# Calibration Strategies for Agent-based Simulation Models with Variability

Claire Rippinger[1*], Martin Bicher[2], Florian Miksch[3]

[1]Dexhelpp, Neustiftgasse 57-59, 1070 Vienna, Austria; *claire.rippinger@dexhelpp.at
[2]Institute for Analysis and Scientific Computing, TU Wien, Vienna Austria
[3]DWH Simulation Services, Vienna Austria

SNE 27(1), 2017, 25-31, DOI: 10.11128/sne.27.tn.10364
Received: November 5, 2016, Revised: January 20, 2017
Accepted: February 15, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271 www.sne-journal.org

**Abstract.** During the development of an agent-based simulation model, the model often has to be calibrated, which means adjusting the parameters such that a reference system can be reproduced. A major problem in calibrating an agent-based simulation model is the variability of the results, due to random choices made by the agents. To reduce the variability, the numbers of agents has to be increased, which in return increases the computation time of the simulation. An attempted solution to this problem consists of increasing the numbers of agents gradually. This approach is tested with two different calibration algorithm: simulated annealing and evolutionary algorithm. Different updating schedules are applied on a test model and examined in terms of their running time and their performance. It is shown that a evolutionary algorithm with an increasing agent count manages to produce similar results as a standard calibration using only half the computation time. To conclude, the best performing calibration process is used to calibrate an existing agent-based model simulating a well known past influenza epidemic.

## Introduction

Agent-based simulation is a relatively new modelling technique [1]. It has experienced increasing application in several fields since it offers many benefits over other modelling methods [2]. According to Bonabeau, the main advantages of agent-based models are their flexibility, their natural way to describe a system, and their ability to produce an emergent behavior [3]. Contrary to other modelling techniques, it does not try to dictate the general behaviour of the system. Instead, it consists of several independent entities, called agents, which are given certain properties, behaviour and rules to change this behaviour. These agents interact with each other and their environment during a simulation run and produce the overall outcome of the system. A typical application for this sort of modelling is the simulation of epidemics.

An important step in developing a model, agent-based or other, is the calibration. It consists of adjusting the different parameters used in the simulation such that the simulated results match a given outcome. When the model is capable of reproducing a reference system, it can be used to test the outcomes of alternatives strategies in this reference system or to make predictions by simulating the reference system in the future. The nature of agent-based simulation models induces different problems regarding the calibration process. Since the result of the simulation emerges from the interaction between the agents, the outcome is hard to estimate. Therefore, it is difficult to say in what way the different parameters affect the simulation outcome. Only by running the simulation, the effects may be observed and appropriate parameter changes can be made. If the model requires only a few parameters, these adjustments can be made manually. With an increasing number of parameters, calibration algorithms are needed.

Since a calibration problem consists of minimizing the distance between the simulated data and the reference system, it can be seen as an optimization problem and algorithms from this application area can be used. The nature of agent-based simulation models requires calibration algorithms which regard the simulation as a black box and only have informations on the outcome of the simulation and not on the internal processes and calculations. The evolutionary algorithm and simulated annealing are two algorithms meeting this criterion. They are presented in Section 1.

A common problem in performing a calibration is the long computation time required to produce the results. Section 1 proposes varying the amount of agents used in the simulation as a solution to this problem. Different configurations of this method are applied to a test model. The configuration with the best performance is then used to calibrate a more complex agent-based simulation model. Both models are described in Section 2.

# 1 Methods

In a calibration process, the agent-based simulation model acts as a function: given a specific parameter set, it produces the simulated data points. These are passed to an error function which calculates the distance between the simulated data points and the data points that should be matched by the model. Often, a weighted Euclidean distance is used. This allows the error function to put more emphasis on the characteristic elements of the data of the reference system. The aim of a successful calibration is to find a parameter set which minimizes this distance. Thus a calibration can be considered as an optimization problem and the respective algorithms can be used.

In this paper, two different optimization algorithm are applied for model calibration: simulated annealing [4] [5] and evolutionary algorithm [6][7]. Flowcharts of these algorithms can be seen in Figure 1 and Figure 2.

