Annales UMCS Informatica AI XII, 4 (2012) 23–37
DOI: 10.2478/v10065-012-0031-1

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Specialized Genetic Algorithm Based Simulation Tool Designed For Malware Evolution Forecasting

Vaidas Juzonis[1*], Nikolaj Goranin[1†], Antanas Cenys[1‡], Dmitrij Olifer[1§]

[1] *Vilnius Gediminas Technical University,*
Saulėtekio al. 11,SRL-I-415, LT-10223, Vilnius, Lithuania

Abstract – From the security point of view malware evolution forecasting is very important, since it provides an opportunity to predict malware epidemic outbreaks, develop effective countermeasure techniques and evaluate information security level. Genetic algorithm approach for mobile malware evolution forecasting already proved its effectiveness. There exists a number of simulation tools based on the Genetic algorithms, that could be used for malware forecasting, but their main disadvantages from the user's point of view is that they are too complicated and can not fully represent the security entity parameter set. In this article we describe the specialized evolution forecasting simulation tool developed for security entities, such as different types of malware, which is capable of providing intuitive graphical interface for users and ensure high calculation performance. Tool applicability for the evolution forecasting tasks is proved by providing mobile malware evolution forecasting results and comparing them with the results we obtained in 2010 by means of MATLAB.

1 Introduction

Nowadays malware, i.e. software created with malicious purposes in order to harm the computer software or to be installed on computer without allowance of the legal user [**1**], is considered to be one of the major threats to information security, information systems and modern communication methods. Significant shift in motivation for malicious activity has taken place over the past several years: from vandalism and recognition in the hacker community, to attacks and intrusions for financial gain. This

*vjuzonis@gmail.com
†ngrnn@fmf.vgtu.lt
‡ac@fm.vgtu.lt
§dmitrij.olifer@gmail.com

shift has been marked by growing sophistication in the tools and methods used to conduct attacks, thereby escalating the network security arms race [**2**]. It is the main reason why security specialists should have tools, which would help them to simulate the evolution processes for existing security entities, such as malware. Such tools would help analyze evolution results, predict future malware evolution tendencies and verify future threats. Nevertheless, simulation tools should be flexible and expeditiously react to any changes in the external environment.

Mobile malware is defined as viruses, worms, Trojans or other types that spread on the SmartPhones or other mobile devices running mobile OS. Although it is a relatively new malware type and not very common in the world yet its portion is highly expected to increase with the increasing smart mobile device market. IDC [**3**] predicts that 1 billion mobile devices will have gone online by 2013. Protection against malware on the mobile platforms is not very common, compared to the traditional computer systems, making them especially attractive for e-criminals. Mobile devices can also provide a variety of services to e-criminals whereas, the traditional systems cannot: SMS-spam, MMS-spam, call-proxy, etc.

Model is a physical, mathematical or logical representation of the system entities, phenomena or processes [**4**]. Modelling allows forecasting the malware propagation consequences damage [**5**] and evolution trends [**6**], understand the behaviour of malware, including spreading characteristics [**7**], understand the factors affecting the malware spread, determine the required effectiveness of countermeasures in order to control the spread and facilitate network designs that are resilient to malware attacks [**8**], predict the failures of the global network infrastructure [**9**] and many other tasks that cannot be investigated without harm to production systems in the world. Existing malware propagation models mainly concentrate on malware epidemic consequences modelling, simulating malware behaviour or economic propagation aspects and are oriented towards traditional malware. This article describes the evolution forecasting simulation tool developed for the security entities, such as mobile malware, which is capable of providing intuitive graphical interface for users and ensure high calculation performance.

Genetic algorithm [**10**] was selected as a modelling tool since it simulates natural selection by means of repeatedly evolving population of solutions and therefore may be used for predicting and modelling possible future propagation strategies. Genetic algorithm modelling has been proved to be effective in many areas such as business decision making, bioinformatics [**11, 12, 13**], information security [**14, 6, 15, 16**] and others.

Despite a large number of GA simulation environments, most of them have disadvantages (such as the reduced flexibility and calculation performance, unavailability to re-use the prepared template for other security entities, complicated interface and so on) from the point of view of information security forecasting.

