from  $U(0, 1)$ . The first three  $f$ s are various measures of degree—the more connections an agent has, the better she can share the video. (Only  $f_1$  and  $f_2$  are used here, i.e.  $w_3 = 0$ . Property  $f_3$  is used only for comparison, see below.) The  $f_4$  measures how likely the agent is to share the video and how likely are her recipients to watch it—the higher, the better she can share the video. The  $f_5$  indicates the agent's absence of clustering—the less likely an agent's neighbors are to be neighbors themselves, the better the video can spread through the population. The  $f_6$  allows including agents into the seed randomly, which may be beneficial e.g. when the agents with the highest degree are connected together.

Since an agent's decisions to share a viral video and to adopt the product advertised by the video are independent, the expected revenue from the viral marketing campaign is then equal to  $\rho\sigma N$ , where  $N$  is the number of agents infected during the campaign,  $\rho$  is the probability that an infected agent adopts the product because of the campaign, and  $\sigma$  is the profit from one adopter. The cost of the campaign is  $\gamma S + F$  where  $\gamma$  is a cost of seeding one agent,  $S$  is the seed size, and  $F$  is a fixed cost. The marketer's problem is then to select the seeding strategy  $(S^*, w^*)$  that maximizes her expected profit from the campaign,  $\Pi = \rho\sigma N - \gamma S - F$ . The seeding strategy  $(S^*, w^*)$  maximizing  $\Pi$  also maximizes  $\pi = N - cS$  where  $c = \gamma/(\rho\sigma)$  is the relative cost of seeding one agent. (Since I do not explicitly address the problem of setting the optimal budget for the video creation, I treat  $F$ ,  $\rho$ , and  $v$  as constants.)

### 2.4 Simulation and implementation

Each simulation consists of two parts: the initialization and the run. In the initialization, the agents are created and connected within the model network. Each agent  $i$  is assigned the probability  $p_i$  that she shares the video with her friends and followers. She then creates the list  $l_i$  of the agents which she shares the video with if infected. At the end of the initialization, the initial agents are infected. Each simulation run proceeds in discrete steps. In each step, each agent  $i$  infected in the previous step infects the agents in  $l_i$ . The run ends when there are no infected agents that have not yet shared the video. Each model is simulated for 250 agents, the activation and network described in sections 2.1 and 2.2, maximal virality  $v = 20\%$ , and relative seeding cost  $c = 10$ . The model was implemented in NetLogo 5.1 [9].

### 2.5 Optimization

To assess the relative performance of the standard genetic algorithm (GA) and the mutation hill climbing (MHC) (as implemented in BehaviorSearch 1.0 [4]), the optimal seeding strategy  $(S^*, w^*)$  was searched by both these methods. The seed size  $S$  was searched on domain of  $1, 2, \dots, 20$  and each weight  $w_j$  on domain  $[0, 1]$  (the model automatically normalizes  $\sum w$  to unity). All variables were encoded as Gray binary chromosomes. The fitness of each individual strategy was evaluated by the mean relative profit  $\pi$  of 50 independent replications of the simulation. The whole search for the optimal parameters was repeated 50 times for both the GA and MHC optimizer (i.e. 50 optimal seeding strategy candidates were produced by each optimizer).

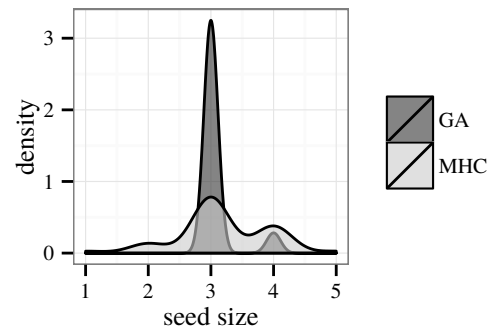
The genetic algorithm started with the initial population of 50 randomly chosen strategies. The standard generational evolution steps (one-point crossover rate 0.7, mutation rate 0.03, and tournament selection with tournament size 3) were performed on the population of the 50 strategies for 80 generations. Total number of model runs was 200 000 (excluding checking replicates) per one search. The time needed for the GA optimization was about 61 CPU-days. The mutation hill climber started with one random strategy. The mutation rate was set to 0.03, the stalled searches were restarted after 300 attempts. The total number of model runs was 30 000 (excluding checking replicates). The mean time needed for the MHC optimization was about 11 CPU days (the mean is computed from 72 similar optimization runs carried for a related project).