In simulated annealing, accepting a point with a larger error should allow the algorithm to escape local minima and converge to a global minimum. The acceptance probability depends on the temperature and is decreased during the calibration process. The cooling schedule applied, as well as an appropriate choice of the neighbourhood of a point, have a large influence on the convergence of the algorithm. The evolutionary algorithm uses several points simultaneously to determine the global minimum. By choosing different methods of selecting the points and combining them to form new candidates, evolutionary algorithms can be adapted to suit the needs of many calibration problems. However, finding the optimal configuration is often difficult.

Agent-based models often have long computation times caused by high agents numbers. This is an important issue during calibration, when the model is simulated hundreds of times. The runtime can be reduced with a lower number of agents. However, the agents behaviour usually depends on random decisions. Hence,

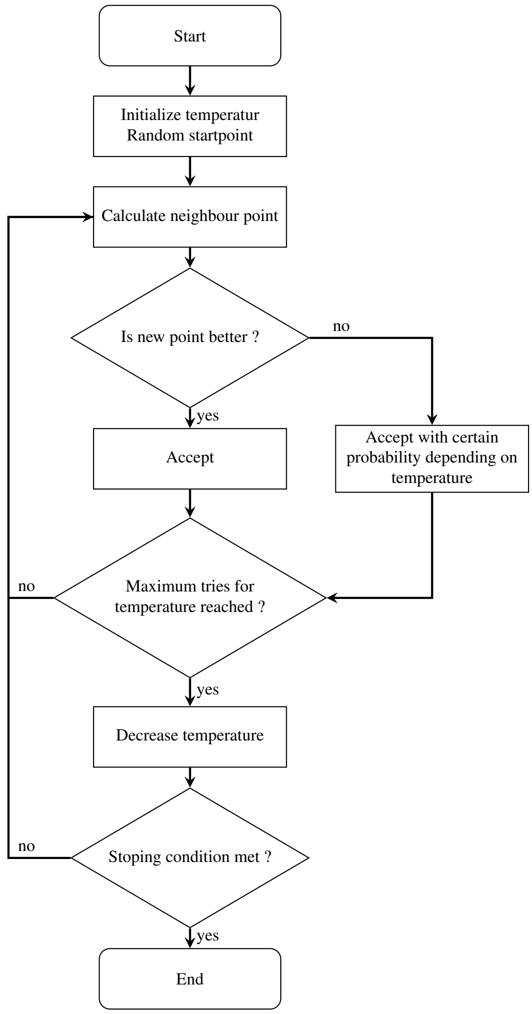


**Figure 1:** Flowchart of simulated annealing based on Kong et al. [8].

the simulation results underlie a variability. For high agent numbers, the variability is naturally low due to the law of large numbers. Lower agent numbers lead to an unwanted higher variability of the results.

A possible solution to this problem consists of varying the number of agents throughout the calibration procedure. Agent-based models can be scaled by simulating them with reduced agent numbers. This does not affect its functionality but increases uncertainty of the results due to a required upscale to the actual problem. At the start, the simulation model is run with a small number of agents, allowing to test many parameter sets in a short time period. During the calibration the number of agents is gradually increased until the targeted agent count is reached.

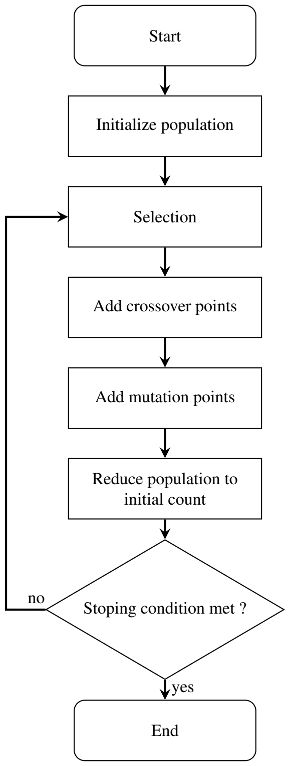


**Figure 2:** Flowchart of the evolutionary algorithm based on Kong et al. [8].

A calibration performing an increase of the agent count requires the following information:

- the starting amount of agents
- the targeted amount of agents
- the number of agent count updates
- the growth behaviour
- the number of simulation runs

