Optimal Choice of Random Variables in D-ITG Traffic Generating Tool using Evolutionary Algorithms

M. R. Mosavi*(C.A.), F. Farabi* and S. Karami*

Abstract: Impressive development of computer networks has been required precise evaluation of efficiency of these networks for users and especially internet service providers. Considering the extent of these networks, there are numerous factors affecting their performance and thoroughly investigation of these networks needs evaluation of the effective parameters by using suitable tools. There are several tools to measure network's performance which evaluate and analyze the parameters affecting the performance of the network. D-ITG traffic generator and measuring tool is one of the efficient tools in this field with significant advantages over other tools. One of D-ITG drawbacks is the need to determine input parameters by user in which the procedure of determining the input variables would have an important role on the results. So, introducing an automatic method to determine the input parameters considering the characteristics of the network to be tested would be a great improvement in the application of this tool. In this paper, an efficient method has been proposed to determine optimal input variables applying evolutionary algorithms. Then, automatic D-ITG tool operation would be studied. The results indicate that these algorithms effectively determine the optimal input variables which significantly improve the D-ITG application as the time cost of determining optimal DITG variables in automatic GA, ICA and ACO based methods has been improved up to 67.3 %, 69.7 % and 82.2 %, respectively.

Keywords: Ant Colony Optimization, D-ITG, Genetic Algorithm, Imperialist Competitive Algorithm, Optimization.

1 Introduction

Recently, the computer networks have been extremely complicated in software, protocols, equipment and etc [1] and lots of efforts have been made to understand the behavior of these networks [2]. There are many factors affect the network performance, including network congestion and load, network equipment and hardware, the amount of network users, the power of wireless network signal and the operating system used by users [3, 4]. In order to evaluate the networks performance there are several tools such as DSLProbe [5], Asymprobe [6], Spruce [7], Pathload [8], Netalyzy [9], Iperf, Netperf, IPtraffic, MGEN and D-ITG which measuring the parameters affecting the network performance. D-ITG has significant advantages over other traffic generators. D-ITG supports IPv4 and IPv6 and measures key parameters such as throughput, latency, jitter and packet loss in packet level. This tool is capable to reproduce various known application program traffic (e.g. Telnet, VoIP, DNS and network games) and support Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Stream Control Transmission Protocol (SCTP), Datagram Congestion Control Protocol (DCCP) and also Internet Control Message Protocol (ICMP) in transfer layer. Besides presenting exact outputs, the results are highly dependent to input parameters like packet number, packet size and packet sending time. Thus, the determination of optimal input parameters is a major challenge in applying this software. In this research, Evolutionary Algorithms (EAs) have been used to determine the best value for the Internet Distributed Traffic (IDT) and Packet Size (PS) random variables. Therefore optimized results can be obtained based on these variables. In the following, we will investigate the effects of proposed methods to improve D-ITG operation. At the second section, the purpose of evaluating computer networks performance and their

Iranian Journal of Electrical & Electronic Engineering, 2014.
Paper first received 25 Apr. 2014 and in revised form 18 Jan. 2015.
* The Authors are with the Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
E-mails: M_Mosavi@iust.ac.ir, f.farabii@gmail.com and karami_samaneh@yahoo.com.

effective parameters will be presented. In addition, in order to measure computer networks, D-ITG tool will be introduced. In the third section, we examine the structure of proposed EAs and at the forth section the simulation results will be presented. The research conclusion is presented at the fifth section.

2 Computer Networks Evaluation using DITG

Evaluation of computer networks efficiency is performed in order to compare the proposed networks structures and selecting the best one, based on the observable parameters affecting the service quality. Generally, the purposes of evaluation the performance of computer networks are as following:

- Determining the criteria of performance of existing systems.
- Modeling of existing systems and systems will be proposed in the future.
- Developing analytical basics.
- Finding the ways to apply and confirm new approaches, e.g. constructing and evaluating performance models.