2 Mobile Malware Evolution and Technical Analysis

According to [**17**] the first mobile virus was the *"Cabir"* virus which appeared on the 15th of June 2004, infected mobile phones running the "Symbian" OS and used of Bluetooth wireless network as a propagation channel. After the successful infection, the virus appended the telephone software with its code, activated the Bluetooth and started searching for another Bluetooth device to forward the infected file. Since Bluetooth network coverage is limited to 10 meters, the propagation rate of the first mobile virus was rather limited. The first Trojan malware (*„Skulls"*) also appeared in 2004, November [**18, 19**]. It infected NOKIA mobile phones, running the „Symbian" operating system. *„Skulls"* propagated by pretending to be a software update, usually as the „Macromedia Flash" update file with *.sis* extension. When the phone user activated the Trojan it changed the phone configuration settings and depicted the skulls on the screen. It also blocked many functions, such as SMS, MMS, calendar, camera, etc. The phone user could only perform telephone calls. The mobile Trojan evolution continued in 2005. A new Trojan *„Locknut.A"* was detected [**20**]. Also created for the Symbian platform it was particular in size. The *„patch.sis"* file that contained the infection was only 2KB size, making it the smallest known Trojan for the mobile platform.

The first mobile malware that started using propagation methods, other than Bluetooth, was the *„Commwarrior.A"* virus, also running on the „Symbian" platform [**21, 22**]. It was using much quicker propagation by MMS, since this method does not have limitations by distance, although Bluetooth was also supported. The MMS message included text in English, which proposed the phone user a new game, update for the antivirus software or similar. The message was sent to all contacts, found in the phone address book. In this case virus authors relied on the social engineering since when the recipient receives the message from his friend or a familiar person, the probability of opening it is higher than when it comes from the unknown number. An interesting thing is that Bluetooth was activated during the working hours and MMS were sent in the evening and at night. After each successful infection, the virus makes a one minute delay and after that starts searching for a new victim. In 2009 the Kaspersky Labs discovered a new mobile malware named *„sms.python.flocker"*, written in the Python language and designed to manipulate the mobile phone accounts. The main malware functionality is dedicated to the financial gain. The virus sends SMS messages to the specific number, which allows transferring money from the account of the infected phone to the account of the malware author [**17**].

According to [**23**], the Android mobile operating system (OS) became the most "popular" platform for the new malware. Malware on Android could be separated into 3 different types: the first, SMS-sending Trojans (such as Android/Wapaxy, Android/LoveTrp, and Android/HippoSMS), which could subscribe victims to the subscription services; the second, Maliciously modified apps, such as Android/PJApp, which could collect sensitive information and disclose it to the unauthorized persons; the third, call recording malware, like Android/NickiSpy.A and Android/GoldenEagle.A. This malware records the user's conversations and forwards them to the attacker.

3 Prior and related work

The non-GA models mainly cover malware epidemic consequences modeling, i.e. forecasting the number of infected computers, simulating malware behaviour or economic propagation aspects and are based only on current malware propagation strategies or oriented to other malware types. The first epidemiological model for computer virus propagation was proposed by Kephart [**24**]. Epidemiological models abstract from the individuals, and consider them units of a population. Each unit can only belong to a limited number of states. The SIR model assumes the Susceptible-Infected-Recovered state chain and the SIS model – the Susceptible-Infected-Susceptible chain. The technical report [**25**] described a model of e-mail worm propagation. The authors model the Internet e-mail service as an undirected graph of relationship between people. In order to build a simulation of this graph, they assume that each node degree is distributed on a power-law probability function.

Malware propagation in the Gnutella type P2P networks was described in [**8**] by Ramachandran et al. An analytical model that emulates the mechanics of the decentralized Gnutella type of peer network was formulated and the study of malware spread on such networks was performed. The Random Constant Spread (RCS) model [**26**] was developed by Staniford et al. using the empirical data derived from the outbreak of the CodeRed worm. The model assumes that a machine cannot be compromised multiple times and operates the constant average compromise rate K, which is dependent on worm processor speed, network bandwidth and location of the infected host, etc. The model can predict the number of infected hosts at time t if K is known. As [**27**] states, that although more complicated models can be derived, most network worms will follow this trend. Other authors [**28**] propose the AAWP discrete time model, hoping to better capture the discrete time behaviour of a worm. However, according to [**9**] the continuous model is appropriate for large scale models. On the other hand, in [**9**] Zanero et al propose a sophisticated compartment based model, which treats the Internet as the interconnection of autonomous systems, i.e. sub networks. Interconnections are so-called "bottlenecks". The model assumes that inside a single autonomous system the worm propagates unhindered, following the RCS model. Zou et al in [**5**] propose a two-factor propagation model, which is more precise in modelling the satiation phase taking into consideration the human countermeasures and the decreased scan as well as the infection rate due to the large amount of scan-traffic. The same authors have also published an article on modelling worm propagation under dynamic quarantine defense [**29**] and evaluated the effectiveness of several existing and perspective worm propagation strategies [**30**].