After every update, the error of a newly considered parameter set is calculated with the new agent count. Normally, this error is only calculated once and then referred to in the further calibration process. However, this might cause a problem. If the error is calculated with a small agent count, it is possible that a low value has only been reached by accident and does not represent the usual outcome of the simulation performed with this parameter set. It is important to eliminate these false error values in time and not base every further search on the corresponding parameter sets. A simple solution would consist of recalculating the error of all the current parameter sets at every update of the agent count. This increases the number of simulation runs during a calibration, especially when an evolutionary algorithm is used. As a trade-off, the parameter sets are not re-evaluated directly at the next agent count update but after two agent count updates. This way, the chances are higher that the parameter set gets discarded by the algorithm before it needs to be recalculated. However, during the last update which increases the agent count to the targeted amount, every parameter set is re-evaluated to ensure that the best error has been calculated with the full agent count.

The increase of the agent count during the calibration has also an effect on the stopping conditions of the calibration algorithm. Normally, a calibration would terminate, if the error has reached a certain value. But, as mentioned above, if this error has been calculated with a small agent count, it might not be valid. Therefore the calibration is not allowed to terminate prematurely but has to perform the full amount of simulation runs.

## 2 Models

First, the calibration methods are tested on a simple SIR model which simulates the spreading of an infectious disease. In such a model, the agents represent people who can be in one of three different states: susceptible, infected or resistant. Every time, a susceptible person comes in contact with an person already infected the disease may be transmitted. After a certain amount of time, an infected person recovers from the disease and becomes resistant. This means, the person can not be infected a second time. In our model, there are two parameters that need to be calibrated: the probability $p_i$ that a healthy person is infected when they comes in contact with an infected person and the probability $p_r$ that an infected person is recovering. To create a reference system, the model is run with a known parameter set $p_i = 0.4$, $p_r = 0.05$. The output consists of the number of infected agents at a given time step. The goal of the calibration is to reproduce this curve. Previous tests have shown, that the variability of the results is negligibly small, if 10 000 agents are used, making this the target population.

Ultimately, the calibration algorithm is used to determine the parameters of a more complex agent-based model simulating an influenza epidemic. A known influenza epidemic in the year 2007 in the Austrian population is used as a reference system. The data that needs to be reproduced consists of eleven data points representing the number of newly infected people per week during the eleven week long influenza wave. For this model, the calibration needs to determine five parameters describing the contact rate between the agents, the probability of infection and development of mild or severe symptoms and the ratio of the population which is naturally immune. The reference system of both the simple SIR model and the influenza model can be seen in Figure 3.

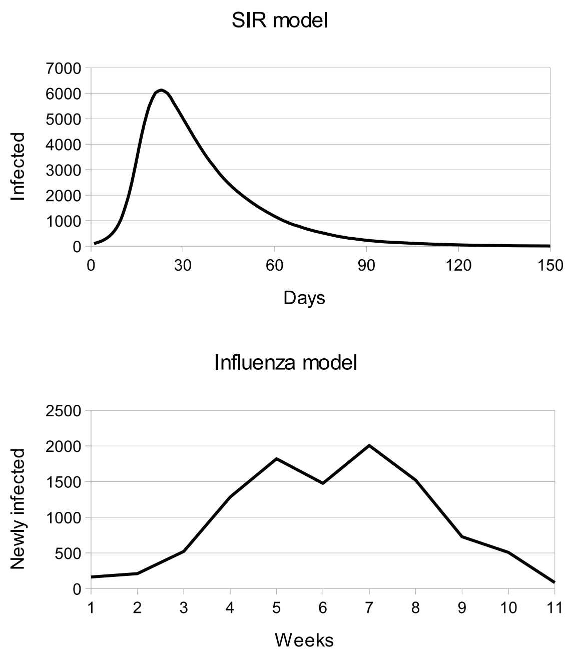


**Figure 3:** Reference systems of the SIR and the influenza model that need to be reproduced during the calibration.

## 3 Results

Both of the models described above, as well as the calibration algorithms have been implemented in Java and all the following calibrations have been calculated on a laptop running with Windows 8.1 using an Intel$^{(R)}$ Core$^{(TM)}$ i5-4200U processor and 8,00 GB RAM.

### 3.1 Results of calibration the SIR model

In order to compare the results of a calibration using an increasing amount of agents, the calibration has been performed with a constant agent count. The error is calculated using an Euclidean distance which puts a larger weight on the peak of the epidemic. In order to scale this error, the value of every data point is divided by the current agent count.