There are several parameters defining service quality including: bandwidth, throughput, average bit rate, latency, jitter, bit error rate, packet loss, signal to noise ratio, attenuation and line length. Recently, many efforts have been made to model internet traffic and derivate internet behavioral model. Excessive complication of huge topologies and their traffic characteristic have made the model analyzing so difficult. In this situation, traffic generation and network behavior simulation using simulation tools would be an efficient method to analyze complicated networks operation. In recent years, increasing improvements in traffic generator tools lead to a new generation of traffic generators which perform simulation, analysis and real traffic generation thoroughly. Due to the fundamental differences between the traffic generator tools in performance procedures, the operational environment, supporting protocols of various network layers, architecture and the generated output, examining and comparing them would be a difficult task. D-ITG is able to generate traffics with IPv4 and IPv6 base through accurate repetition of traffic load in the internet application programs. Moreover, D-ITG is capable to measure the network performance based on performance measurement common parameters such as throughput, latency, jitter and packet loss. This tool could apply random models to PS and IDT. Determining the distributions of PS and IDT random variables make it possible to apply various reconstruction processes for packet generating. This tool is able to regenerate traffic statistical specifications of identified various applications such as Telnet, VoIP, DNS and network based games. D-ITG supports TCP, UDP, SCTP, DCCP and also ICMP protocols in transfer level. Recently, D-ITG supports SCTP and DCCP protocols in order to overcome some limitations of UDP and TCP [10].

The distributed structure of D-ITG traffic generator is represented in Fig. 1. This structure is made up of four main parts including sender/receiver, logger, controller and analyzer. All these platform components apply multithread programming [11, 12].

Regardless all the benefits of D-ITG mentioned above, determining the proper and optimal input variables is an important challenge in using of this tool. In the next section, we will introduce some different EAs to find optimal input variables for improving the accuracy of DITG.

3 Review of Proposed EAs for Optimizing D-ITG

In this section an attempt has been made to use EAs for optimizing the performance of DITG. Optimization algorithms would lead to the most optimal response by reducing the search space and prevent getting stuck in local minima by providing the requisite perturbation to bring it out of the local minima.

An effective optimization algorithm would be able to find the optimal response without considering the initial values of optimization variables. Three of the most common EAs and application of them in finding D-ITG optimal input variables will be described subsequently.

3.1 Optimization Based on Genetic Algorithm

Genetic Algorithms (GAs) are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information [13]. An implementation of a GA begins with a random population of chromosomes. In our proposed algorithm, each chromosome has a length equal to three genes named packet size, number of packets and inter departure time. Then these chromosomes will be evaluated and allocated with reproductive opportunities in such a way that those chromosomes which represent a better solution to the target are given more chances to reproduce. The fitness of each chromosome is determined by fitness function. In this research, the fitness values of chromosomes are obtained by Eq. (1):

$$Fitness = \frac{b * BitRate}{\alpha * delay + \beta * Jitter + \gamma * PacketLoss} \tag{1}$$

According to above fitness function the final purpose is to find the optimal responses in a way that maximum bit rate, minimum delay and minimum jitter will be achieved. Furthermore, the packet loss would be minimized in responses. Thus, it could be guaranteed that the best response for the under test network can be achieved by resultant D-ITG input parameters. Moreover, by choosing appropriate coefficients in fitness function, we would be able to determine each of four parameters (bit rate, delay, jitter and packet loss) weight based on the importance of them in our network.

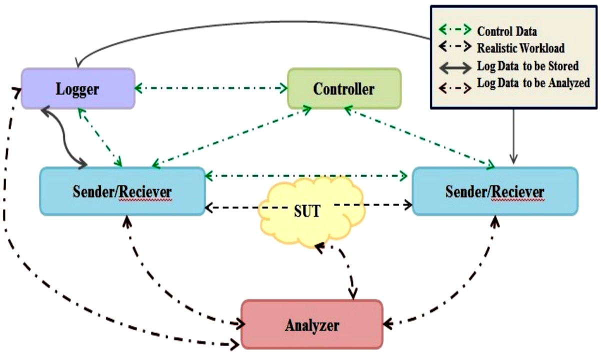

Fig. 1 D-ITG architecture.