Lelarge in [**31**] introduces an economic approach to malware epidemic modelling (including botnets). Li et al. [**32**] model botnet-related cybercrimes as a result of profit-maximizing decision-making from the perspectives of both botnet masters and renters/attackers. From this economic model, they derive the effective rental size and the optimal botnet size. Fultz in [**33**] describes the DDoS attacks organized with the help of botnets as economic security games.

The increase of mobile device popularity has called out the appearance of models dedicated to the mobile malware modelling. Ruitenbeek et al. in [**34**] simulate virus propagation using the parameterized stochastic models of a network of mobile phones, created with the help of Mobius tool and provide insight into the relative effectiveness of each response mechanism. Two models of the propagation of mobile phone viruses were designed to study the impact of viruses on the dependability and security of mobile phones: the first model quantifies the propagation of MMS viruses and the second - of Bluetooth viruses. Bulygin in [**35**] analyses two viruses using different propagation methods (MMS and Bluetooth) in the SI (Susceptible->Infected) model.

Although it is widely accepted that malware evolution forecasting is an important information security task, the first model-based research paper on this topic appeared only in 2008 [**6**], which discussed the Internet worm evolution trends. It was shown that GA may be used for malware characteristic evolution forecasting. The tests have proved the effectiveness of the model in evaluating propagation rates and have shown the tendencies of worm evolution. Rather a similar concept was proposed almost one year later in [**36**]. The authors validate the notion of evolution in viruses on a well-known *Bagle* virus family. The results of the proof of- concept study showed that new viruses–previously unknown of *Bagle* family have successfully evolved starting from a random population.

In spite of a large number of GA simulation environments existing on the market, not all of them could be used for the security entities evolution forecasting. During the assessment of simulation tools based on the genetic algorithms four existing tools were evaluated: MATLAB, Global optimization packet "GMJ", EGALT, JGap packet.

As it is known, MATLAB is a powerful mathematical programming packet, capable of effective work with the data representing graphic and matrix views, also it could be easily integrated with the additional tools written in C, C++, Fortran or Java. The genetic algorithm and the direct search toolbox expanded MATLAB possibilities and could be used for the evolution forecasting.

The global optimization packet from the Genetic algorithms point of view "GMJ" has various possibilities and every year this packet is expanded by new tools for the method optimization and results analysis. The main optimization methods implemented in the "GMJ" packet are [**37**]: Genetic algorithm with parameters; genetic algorithm without parameters; neural network optimization task.

EGALT tools were created as educational application, which should help to understand the genetic algorithm internal processes. This tool was developed with the C++ programming language and worked only on the Windows platform. It could be found here: `www.ewh.ieee.org/soc/es/May2001/14/Begin.htm`. These tools allow students to choose gens amount, population size, iteration number, mutation and crossing possibilities, also a crossing type could be chosen.

The main idea of the Java genetic algorithm packet (JGap) is to develop a common genetic application function system (library) for the implementation of main genetic

operations. Such system could be used within the application development process and integrated into a new application if such functionality is necessary [**38**].

Evaluated tools \ Parameters	Intuitive graphical user interface	Easy data input	Set of Genetic algorithm parameters used in the tool						Graphical result representation	Price
			Mutation	Crossing	Standard Fitness function	Fitness function modification	Gens dependencies	Multiple gens parameters		
MATLAB	-	-	+	+	+	+	-	-	+	-
Global optimization packet "GMJ"	-	-	+	+	+	+	+	+	-	+
EGALT	+	+	+	+	+	-	-	-	+	+
Java genetic algorithm packet	-	+	+	+	+	+	+	+	+	+

Standard simulation tools have some common disadvantages. Often the genetic algorithm is implemented in applications as the additional functionality and such approach reduces the flexibility and calculation performance of the application. Another problem is that the user cannot often add easily a new parameter or modify existing fitness function or gens probabilities. As a result, some application could not be re-used to calculate evolution forecasting for other security entities or for the same entity with new parameters. Besides, simulation tools are quite complicated and require from the user additional knowledge about the tool interface or internal functions and procedures before he will be able to start working with them. Trying to solve the described problem or reduce its influence, it was decided to develop the basic simulation tool based on the genetic algorithm, which could be effectively used only by security specialists for the evolution simulations and forecasting of information security entities.

