Digital Signature of Network Segment Using Genetic Algorithm and Ant Colony Optimization Metaheuristics

Paulo R. G. Hernandes Jr.*, Luiz F. Carvalho†, Gilberto Fernandes Jr.†, Mario Lemes Proença Jr.†
*Security Information Department, São Paulo State Technological College (FATEC), Ourinhos, Brazil
†Computer Science Department, State University of Londrina (UEL), Londrina, Brazil
{paulogalego, luizfcarvalhoo, gil.fernandes6}@gmail.com, proenca@uel.br

Abstract—**Every day computer networks have more significance in our lives, and these network's complexity is still growing. To help customers achieve maximum productivity and avoid security risks, network administrators have to manage network resources efficiently. Traffic monitoring is an important task, which describes the network's normal behavior. Thereby, we present a Digital Signature of Network Segment using Flow Analysis (DSNSF) as a mechanism to assist network management and information security through traffic characterization. Our new approach uses a genetic algorithm to optimize the process. In order to accomplish this task, we compared the novel model with another similar method, Ant Colony Optimization for Digital Signature (ACODS), using a real data set of traffic for bits and packets. We also evaluate these models to measure their accuracy.**

Keywords–Traffic Characterization; Traffic Monitoring; Network Management; Genetic Algorithm, sFlow.

I. Introduction

Network Management is a complex task which utilizes different tools and techniques. These tools and techniques aim not only to help network administrators in their daily tasks, but also to provide them with mechanisms which enable them to detect information regarding security events in order to avoid security incidents.

Since the first networked computers, the administrators required all the necessary information about their equipments, so they could understand the behavior of their network by observing information, such as an interface's traffic or which port in a remote switch are being used. Thereby management protocols and tools emerged.

The Simple Network Management Protocol (SNMP) became popular because of its simplicity and its ability to be used by most equipment manufacturers [1]. Using SNMP, we can monitor whether the equipment is functioning, its traffic average and other additional information when required. Nevertheless, with the increase of the complexity of applications that run on networks, such as VoIP, P2P, video on demand, and also the increase of mobile equipment and the Internet of Things, an SNMP protocol alone was not enough for all information required by the network administrators. With the use of data flow, network administrators could have more detailed information to take decisions quicker and more efficiently.

A flow record reports at least the endpoint addresses, time, and volume of information transferred between two sockets. This gives a better view of the traffic than interface-level counters queried by SNMP, and it provides significant data reduction compared to packet traces, allowing it to scale to large networks [2]. The study of flow records can help network administrators identify anomalies in their environments. As a result, researchers are trying to find anomaly detection models based on traffic characterization. These models, as described by Lakhina *et al.* [3], are able to identify an anomalous behavior based on traffic history, learning the standard behavior of an environment, and based on its history detect changes in the network routine.

A network anomaly detection system, first creates a baseline profile of the normal system, network, or program activity. Thereafter, any activity deviating from the baseline is treated as a possible intrusion [4]. It helps administrators to identify any attack or network anomalous behavior, such as users running a P2P application or any other activity which is against company policies.

To reach our target, we use a Genetic Algorithm (GA), a model which simulates the natural evolution process through operators such as selection, crossover and mutation [5]. GA is recognized as an ideal optimization technique to solve large variety of problems. One of the best uses for GA is to optimize search problems, or organize data under some conditions.

Our proposal is to create a Digital Signature of Network Segment using Flow Analysis (DSNSF) utilizing GA to optimize the clustering process and characterize network traffic using flow analysis to permit detection of network anomalies. We use a real set of data to perform the process and evaluate the results to prove the accuracy of our model. Also, we compared this technique with another approach, the Ant Colony Optimization for Digital Signature (ACODS).

This paper is organized as follows: Section II presents the related work. Section III details the novel method DSNSF-GA and also the ACODS approach, both used to characterize network traffic. Section IV delivers the generation of the DSNSF-GA. Section V presents the result of our evaluation tests, and finally Section VI concludes this paper.

II. Related Work

The target of our work is to characterize network traffic and permit network administrators identify anomalous behavior in their environments. For this purpose, we created a DSNSF. This methodology to characterize network traffic was proposed by Proença *et al.* [6] in which a Digital Signature of Network Segment (DSNS) was created using data of each day, usually a workday, based on the history of the previous weeks.

A Firefly Harmonic Clustering Algorithm (FHCA) [7], an optimized clustering algorithm based on the fireflies behavior and its emitted light characteristics, used data acquired using SNMP. To characterize network traffic, certain techniques could be applied such as Holt-Winters for Digital Signature

ISBN: 978-1-61208-376-6

(HWDS), a modification of the classic statistical method of forecasting Holt-Winters [8] or the K-means for Digital Signature (KMDS) [9], where a DSNS is created using K-Means clustering technique. The ACODS approach presented by Carvalho *et al.* [10] is based on Ant Colony Optimization metaheuristic. For DSNSF creation, the ACODS aims to optimize the clustering process, seeking solutions to make it possible to extract patterns of network traffic.