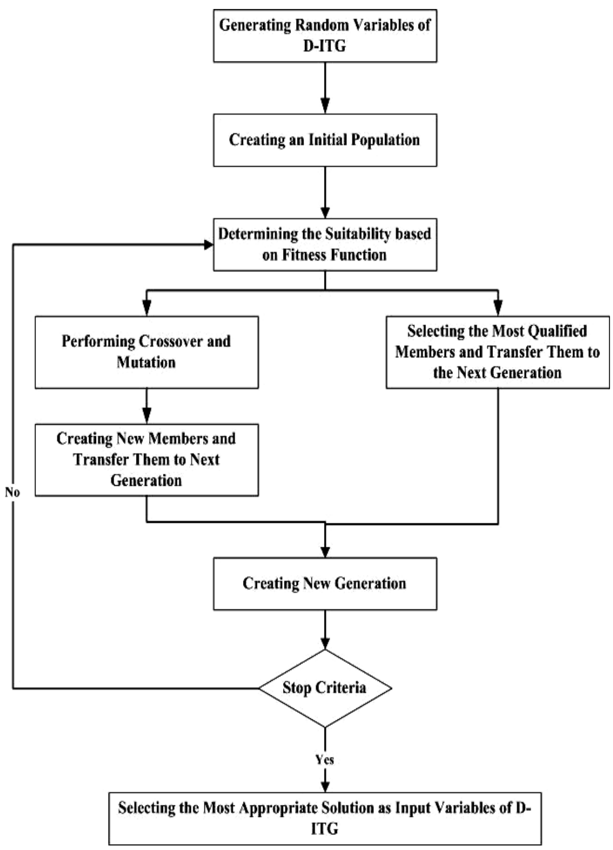

Fig. 2 Flowchart of GA based proposed algorithm.

Fig. 2 shows the flowchart of this algorithm. Each stage of this flowchart will be described subsequently. In the first step of the algorithm all chromosomes are evaluated using the fitness function to determine their fitness. These fitness values are then used to decide whether the chromosomes are eliminated or retained. According to the principle of survival of the most fitted, the more adaptive chromosomes are kept, and the less adaptive ones are discarded in the course of generating a new population. After selection has been carried out the crossover can occur. This can be viewed as creating the next population from the selected chromosomes.

In one-point crossover, first, one point is selected as the reproduction start point. Then, the first chromosome left hand genes would be copied in new chromosome till the start point of reproduction. Then, the second chromosome right hand genes are copied into the right hand genes of the new chromosome. In other word, swapping the fragments between the two parents produces the offspring. After crossover we apply the mutation operator. In mutation process, one of the genes of the randomly selected chromosome would be changed randomly. In our algorithm the mutation rate is applied with 1 % probability.

After the process of selection, crossover and mutation is completed, the next population can be evaluated. The algorithm continues as described until the stopping conditions will be reached. In this case reaching a maximum number of iterations or having no more improvement (or improvement less than a determined threshold) in final results for several consecutive iterations may be chosen as stopping criteria. At the end of the algorithm, the most fitted chromosome would be selected as the optimal solution and each of the genes would be regarded as one of the D-ITG random variables.

3.2 Optimization Based on Imperialist Competitive Algorithm

Imperialism is the policy of extending the power and rule of a government beyond its own boundaries. Imperialist Competitive Algorithm (ICA) is a novel global search strategy that uses imperialism and imperialistic competition process as a source of inspiration. This algorithm is based on that in real world countries try to extend their power over other countries in order to use their resources and bolster their own government [14-16]. The block diagram of this algorithm is shown in Fig. 3. As shown in this figure, ICA starts with some initial countries that are randomly dispread in search space. Each country has a vector of cultural, social and economical features. Here the countries are arrays of three real numbers which each of them represents one of the D-ITG input parameters. After forming primary countries, the power of each country is equivalent to fitness value and is calculated by using Eq. (1) [16]. After calculating the power of all countries, some of the stronger countries (N_{imp} most fitted countries) in the population are selected to be the imperialists and then during a competitive process all the other N_{col} countries are divided among imperialists based on their normalized power. In fact the initial number of colonies of each imperialist should be directly proportionate to its normalized power. The normalized power of an imperialist is defined by [17]:

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \tag{2}$$

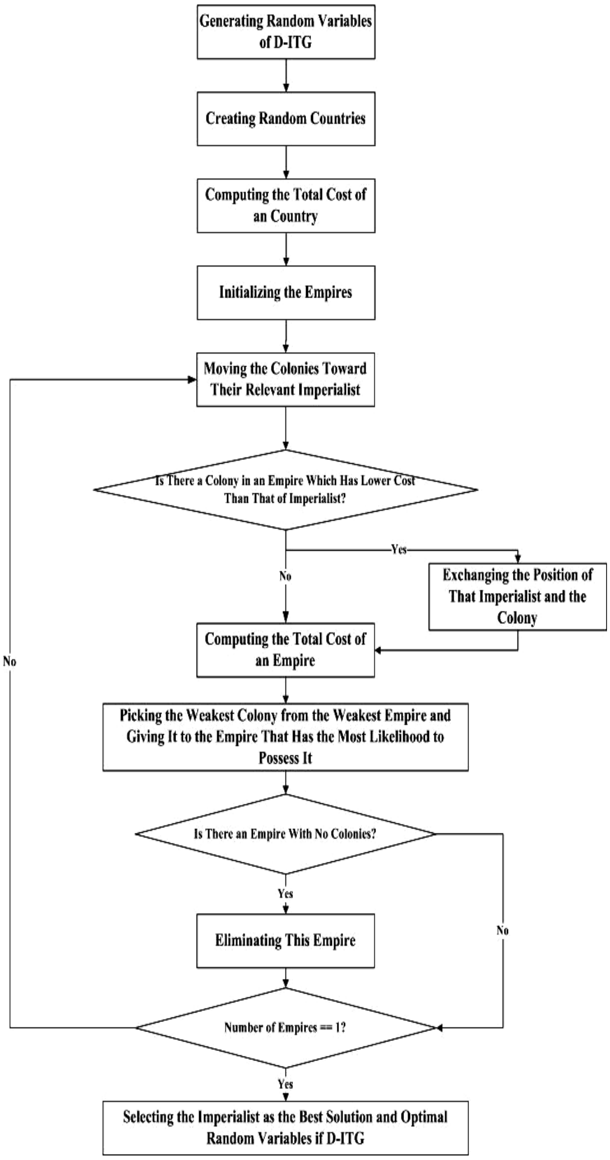

Fig. 3 Flowchart of ICA based proposed algorithm.

where C_n is the normalized cost of n-th imperialist and is defined as difference between the cost of imperialist and total cost of all imperialists. The number of colonies of each empire is calculated by Eq. (3) [17]:

$$N.C_n = round\{P_n.N_{col}\} \tag{3}$$

After creating the initial empires, colonies start moving toward their relevant imperialist state. This process is called assimilation policy. In assimilation, each colony moves on the line that connects the colony and it's imperialist by x units, which x is a random variable with uniform distribution. To creating new positions for colonies around the imperialist in different directions a random deviation angle θ is added to the direction of movement. If this movement causes to find a colony with better position (more power) than that of imperialist, the colony moves to the position of the imperialist and vice versa.

The total power of an empire is defined by the power of imperialist state plus a percent of the mean power of its colonies.

In imperialistic competition, the weakest colony of weakest empire would be transferred to the most powerful empire and the empires with no colonies would be omitted and the imperialist would be added to the most powerful empire. As a result of imperialistic competition, the colonies of powerless empires will be divided among other imperialists and the empire will be collapsed. This process would be repeated until just one empire remains or a certain amount of decades be passed.

As a result, all the countries converge to a state in which there exists only one empire that possesses all other countries as its colonies. In this situation the imperialist consists of an array of variables which is an optimal solution of the problem. In addition to this stopping condition, in ICA like other EAs reaching a maximum number of iterations (decades) or having no improvement in final results for several consecutive iterations may be chosen as stopping criteria. In these situations, the solution of problem is determined by the imperialist of the most powerful empire.

3.3 Optimization Based on Ant Colony Optimization Method

The inspiring source of Ant Colony Optimization (ACO) is the foraging behavior of real ants. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the ant deposits a chemical pheromone trail on the ground. The quantity of pheromone deposited, which may depend on the quantity and quality of the food, will guide other ants to the food source [18]. This algorithm is a repetitive three phase algorithm including development of the solution, updating process and local research. Different stages of this algorithm are shown in Fig. 4.

At the first step, m ants are distributed uniformly in the search area. The optimization process is started with some random solutions and then each i-th ant would develop a three dimension solution $X_i = (x_1, x_2, x_3)$ for three-dimensional problem by developing three normal distributions and sampling them [18]. The j-th dimension of $N_i(\mu_j, s_j)$ normal distribution has the mean and variance equal to μ_j and s_j, respectively. To form distribution features, each ant needs a guide solution (X_{guide}). For selecting the guide solution, we allocate a P_{ci} probability to each ant. Each ant would select the best overall solution as guide solution with P_{ci} probability and would consider the best solution in its memory as the guide solution with 1-P_{ci} probability. This process is illustrated in Eq. (4) [19]:

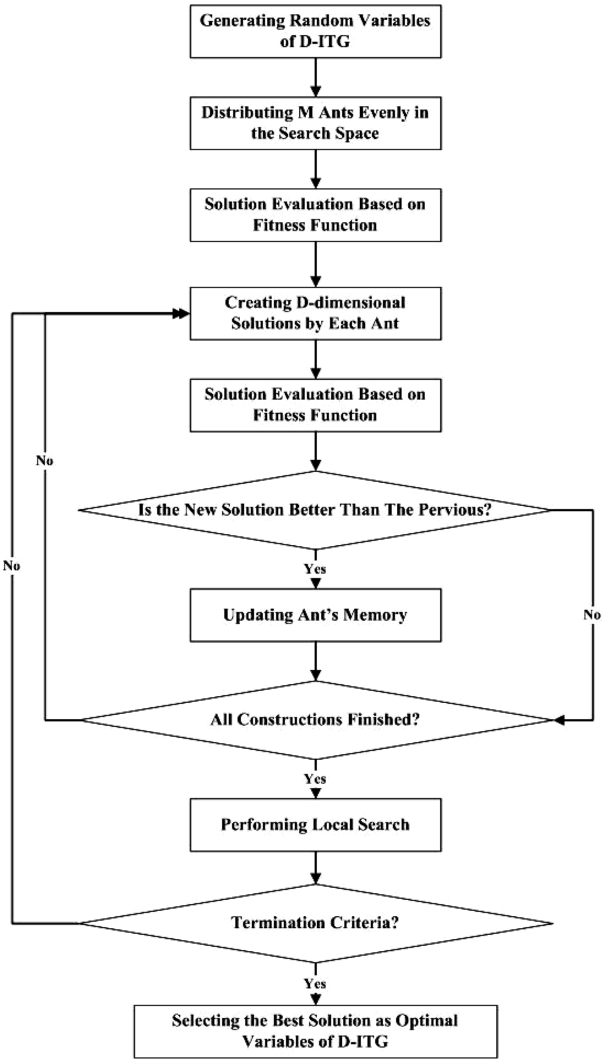

Fig. 4 Flowchart of ACO based proposed algorithm.

$$X_{guide} = \begin{cases} X_{best} & \text{if} \quad \text{rand}(0.1) > P_{c_i} \\ X_i & \text{else} \end{cases} \quad (4)$$

Different P_c values would lead to more variety in ant population and so, better results would be obtained. By selecting different P_c for each ant, different levels of extraction and discovery would be achieved by ants.
P_c probability is calculated by Eq. (5) [19]:

$$P_{c_i} = 0.05 + 0.45 * \left(\frac{\exp\left(\frac{10(i-1)}{P^{s^6} - 1}\right) - 1}{\left(\exp(10) - 1\right)} \right) \quad (5)$$

The elements of guide solution are regarded as normal distribution mean, so we have [19]:

$$\mu_i = \{\mu_i^1, \mu_i^2, \mu_i^3\} = \{X^1_{guide\,i}, X^2_{guide\,i}, X^3_{guide\,i}\} = X_{guide_i} \quad (6)$$

The j-th distribution variance is equal to the guide solution distance from j-th dimension of the best solution of the other ants multiplied by ρ (Eq. (7)) [19]:

$$\sigma_i^j = \rho \sum_{l=1}^{m} \frac{\left| X^j_{guide_i} - X^j_l \right|}{m-1} \quad (7)$$

Positive ρ is equal in all dimensions and has similar effect as the pheromone evaporation rate in ACO. Higher values of this parameter would reduce the convergence rate of the algorithm. Pheromone evaporation rate affects long term memory and causes to forget worse solutions more quickly. At the second step, the fitness value would be calculated for new solutions and for each ant a new solution would be presented. If the presented solution by m-th ant be better than the solution existing in its memory then the ant's memory would be updated using the new solution.

In order to increase the accuracy of the algorithm, a local research strategy is applied to improve the solutions founded by the algorithm. In each repetition, the local research process is recalled for the best ant in the colony. HJ is a pattern search method being presented by Hooke and Jeeves in 1961 and in spite of passing a long time, for many local search problems, this algorithm is applied as the first choice [20]. HJ method selects the solution obtained from overall search as a base point and improves it applying discovery movement and pattern movement. In discovery movement, all base point variables move in predefined steps in turn and if no improvement observe, they would move toward an opposite direction. Then the pattern movement simultaneously repeats all the successful changes in the discovery phase. Then the results obtained by the movements would be evaluated and in case of successful movement; the new point would be regarded as the base point. The process is continuously repeated till there would be no improvement at any dimension. The step size is reduced by passing time and the reduction should be in a way that a new pattern could be developed by that. A great decrease of the step size could result reduction in search process. The final search would be stopped when the step size is small sufficiently [19].