4 Requirements for specialized simulation tools

According to common application development methodologies, the functional and non-functional requirements were described in detail before starting the development process of simulation tools.

4.1 Functional requirements

The main attention in specifying requirements for the tool was paid to the simplification of the security entities evolution simulation process and making it more intuitive and effective. Seeking to achieve the specified goals, the following requirements should be implemented:

- Possibility to insert and modify the initial data, such as parameters' names, from the data sources (csv documents);
- Provide reports on all evolution steps and best evolution results, which are noticed during the simulation.

Detailed description of the functional requirements:

1. Fitness function:

1.1. Fitness function could be provided for the application by using main arithmetic symbols, trigonometry, evolution, exponentiation, Sum sign ($\sum$) , multiplication sign ($\prod$);
2.2. Inserted fitness function could be stored and re-used later in other calculations;
2. Gens:
2.1 Before gens input, the user could choose gens amount and gens parameters amount;
2.2 Inserted gens parameters could be stored and reused later in other calculations;
3. Other parameters:
3.1 The user could choose such parameters as:
- the population individual amount – how many chromosome sets will be generated;
- iteration amount – number of calculation loops;
- mutation probability – with what probability gens could mutate.

4. Results output:
4.1 Graphical representation with the best fitness function result, generated during each iteration step. According to this graphic description we can analyze how evolution process developed and changed the population individual fitness during this process;
4.2 At the end of all calculations, the application provides information about the best population individual and its gens values. The result is saved in a separate file.

4.2 Non - functional requirements

User interface requirements:

- Intuitive graphical user interface (GUI);
- Interface management performs by keyboard and mouse;
- Main windows do not have the input fields, which are not related to the main data or calculation process.

Usage requirements:

- Data input should be intuitive and easy;
- Application should inform user about errors or prevent wrong information input;
- Application should be intuitive and ensure that user could use it without the additional training;
- Compatibility with Internet services should be ensured.

Calculation requirements:

- Application should implement genetic algorithms methods and perform calculations in optimized way;

- Application should put calculation results into the csv data files. Additional requirements could be used to ensure that data transfer is performed in an optimal way.

Application requirements:

- Operation system – Microsoft Windows and Unix;
- Microsoft Office (Microsoft Excel);
- Browser: Internet explorer; Mozila; Google Chrome; Opera; Safari;
- Java platform.

4.3 Developed application

According to the described functional and non-functional requirements, simulation tools were developed. The user interface was designed as one main window, where the user can see all parameters, which will be used during the simulation. Input implemented by using the csv files and such approach allows to re-use the existing data in the future. All results simulated during the evolution forecasting process are also stored in the csv file and it means that the calculation results could be easily re-used in other simulations as an input data.

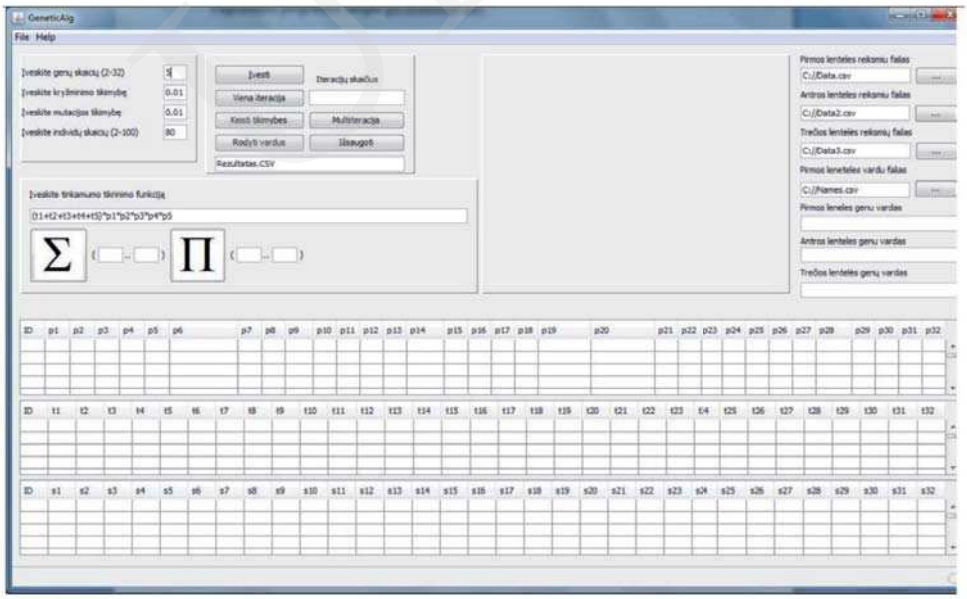

Fig. 1. Developed simulation tools main window.