### 3 Results

The performance of the GA and MHC optimizer is compared in two ways: 1) the “optimal” seeding strategy parameters are compared and 2) the profits realized from the two strategies are compared with each other and with the profits from the simple seeding strategies. The performance analysis was carried in R 3.2.0 [3].

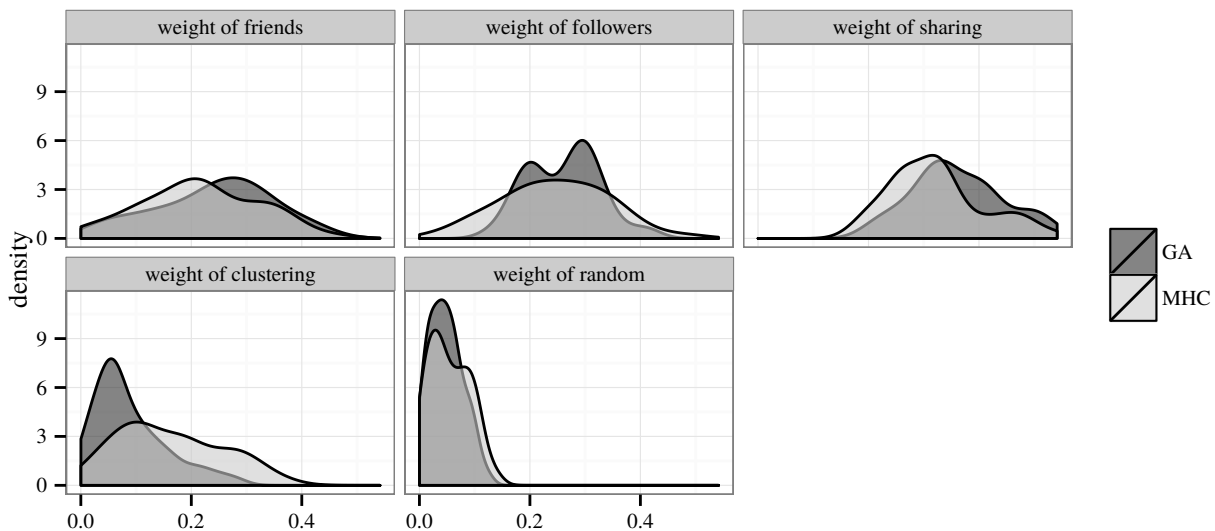
Table 1 presents the parameter of the optimal seeding strategies found by the standard genetic algorithm and the mutation hill climber respectively. The statistics are computed from 50 candidates for each optimizer. The full distributions of the parameters are presented in Figures 1 and 2. The parameters are very similar, with the exception of the sharing weight  $w_4$  and clustering weight  $w_5$  (the mean values of these two parameters differ significantly between the respective optimizers at the significance level 5 %). The standard genetic algorithm is slightly more efficient—the standard deviations of the parameters found by GA are in general lower, i.e. individual runs of the optimizer produce candidates that are closer to each other than the candidates produced by MHC. However, the difference between the variances of the candidates’ parameters is statistically significant at 5 % only in the case of the seed size and the weight of followers  $w_2$ .

|                  | mean  |       | sd    |       |
|------------------|-------|-------|-------|-------|
|                  | MHC   | GA    | MHC   | GA    |
| seed size        | 3.180 | 3.080 | 0.720 | 0.274 |
| $w_1$ friends    | 0.216 | 0.241 | 0.106 | 0.106 |
| $w_2$ followers  | 0.245 | 0.261 | 0.098 | 0.065 |
| $w_4$ sharing    | 0.323 | 0.362 | 0.086 | 0.083 |
| $w_5$ clustering | 0.161 | 0.088 | 0.094 | 0.066 |
| $w_6$ random     | 0.055 | 0.048 | 0.037 | 0.030 |