4 Experimental Results

In this section, we present the results of our research. The experimental results are provided applying our three proposed optimization algorithms on real different data sets. These data sets are collected by measuring D-ITG software output with different inputs on ADSL line with 256K speed rate. These algorithms are implemented in C#.Net software and the experiments have been repeated 100 times. In our three-dimensional search space we considered the number of initial chromosomes, countries and ants equal to 50.

In the proposed GA based method the crossover rate and mutation rate are considered 50% and 30%, respectively.

In ICA, we consider 5 initial imperialists and ACO evaporation rate is equal to 0.45 and the primary step is equal to 8, each step would decrease to 1/2.

In fitness function the values of b, α, β and γ variables illustrate the effects of bit rate, delay, jitter and packet loss rate parameters in the fitness function and their experimental values are considered 2, 0.01, 2 and 0.01, respectively.

These values are defined in a way that optimal responses with high bit rate and low delay, jitter and packet loss would be achieved. Moreover, these weights should be proportionate with the effect of each parameter on output value. Thus, since bit rate and jitter could have notable effects on network performance, the weights should be selected in a way that even their negligible change could significantly affect the fitness value.

Fig. 5 illustrates the initial input values in three-dimensional feature space and Fig. 6 shows their corresponding fitness values. The values of fitness function and the changes during different iterations for GA, ICA and ACO are shown in Figs. 7-9, respectively.

In order to indicate the promising ability of the proposed algorithms in determining the optimal input parameters of D-ITG, the best D-ITG input parameters obtained by automatic D-ITG, are compared to manually results (results obtained by D-ITG with manual input parameters). The comparative results are shown in Table 1. As shown in Table 1 our proposed algorithms have indicated their ability in finding the best responses with high fitness values in compare with manual method. Moreover, among three proposed methods the fitness value obtained by ICA based method is greater than the others. On the other hand, we compare the speed of convergence of proposed ICA based algorithm and related GA based and ACO based methods to their desired value of fitness. Results obtained from runs of the algorithms when they are allowed to executed for a maximum of 200 iterations, show that the GA based and ACO based algorithms are able to arrive at the desired fitness value in average 11 iterations, while ICA based technique is able to obtain the desired fitness value, in relatively more number of iteration (average in 14 iterations).

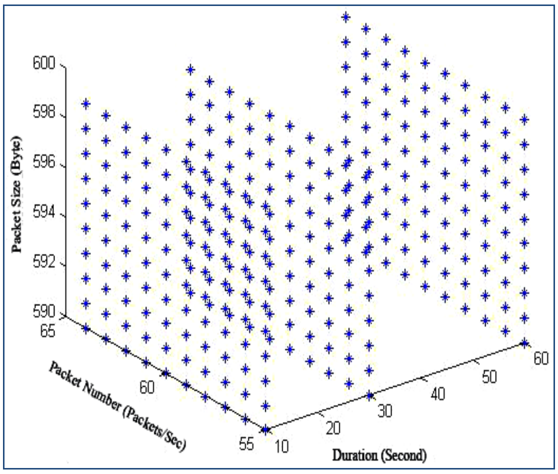

Fig. 5 Simulated data in three dimensional spaces.

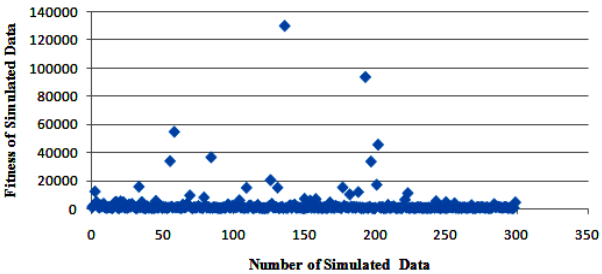

Fig. 6 Simulated data fitness.

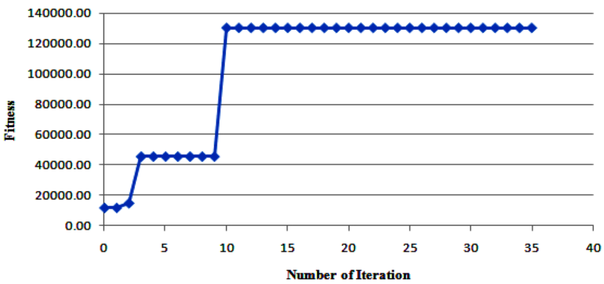

Fig. 7 Convergent of the GA toward the fitness best value.