5 Specialized tool for mobile malware evolution forecasting and result verification

5.1 General GA assumptions

In the case of GA modelling the main task consists of three parts: appropriate selection of chromosome structure, which represents the solution, definition of the fitness function and GA operating conditions, such as population size, mutation rates, parent selection, etc. In our case we define the propagation strategy as combination of methods and techniques, used by malware to insure malware population increase. In the current study, we have chosen to model strategies for a theoretical mobile virus, which aims at infecting the largest number of mobile devices during a fixed relatively short period of time.

GA consists of initialization, selection and evolution stages. During the initialization stage initial population of strategies is generated. Each strategy is represented as a chromosome. In the selection stage strategies are selected through a fitness-based process and in the case the termination condition is not met, evolutionary mechanisms are started. If the termination condition is satisfied, algorithm execution is finished. If not – evolutionary mechanisms are activated.

The initial population is generated on a random basis, i.e. each individual, representing a separate strategy is combined from random genes' values. The population size N is equal to 50. The population size remains constant after each new generation. The algorithm would stop producing new generations in the case the number of generations reached 100. Parent selection is random. The mutation operator is activated to each newly generated individual with a 0.05 probability.

5.2 Experiment results

During the experiment [**37**] GA consisted of initialization, selection and evolution stages. At the initialization stage the initial population of strategies was generated. Each strategy was represented as/by a chromosome. In the selection stage strategies were selected through a fitness-based process and in the case the termination condition was not met, evolutionary mechanisms were started. If the termination condition was satisfied, the algorithm execution process was terminated. If not – evolutionary mechanisms were activated.

The initial population was generated on a random basis, i.e. each individual, representing a separate strategy was combined from random genes' values. The population size N was equal to 50. The population size remained constant after each new generation. The algorithm would stop producing new generations in the case the number of generations reached 100. Parent selection was random. The mutation operator was activated to each newly generated individual with a 0.05 probability.

Modelling performed in 2010 showed that the best fitness result achieved during the algorithm test was equal to $F(S_d) = 0.023$ and the best result was achieved in the 42^{nd} generation [**37**].

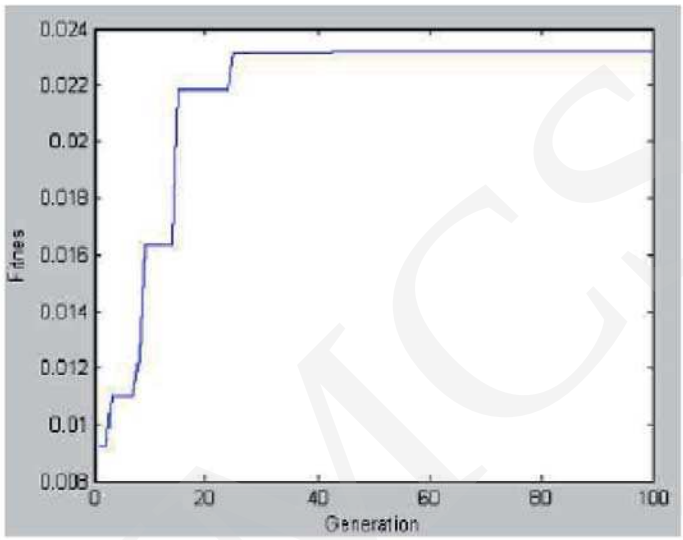

Fig. 2. Best strategy fitness change graph [**37**].