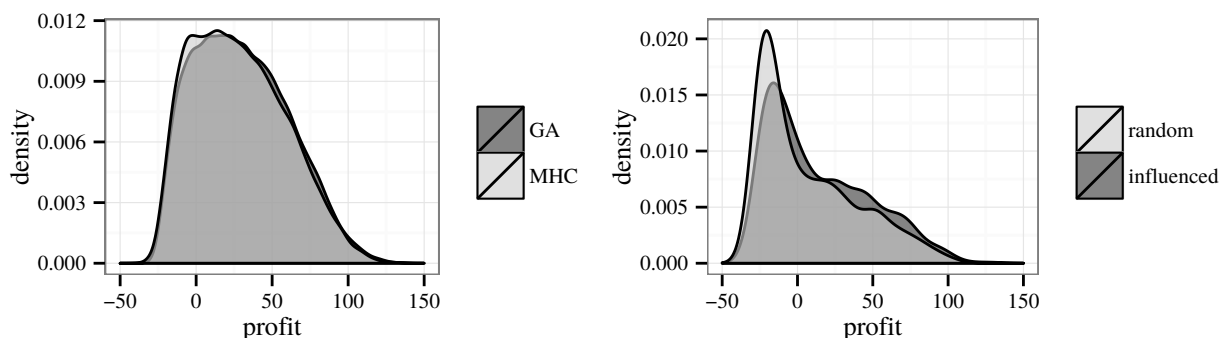
**Table 1:** The optimal seeding strategy candidates’ parameters found by the standard genetic algorithm (GA) and the mutation hill climber (MHC). The statistics are computed from 50 candidates for each optimizer.

**Figure 1:** Distribution of the optimal seed size found by the standard genetic algorithm (GA) and the mutation hill climber (MHC). The kernel densities are computed from 50 candidates for each optimizer.



**Figure 2:** Distribution of the optimal seeding strategy candidates’ weights found by the standard genetic algorithm (GA) and the mutation hill climber (MHC). The kernel densities are computed from 50 candidates for each optimizer.

The previous results in optimization of the agent-based viral marketing models ([6, 2]) show there might be a plateau in the objective function (here profit) around the optimal strategy, i.e. that there is some range of parameters that produce very similar outcomes as the optimal seeding strategy. Furthermore, the individual parameters could



**Figure 3:** Distribution of profits of the optimal strategy candidates. The kernel densities are computed from 1 000 replication for each candidates; there are 50 candidates for GA and MHC and 1 candidate for each simple strategy.

partially substitute for each other. Thus the difference between the optimal strategy candidates found by the GA and MHC is not a sufficient proof that one routine is inferior to the other, especially since we do not know the truly optimal parameters. Thus it is necessary to compare the profits yielded by the strategies.

For this reason, I simulated the model 1 000 times for each of the 50 optimal strategy candidates and 1 000 times for other four strategies. First, for the mean GA and MHC optimal strategies; their parameters were computed as a mean of the respective parameter of the 50 candidates found by the respective optimizer and are the same as values presented in the first two columns of Table 1 with the exception of the seed size which was rounded to the nearest integer. Second, for two simple strategies created for a comparison: strategy  $(S^R, 0, 0, 0, 0, 0, 1)$  that selects the agents into the seed randomly (i.e.  $w_6 = 1$ ) and strategy  $(S^F, 0, 0, 1, 0, 0, 0)$  that selects into the seed the agents that can influence most other agents, i.e. the agents with the maximal number of friends and followers together (i.e.  $w_3 = 1$ ). These two simple strategies are natural candidates for a marketer that does not want to optimize—she would either seed the agents randomly or seed the agents with most connections. The seed sizes  $S^F = S^R = 3$  were selected because they maximized the expected profit.