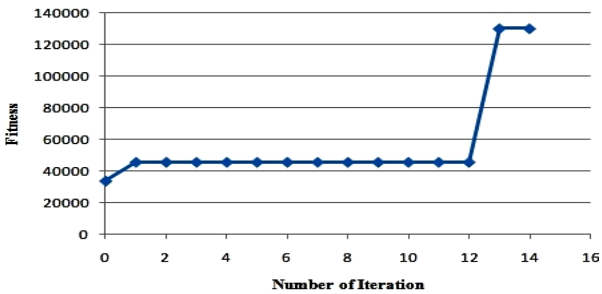

Fig. 8 Convergent of the ICA toward the best fitness.

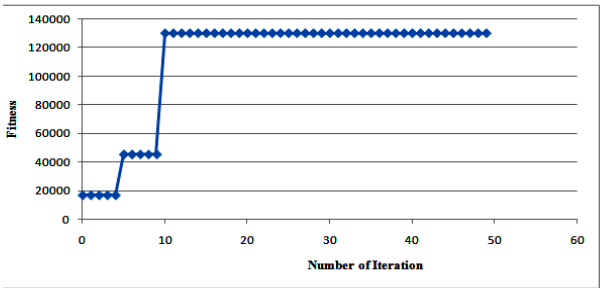

Fig. 9 Convergent of the ACO toward the best value of fitness.

Table 2 illustrates the percentage of improvement of results obtained by three proposed algorithms in compare with the manual method. Finally, experimental results show that the automatic D-ITG is able to find the best input variables and the final solutions of all three proposed algorithms are convergent toward one response.

Table 1 Manual D-ITG and automatic D-ITG compared to GA, ICA and ACO.

Methods	Maximum Fitness	Minimum Fitness	Mean	Variance	Number of Tests	Testing Time [sec.]	Number of Iterations to Achieve Convergence
Manual D-ITG	130312.4	223.2519	3072.959	127239.4	300	10000	-
Automatic D-ITG by GA	130312.4	11811.38	44346.64	85965.81	104	3270	11
Automatic D-ITG by ICA	130312.4	33607.2	60080.65	79539.79	77	3030	14
Automatic D-ITG by ACO	130312.4	17022.08	40299.69	90012.71	54	1780	11

Table 2 Comparing the introduced methods improvements to the manual one.

Methods	Mean to Variance Ratio	Percent Improvement in Number of Tests	Percent Improvement in Testing Time
Automatic D-ITG by GA	0.51	65.34%	67.3%
Automatic D-ITG by ICA	0.75	74.34%	69.7%
Automatic D-ITG by ACO	0.44	82%	82.2%

6 Conclusion

In this paper, GA, ICA and the ACO have been applied in order to obtain the best D-ITG input parameters and to present automatic solution to analyze the network performance parameters. These algorithms are implemented on data derived from 256K ADSL line. By using the proposed methods, it was shown that all three algorithms were able to determine the optimal values of D-ITG input parameters with a high accuracy. Results show the promising ability of these algorithms in automating the D-ITG traffic generating tool. The results indicated that the algorithms would find the optimized variables in predetermined ranges and would improve the experiments time and number of test in compare with the manual method. The results illustrated that the ACO has a higher improvement percentage in relation to the GA and ICA.

Acknowledgment

The authors would like to thank Telecommunication Company of Tehran for their valuable support during the authors' research work.

References

[1] A. Botta, A. Dainotti and A. Pescape, "Multi-Protocol and Multi-Platform Traffic Generation and Measurement", *Proceedings of the 26-th IEEE Conference on Computer Communication (INFOCOM 2007) DEMO Session*, pp. 526-532, 2007.

[2] S. Molnar, P. Megyesi and G. Szabo, "How to Validate Traffic Generators?", *Proceedings of IEEE Conference on Communications Workshops (ICC)*, pp. 1340-1344, 2013.

[3] J. Balen, G. Martinovic and Z. Hocenski, "Network Performance Evaluation of Latest Windows Operating System", *Proceedings of 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1-6, 2012.

[4] S. Debbarma and A. Das, "Empirical Measuring IPv4/IPv6 Network Performance on Microsoft's Windows Operating Systems", *Proceedings of Third International Conference on Advances in Computing and Communications*, pp. 393-395, 2013.

[5] L. Chen, T. Sun, G. Yang, M. Y .Sanadidi and M. Gerla, "End-to-end Asymmetric Link Capacity Estimation", *Proceedings of the 4th IFIP-TC6 International Conference on Networking Technologies*, pp. 780-791, 2005.