The quality assurance process of a new simulation tool with old parameters showed the following results: the result bias was smaller than 0.01. The best fitness result was $F(S_d) = 0.0229$. The best strategy fitness change looks like this:

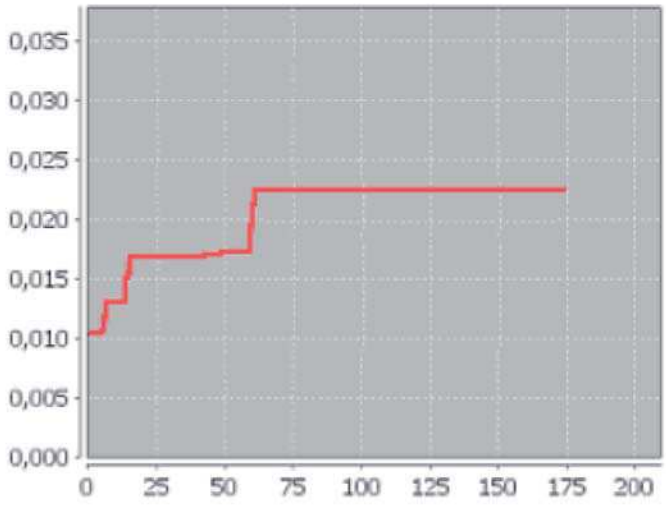

Fig. 3. Best strategy fitness change graph.

This shows that the achieved results are rather similar to those in 2010 [**37**] and a small difference could be explained by "random" nature of genetic algorithms.

The second quality assurance step was to find out how the developed simulation tool could be re-used to calculate the evolution forecasting for other or changed security entities. The gens parameters matrix was modified according to the changes on the mobile market and new parameters like Android OS were added, also some gens probabilities were changed (Table 1).

Table 1. New Gens parameter values.

Gens Nr.	1	2	3	4	5	6		
Value	MMS	SMS	Bluetooth	E-mail	Wi-Fi	Address book	Accepted / Dialed numbers	Random
p	0.7	0.7	0.2	0.05	0.4	0.4	0.1	0.3
t	3	3	1	0.01	0.005	0.01	0.01	0.03
Gens Nr.	7		8	9		10		
Value	Scan BT	Non scan BT	Address book; e-mail address DB	Scan	Non scan	Google Android	Symbian	iOS
p	0.2	0	0.2	0.4	0	0.329	0.306	0.16
t	0.5	0	0.01	0.005	0	0.005	0.005	0.005
Gens Nr.	11					12/13/14	15	16
Value	Blackberry	Nokia	RIM	HTC, Samsung, Sony Ericsson, Motorola	Apple	EXP_ON/OFF	Expl1	Expl2
p	0.144	0.31	0.07	0.36	0.21	0	0.1	0.05
t	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005
Gens Nr.	17	18			19			
Value	Expl3	10 a.m – 8 p.m.	Always	8 p.m. - 10 a.m.	10 a.m – 8 p.m.	Always	8 p.m. - 10 a.m.	
p	0.2	0.4	0.6	0.1	0.4	0.6	0.1	
t	0.005	0.01	0.01	0.01	0.01	0.01	0.01	
Gens Nr.	20			21	22			
Value	10 a.m – 8 p.m.	Always	8 p.m. - 10 a.m.	Stand. Funct.	Web - update	E-mail	Wi-Fi	
p	0.4	0.6	0.1	0	0.1	0.01	0.01	
t	0.01	0.01	0.01	0.005	0.5	1	0.5	

New simulation results generate the new best fitness strategy. The best fitness result was $F(S_d) = 0.0059$. We ought to mention that the best fitness result is smaller than it was with the previous parameters, but it is natural as during the last simulation new parameters and probabilities were used.

According to the data saved in the result.csv file we can extract the best fitness strategy chromosome and in our case it is:

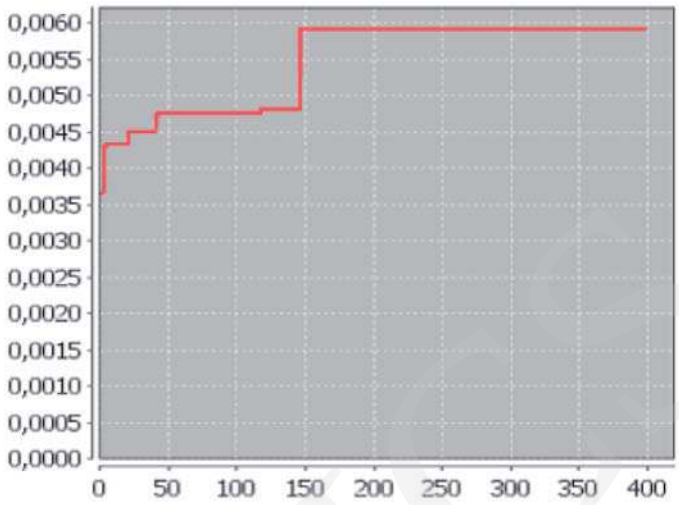

Fig. 4. Best strategy fitness change graph for new gens parameters.