In this paper, 15 different update schedules are tested as shown in Table 1.

| | Agents at Start | Growth | Updates |
|---|---|---|---|
| **1** | 1000 | no | 0 |
| **2** | 3000 | no | 0 |
| **3** | 10 000 | no | 0 |
| **4** | 1000 | geom. | 2 |
| **5** | 1000 | geom. | 4 |
| **6** | 1000 | geom. | 8 |
| **7** | 3000 | geom. | 2 |
| **8** | 3000 | geom. | 4 |
| **9** | 3000 | geom. | 8 |
| **10** | 1000 | linear | 2 |
| **11** | 1000 | linear | 4 |
| **12** | 1000 | linear | 8 |
| **13** | 3000 | linear | 2 |
| **14** | 3000 | linear | 4 |
| **15** | 3000 | linear | 8 |

**Table 1:** Updating schedules.

The calibration is terminated after 1000 model runs. For each updating schedule the calibration is performed with three different calibration algorithm configurations providing good results in previous tests using a constant agent count. For each of these configurations, the calibration is run 10 times. For the simulated annealing, these configurations use a geometrical cooling schedule with the temperature being lowered every 10 loops by a factor 0.8, 0.85 resp. 0.9. In the evolutionary algorithm the population consists of 40 agents. A ranking based selection is applied [9]. During the crossover, 10 parameter sets are formed by calculating the mean of the two parent parameter sets and during mutation 8, 12 resp. 16 parameter sets are created by replacing one parameter with a random value.

Figure 4 shows the results of these calibrations. The bars represent the mean error of the 30 calibration results performed with the update schedule and the line represents the mean time to perform these calibrations. Note that the update schedule **3** represents a calibration performed with the targeted amount of agents throughout the whole process, making it the method applied