The results of these simulations are presented in Table 2 and Figure 3. The average profit of the GA candidates is slightly higher than the average profit of the MHC candidates though their distributions are very similar. The profits of the mean GA and mean HMC optimal strategies are virtually the same. The comparison with the profits of the simple strategies (random and influenced) show that this is not due the plateau in the profit function—both these simple strategies perform much worse than both the GA and MHC candidates.

| seed strategy | mean profit | sd profit |
|---------------|-------------|-----------|
| GA            | 30.18       | 30.87     |
| MHC           | 28.24       | 30.69     |
| mean GA       | 30.27       | 31.64     |
| mean MHC      | 30.90       | 31.08     |
| random        | 6.86        | 32.65     |
| influenced    | 14.21       | 33.78     |

**Table 2:** Profits of the strategy candidates. The statistics for the standard genetic algorithm (GA) and the mutation hill climber (MHC) were computed from 50 000 independent replications (1 000 replications for each of 50 optimal strategy candidates), the statistics of the mean and simple strategies were computed from 1 000 replications.

#### 4 Conclusions

The results show that the standard genetic algorithm yields slightly better optimal strategy candidates—individual candidate strategies yield slightly higher profit and vary less among themselves than the optimal strategy candidates found by the mutation hill climbing. However, the loss of precision is relatively small even with the individual candidates and virtually disappears when strategies constructed as averages of the individual candidates are used. The result seems to be genuine and not caused by a too huge plateau in the objective function. Definitely, both GA

and MHC optimal strategies perform much better than simple strategies that do not require much optimization, e.g. the random strategy or the strategy of including the agents with most connections. Hence the conclusion could be made that the MHC performs almost as well as the standard genetic algorithm but is much faster since it does not need so many replications. Thus it seems that the mutation hill climbing could be safely used as an approximate optimization routine for robustness check and scenario analysis—at least in the case of the agent-based viral marketing studies.

## Acknowledgements

The paper has been created as a part of specific research no. MUNI/A/1203/2014 at the Masaryk University.

Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144.

## References

- [1] Kvasnička, M.: Viral video diffusion in a fixed social network: an agent-based model, *Procedia Economics and Finance*, 12(2014), 334342.
- [2] Kvasnička, M.: Search for the optimal strategy to spread a viral video: An agent-based model optimized with genetic algorithm, 32nd International Conference Mathematical Methods in Economics Conference Proceedings, (2014), 548553.
- [3] R Core Team: *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. <http://www.R-project.org/>.
- [4] Stonedahl, F. BehaviorSearch, computer software, 2010. <http://www.behaviorsearch.org/>.
- [5] Stonedahl, F.: *Genetic Algorithms for the Exploration of Parameter Spaces in Agent-Based Models*, Ph.D. thesis, 2011, available at [http://forrest.stonedahl.com/thesis/forrest\\_stonedahl\\_thesis.pdf](http://forrest.stonedahl.com/thesis/forrest_stonedahl_thesis.pdf).
- [6] Stonedahl, F., Rand, W., and Wilensky, U.: Evolving viral marketing strategies. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, ACM, 2010, 1195–1202.
- [7] Watts, D. J., and Dodds, P. S.: Influentials, networks, and public opinion formation. *Journal of consumer research* **34**(4), 2007, 441–458.
- [8] Watts, D. J., and Strogatz, S. H.: Collective dynamics of ‘small-world’ networks. *Nature* **393**, 1998, 400–442.
- [9] Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999. <http://ccl.northwestern.edu/netlogo/>.
- [10] Wilensky, U.: NetLogo Small Worlds model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 2005. <http://ccl.northwestern.edu/netlogo/models/SmallWorlds>.
- [11] Wilhite, A.: Economic activity on fixed networks. In: *Handbook of Computational Economics* (Tesaftson, J., and Judd, K. L., eds.), vol. 2. Elsevier, 2006, 1013–1045.