S_d=(TRANSF1="SMS", TRANSF2='"Bluetooth", TRANSF3="e-mail", TRANSF4="Wi-Fi", TRANSF5="MMS", NR="Address book", BT="Scan BT", EMAIL="E-mail address", WIFI=" Scan", OS_PLATF="Google Android", TEL="HTC, Samsung, Sony Ericsson, Motorola" EN_EXPL_N="EXP1_ON_ALL", EXPL_1=" Expl1", EXPL_2=" Expl2", EXPL_3=" Expl3", NR_TIME="Always", BT_TIME="Always", WIFI_TIME="Always", EXEC=" Stand. Funct.", EXEC_CHAN="web-update",)

Comparing the last simulation results with the previous one, we should conclude that gens such as TEL and OS_PLATFORM have changed. Currently they show domination of Google Android OS and there are even three dominating telephone brands, such as HTC, Samsung, Sony Ericsson, Motorola, that could be evaluated by malware developers as perspective for malware spreading. These changes appeared due to the changes in Mobile devices and Mobile OS markets. Large part of this market presently belongs to the devices working on the Android OS platform. In our previous simulation OS_PLATF was "Symbian" and Nokia devices were dominating. The fitness results decreased by 3.7 times in comparison with our previous simulation, but such results were predictable because the Mobile devices OS market is divided among a few big players and malware working on one OS could not work on another OS. Also lately a lot of countermeasures, like OS and application patch and fix, antivirus software for the Mobile device have appeared, which decreased the malware threat probability and its propagation rates.

6 Conclusions

In this article the specialized tool based on the principles of the genetic algorithms for malware and other information security entities evolution forecasting was presented. The main tool difference from the existing modelling tools is its usability and tuning for the specific problem area to be modelled, possibility to represent specific information security issues, such as interrelated gens while modelling malware evolution and others.

Practical applicability of the developed tool was tested by modelling evolution of rapidly evolving mobile malware. Tool functioning correctness was proved by comparing the earlier modelling results with those obtained by the tool supplied with the same initial parameters.

Evolution modelling with the changed initial parameters has shown changes in the results we have obtained in our previously performed modelling tests. These changes depict the one in the currently popular mobile device platforms (the Android platform is currently more prospective for malware spreading than the Symbian one, that is no longer popular) and general decrease of malware spreading efficiency compared to the 2010 results due to the increase of a number of mobile OS platforms. Still that does not show that the total population of infected mobile devices in the world has decreased, but it rather shows the infection spreading speed. The decrease of spreading efficiency could be also explained by the increased awareness about mobile malware and countermeasures applied.

References

[1] Monga R., MASFMMS: Multi Agent Systems Framework for Malware Modeling and Simulation, Lecture Notes in Computer Science 5269 (2009): /2009, 97.

[2] Barford P., Yegneswaran V., An Inside Look at Botnets, Advances in Information Security, 27 (2007): 171.

[3] Shah A., IDC: 1 Billion Mobile Devices Will Go Online by 201, IDG News Service, Interactive (2009); http://www.pcworld.com

[4] Defense Acquisition University, Systems Engineering Fundamentals: January 2001, Defense Acquisition University Press. (2001).

[5] Zou C.C., Gong W., Towsley D., Code Red Worm Propagation Modeling and Analysis, CCS '02: Proceedings of the 9th ACM Conference on Computer and communications security, ACM. (2002): 138.

[6] Goranin N., Cenys A., Genetic Algorithm Based Internet Worm Propagation Strategy Modeling, Information Technology And Control. 37 (2008): 133.

[7] Garetto M.W., Towsley G. D., Modeling Malware Spreading Dynamics, Proceedings of INFOCOM (2003).

[8] Ramachandran K., Sikdar B., Modeling malware propagation in Gnutella type peer-to-peer networks, Proceedings of the Parallel and Distributed Processing Symposium, IPDPS 20 (2006): 8.

[9] Serazzi G., Zanero S., Computer Virus Propagation Models. Lecture Notes in Computer Science (2004): 26.

[10] Holland J. Adoption in natural and artificial systems, The MIT press (1975).