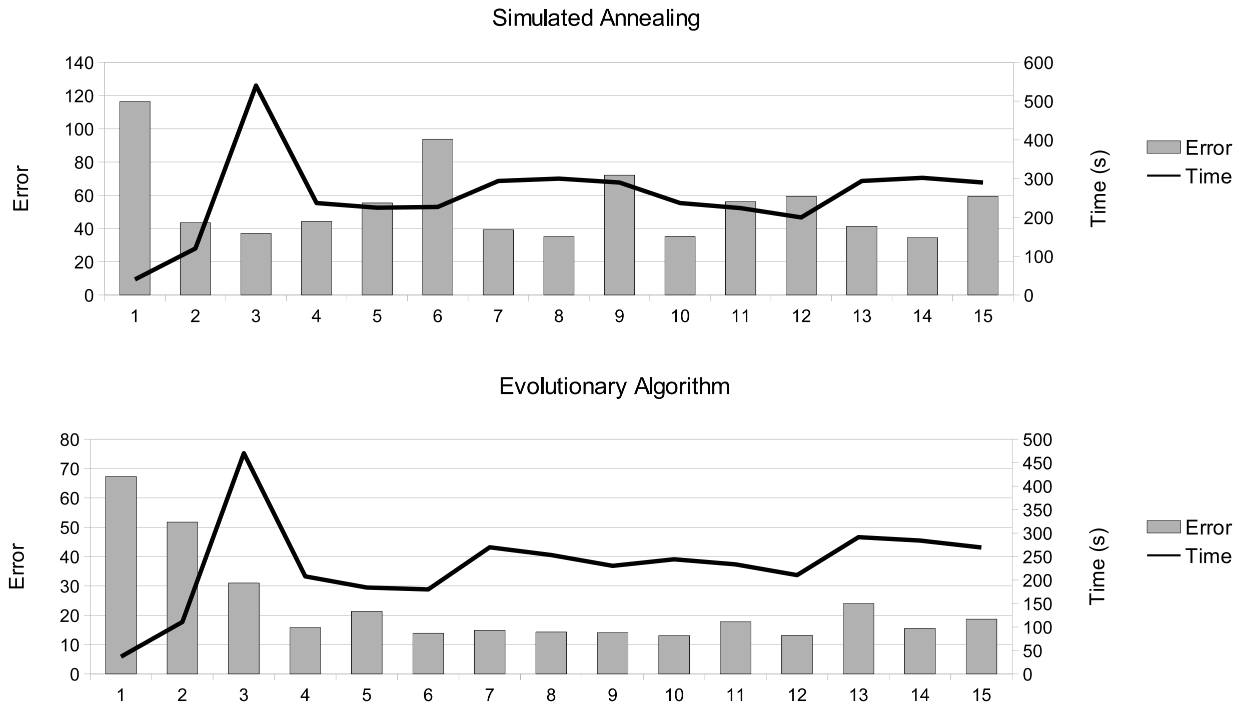


**Figure 4:** Error and running time of the calibration performed with simulated annealing and an evolutionary algorithm applying the different updating schedules listed in Table 1.

during a standard calibration. Update schedule **1** and schedule **2** represent calibrations using a constantly low agent count.

For the simulated annealing it can be seen, that only certain update schedules yield comparable results as a calibration performed with a constantly high amount of agents. Generally, it can be observed, that a higher number of agents at the start is preferable, as well as a smaller number of updates. The evolutionary algorithm produces overall better results than simulated annealing. The error obtained by updating the number of agents is even generally smaller than the one calculated by a standard calibration. There is no significant difference between the results of the different update schedules and no trend can be observed.

As expected, the computation time is much smaller for calibration updating the number of agents. In general, the running time is about half of the time required by a standard calibration represented by the update schedule **3**. Calculations using geometrical growth or a small number of starting agents require less time than those using linear growth and a higher amount of starting agents. Furthermore, the computation time decreases slightly with the number of updates performed. However it is expected that this decrease in running time is not an ongoing trend. At some point, the benefits of calculating with a lower agent count will be outweighed by the costs of re-evaluating the current population of parameter sets at every update. To verify this presentiment a new series of tests have been performed. Calibrations using geometrical growth and a starting agent count of 1000 are calculated using different numbers of updates, extending the calibrations **4**, **5** and **6** from Figure 4. The mean error and running time of these calibrations are shown in Figure 5. It can be seen, that the running time does increase with a larger number of updates without producing significantly better results.

It has been mentioned above that calibrations performed with simulated annealing provide worse results when the number of updates increases. Concerning this observation, further analysis of the error produced by these calibrations have revealed the following phenomenon: there are two different ways in which the er-

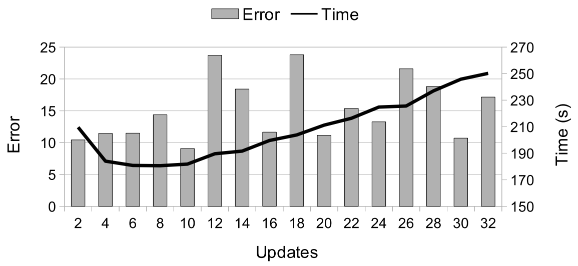


**Figure 5:** Error and running time for a calibration starting with 1000 agents and applying geometrical growth depending on the number of updates performed.

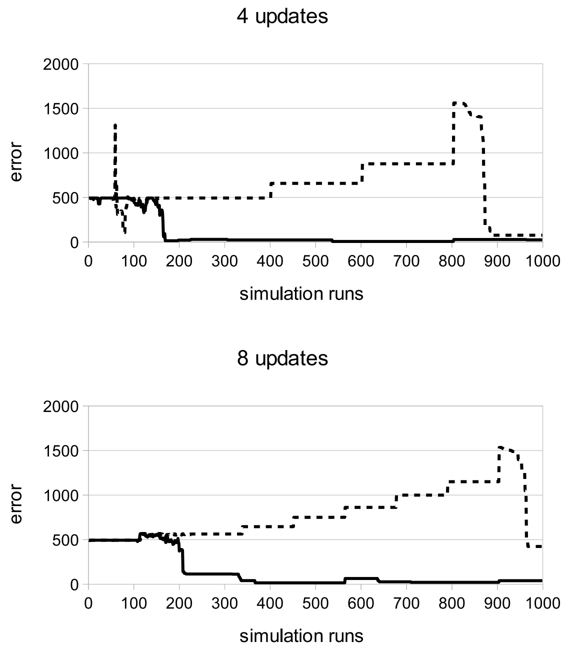


**Figure 6:** Two possible evolutions of the error calculated with simulated annealing during a calibration using geometrical growth, 1000 agents at the start and 4, resp. 8 agent count updates.