[6] W. W. Chan, A. Chen, X. Luo, R. K. P. Mok, W. Li and R. K. C. Chang, "TRIO: Measuring Asymmetric Capacity with Three Minimum Round-trip Times", *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies*, pp. 1-12, 2011.

[7] J. Strauss, D. Katabi and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools", *Proceedings of the 3rd ACM*

SIGCOMM Conference on Internet Measurement, pp. 39-44, 2003.

[8] M. Jain and C. Dovrolis, "End to End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput", *Journal of IEEE/ACM Transactions on Networking (TON)*, Vol. 11, No. 4, pp. 537-549, 2003.

[9] C. Kreibich and N. Weaver, "Netalyzr: Illuminating the Edge Network", *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pp. 246-259, 2010.

[10] A. Tirumala, F. Qin, J. Dugan, J. Ferguson and K. Gibbs, "Throughput Analysis and Measurements in IEEE 802.11 WLANs with TCP and UDP Traffic Flows", *IEEE Transaction on Mobile Computing*, Vol. 7, No. 2, pp. 201-216, 2008.

[11] A. Botta, A. Dainorri and A. Pescape, "A Tools for the Generation of Realistic Network Workload for Emerging Networking Scenarios", *Journal of Computer Networks*, Vol. 56, No. 15, pp. 3531-3547, 2012.

[12] A. Botta, W. Donato and A. Dainorri, "D-ITG 2.8.1 Manual", *Computer for Interaction and Communications (COMICS) Group*, pp. 3-6, 2013.

[13] I. Abuziah and N.Shakarneh, "A Review of Genetic Algorithm Optimization", *International Journal of Physical, Nuclear Science and Engineering*, Vol. 7, No. 12, pp. 66-70, 2013.

[14] S. Karami and S. B. Shokouhi, "Optimal Hierarchical Remote Sensing Image Clustering using Imperialist Competitive Algorithm", *Recent Advances in Computer Science and Information Engineering*, Vol. 124, No. 1, pp. 551-561, 2012.

[15] S. Arish, A. Amiri and K. Noori, "FICA: Fuzzy Imperialist Competitive Algorithm", *Journal of Zhejiang University SCIENCE C*, Vol. 15, No. 5, pp 363-371, 2014.

[16] S. Najafi Ravadanegh, "A Multistage Expansion Planning Method for Optimal Substation Placement", *Iranian Journal of Electrical & Electronic Engineering*, Vol. 10, No. 1, pp. 65-74, 2014.

[17] E. Atashpaz-Gargari, F. Hashemzadeh and C. Lucas, "Evolutionary Design of PID Controller for a MIMO Distillation Column using Colonial Competitive Algorithm", *International Journal of Intelligent Computing and Cybernetics*, Vol. 1, No. 3, pp. 337-355, 2008.

[18] M. R. Mosavi and A. A. Akhyani, "PMU Placement Methods in Power Systems based on Evolutionary Algorithms and GPS Receiver", *Iranian Journal of Electrical & Electronic Engineering*, Vol. 9, No. 2, pp. 76-87, 2013.

[19] M. Abedi and S. Jalil, "An Ant Colony Optimization Algorithm for network Vulnerability Analysis", *Iranian Journal of Electrical & Electronic Engineering*, Vol. 2, No. 3, pp. 106-120, 2006.

[20] R. Hooke and T. A. Jeeves, "Direct Search Solution of Numerical and Statistical Problems", *Journal of the ACM (JACM)*, Vol. 8, No. 2, pp. 212-229, 1961.

Mohammad-Reza Mosavi received his B.Sc., M.Sc., and Ph.D. degrees in Electronic Engineering from Iran University of Science and Technology (IUST), Tehran, Iran in 1997, 1998, and 2004, respectively. He is currently faculty member of Department of Electrical Engineering of IUST as professor. He is the author of more than 210 scientific publications on journals and international conferences. His research interests include circuits and systems design.

Fatemeh Farabi received her B.Sc. degree in Computer Engineering from Iran University of Science and Technology in 2009 and her M.Sc. degree in Computer Engineering from University of Islamic Azad of Qazvin in 2014. His research interests include computer networks and intelligent systems.

Samaneh Karami received her B.Sc. degree in Electrical Engineering from Amirkabir University, Tehran, Iran in 2006 and the M.Sc. degree in Electrical Engineering from Iran University of Science and Technology (IUST), Tehran, Iran in 2009. She is currently Ph.D. student in IUST. Her research interests are signal and image processing.