[11] Birchenhall C., Kastrinos N., Metcalfe S., Genetic algorithms in evolutionary modeling, Journal of Evolutionary Economics 7 (1997): 375.
[12] Hill R.R., McIntyre G.A., Narayanan S., Genetic Algorithms for Model Optimization, Proceedings of Simulation Technology and Training Conference (SimTechT) (2001).
[13] Stender J., Hillebrand E., Kingdon J., Genetic Algorithms in Optimization, Simulation and modeling, IOS Press (1994).
[14] Faraoun K.M., Boukelif A., Genetic Programming Approach for Multi-Category Pattern Classification Applied to Network Intrusions Detection, International Journal of Computational Intelligence 3(1) (2007): 79.
[15] Goranin N., Cenys A., Genetic algorithm based Internet worm propagation strategy modeling under pressure of countermeasures, Journal of Engineering Science and Technology Review 2 (2009): 43.
[16] Goranin N., Cenys A., Malware Propagation Modeling by the Means of Genetic Algorithms, Electronics and Electrical Engineering 86 (2008): 23.
[17] Kaspersky Lab, Kaspersky Lab reports (2009); Interactive: http://www/.kaspersky.com
[18] Naraine R., Cell Phone Security: New Skulls Mutant Comes with Virus Extras (2004); Interactive: http://www.eweek.com
[19] Niemela J., F-Secure Virus Descriptions: Skulls D. F-Secure Corporation (2005); Interactive: http://www.f-secure.com
[20] Jarno U., Disinfection tool for SymbOS/Locknut.A (Gavno.A and Gavno.B), F-Secure Corporation (2005); Interactive: http://www.f-secure.com/
[21] F-Secure, Worm: SymbOS/Commwarrior, F-Secure Corporation (2006); Interactive: http://www.f-secure.com/
[22] Sundgot J., First Symbian OS virus to replicate over MMS appears (2005); Interactive: http://www.infosyncworld.com/
[23] McAfee Threats Report: Third Quarter 2011 [Reviewed 2011-11-14], Link: http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q3-2011.pdf
[24] Kephart J. O., White S. R., Directed-graph epidemiological models of computer viruses, Proceedings of IEEE Computer Society Symposium (1991): 343.
[25] Zou C.C., Towsley D., Gong W., Email Virus Propagation Modeling and Analysis, Technical report TRCSE-03-04, University of Massachusetts (2004).
[26] Staniford S., Paxson V., Weaver N., How to 0wn the Internet in Your Spare Time, Proceedings of the 11th USENIX Security Symposium, USENIX Association (2002): 149.
[27] Nazario J., Defense and Detection Strategies against Internet Worms, Artech House Publishers (2003).
[28] Chen Z., Gao L., Kwiat K., Modeling the Spread of Active Worms, Proceedings of NFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, IEEE Societies 3 (2003): 1890.
[29] Zou C. C., Gong W., Towsley D., Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense, WORM '03: Proceedings of the 2003 ACM workshop on Rapid malcode, ACM (2003): 51.
[30] Zou C. C., Gong W., Towsley D., On the performance of Internet worm scanning strategies Performance Evaluation, Elsevier Science Publishers B. V. 63 (2005): 700.
[31] Lelarge M., Economics of Malware: Epidemic Risks Model, Network Externalities and Incentives, Proceedings of Fifth biannual Conference on The Economics of the Software and Internet Industries (2009).
[32] Li Z., Liao Q., Striegel A., BotnetEconomics: Uncertainty Matters, Managing Information Risk and the Economics of Security, Springer US. (2009): 1.
[33] Fultz N., Distributed attacks as security games, Master thesis, US Berkley School of Information (2008).

[34] Ruitenbeek E.V., Courtney T., Sanders W.H., Stevens F., Quantifying the Effectiveness of Mobile Phone Virus Response Mechanisms, IEEE/IFIP International Conference on Dependable Systems and Networks (2007): 790.

[35] Bulygin Y., Epidemics of Mobile Worms, Performance, Computing, and Communications Conference, 2007. IPCCC 2007, IEEE International (2007): 475.

[36] Noreen S., Murtaza S., Shafiq M.Z., Farooq M., Evolvable malware, GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM (2009): 1569.

[37] Global and Discrete Optimization, Kauno technologijos universitetas, Programinės įrangos katedra, [Reviewed 2010-04-17]; Link: http://soften.ktu.lt/m̃ockus.

[38] Global Optimization Toolbox, [Reviewed 2010-10-26]; Link: http://www.mathworks.com/help/toolbox/ gads/ ga.html.