ror evolves. During some calibrations a relatively small error is already achieved using only a low agent count. The rest of the calibration process is then used for the fine tuning of the parameter set. However, if the error produced with a small amount of agents is not small, the current parameter set is not replaced with a better solution for the most part of the calibration. It is not until the simulation runs with the targeted amount of agents, that the calibration algorithm is effective and starts to lower the produced error. If the calibration process is performed using a high number of updates, the amount of simulation runs performed with the targeted agent count is too low to reach an acceptable error in time. Figure 6 shows the two different evolutions of the error during a calibration performed by simulated annealing using 4, resp. 8 updates. Each line represents one of the two typical behaviours of the error.

### 3.2 Results of calibration the influenza model

The findings of these tests are now used to calibrate the more complex influenza model. The error function used is similar to the one described at the beginning of this section. Since the data point at week six is most probably inaccurate, the weight posed on this point is very small. One simulation of the influenza model with 800 000 agents takes 300s on average. To improve the running time of the calibration, parallel computing on three kernels is applied. Since 1000 calibrations are performed, the total running time of a calibration without agent count updates would amount to approximately 28 hours. To further reduce this computation time, a calibration using 4 agent count updates is applied. The starting agent count consists of 50 000 agents which is increased geometrically to reach the targeted count of 800 000 agents. Due to the better results with the SIR model, the evolutionary algorithm is used for the calibration. The result can be seen in Figure 7. The running time of this calibration consisted of about 560 minutes, only a third of the estimated time required by a standard calibration.

## 4 Discussion

This paper briefly describes the approach of an agent-based simulation model and the procedure of calibrating such a model. The variability of the results of an agent-based model complicate the task of calibration and the usual methods of reducing this variability lead to an increase in the running time of the calibration procedure. By gradually increasing the number of agents used in a simulation, this paper proposes a possible solution to this problem. This strategy is then tested on a simple agent-based simulation model. The performance looks very promising, but leaves a few open questions that require further research. For example, it might be possible to improve the performance of the simulated

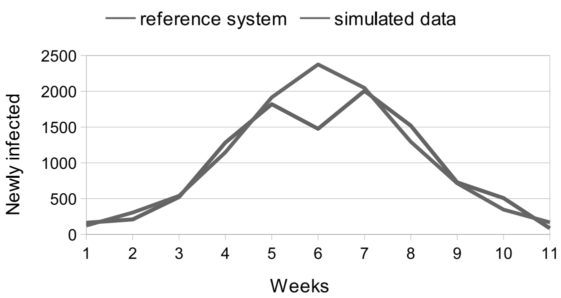


**Figure 7:** Results of a calibration of the influenza model using an evolutionary algorithm, 4 agent count updates and geometrical growth.

annealing with a cooling schedule adapted to the increase in the number of agents. Furthermore, the optimal number of updates needs to be determined. This number may depend on the simulation model used. In this paper, the strategy has only been applied to one type of agent-based simulation model. It needs to be tested if the results are similar with another type of model and what factors are beneficial to a good performance of this calibration method.

## References

[1] Macal C, North M. Tutorial on Agent-Based Modeling and Simulation PART 2: How to Model with Agents. Proceedings of the 2006 Winter Simulation Conference. 2006; pp. 73–83.

[2] Siebers P, Macal C, JGarnet, Buxton D, Pidd M. Discrete-Event Simulation is Dead, Long Live Agent-Based Simulation! *Journal of Simulation*. 2010; 4(3):204–210.

[3] Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*. 2002;99(no. Supplement 3):7280–7287.

[4] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *SCIENCE*. 1983; 220(4598):671–680.

[5] Cerny V. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimisation Theory Application*. 1985; 45(1):41–51.

[6] Holland JH. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. USA: U Michigan Press. 1975.

[7] Rechenberg I. *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Deutschland: Frommann-Holzboog Verlag. 1973.

[8] Kong C, McMahon P, Gazelle G. Calibration of Disease Simulation Model Using an Engineering Approach. *Value in Health*. 2009;12(4):521–529.

[9] Goldberg D, Deb K. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*. 1991;1:69–93.