GA was proposed by Holland [5] to simulate the natural evolution process, and it is recognized as an ideal solution to solve problems with a large solution variation. One of the usages of GA is the optimization of the clustering process. A cluster is a group of data which are organized in groups. Data within the same group should be similar and data within other groups should be different. A group is also called cluster. A genetic algorithm-based clustering technique was proposed by [11] and uses the Euclidean distance as the objective function, to classify in which cluster each point should be. This is an example of a profitable way to organize data among clusters using GA and clusterization.

In Xiaopei *et al.* [12], an Artificial Immune System is used along with GA in order to optimize the process. An immune system produces plenty of antibodies to resist an antigen, which is an attribute much similar to the individual diversity of GA. Applying GA to this memory identifying function can enhance the quality of the generated detectors therefore improving the efficiency of detection. In Guo *et al.* [13], a Network Anomaly Intrusion Detection based on Genetic Clustering uses the cluster centers as binary code to organize data and detect intruders. However, if the number of clusters and the length of chromosomes are too large, a system operation inefficiency will be detected.

III. Generation of DSNSF

In this section, we present two metaheuristic strategies to create a DSNSF using data as bits and packets per second. These data were collected using sFlow to generate flows from the network's assets. Our purpose in this work is to demonstrate that flow attributes bits and packets per second can be used to identify a normal, or expected, traffic pattern. The first model is based on the natural evolution of species theory, implemented in computing as Genetic Algorithm, which simulates the natural process of evolution in a population. The second uses the Ant Colony Optimization process, which is based on ant colonies' behavior. Both methods are appropriate to the DSNSF construction and they will be described ahead.

A. DSNSF-GA

Our DSNSF-GA uses the Genetic Algorithm based approach to organize data in clusters. These data were collected using sFlow in State University of Londrina (UEL). We use the average among cluster centroids to generate a graph that will show the network traffic by bits and packets per second using the last three determined days in the week, and compare them with the data of the current day to detect network anomalies. For example, for a Monday we use data from the last three Mondays (excluding the current day) to plot a graph with the characterized network, and compare with these to detect anomaly behavior.

GA is a technique which manipulates a population of potential problem solutions trying to optimize them. Specifically, they operate with a coded representation of solutions which would be equivalent to genetic material (chromosomes) of individuals in nature. Each individual will be assigned a fitness value that will reflect the individual adaptability in an environment in comparison with others. As in nature, the selection will elect the fittest individuals. These will be assigned for a genetic combination, also called crossover, which will result in the exchange of genetic material (reproduction), generating a new population.

Mutation is a value modification in some genes belonging to a solution with some probability p' (the mutation probability). The function of mutation in GA is to restore lost or unexplored genetic material in the population to prevent a premature convergence of a sub-optimal solution, and also to try to find a better solution. Selection, crossover and mutation will repeated for several generations for a fixed number of times, or until some condition is reached. This cycle is represented in Figure 1.

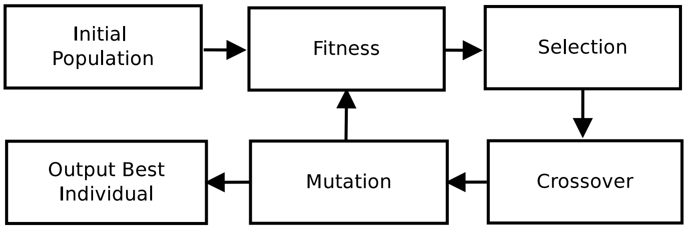

Figure 1. Genetic Algorithm cycle.

To help these mechanisms, there are other operations which must be used to complete the GA process, such as selection engine and crossover point selection. We must also determine our objective and fitness functions, which are the functions being optimized. In our approach we use the Euclidean distance as an objective and fitness function.

The number of clusters is fixed. We use the Euclidean distance to determine the best distance of each point from their respective cluster centers. Our chromosome was represented by a real number, which is the value of the cluster center.

The Euclidean distance is given by

$$J = \sum_{i=1}^{E} \sum_{j=1}^{K} \sqrt{\sum_{n=1}^{N} (x_{in} - c_{jn})^2} \tag{1}$$

in which K is the total number of clusters, E represents the amount of flows to be clustered and N indicates data dimensionality, i.e., number of features to be processed. The collected flows are divided in 5 minute intervals, totaling 288 data sets throughout the day. The variable x_{in} denotes value of the feature n of flow i and c_{jn} stores value of center of cluster j at n dimension.

We chose the Roulette Wheel approach to find the fittest in the population, which conceptually consists of giving each individual a slice of a circular wheel equal in area to the individual's fitness [14]. We spun the roulette for the same number of individuals in the population. Those which were selected more often were the fittest and will breed the new generation.

The crossover operator combines the chromosomes of two parents to create a new one. In our approach, we have chosen a random number to divide both chromosomes in the same point and merge them in another one (child). This process will continue until the old population be replaced by a new population of ”children”.

To start the process, we generate a random initial population in which we began applying the crossover, selection and mutation. As we have described, our chromosomes are the cluster centroids values. We appointed an initial population of forty parents. They create the new generation, which will replace the old one. It will repeat for a number of sixty iterations. For our purpose we conclude that sixty is an ideal number, because a higher number will increase the effort unnecessarily, and a lower one will not create an ideal solution.

At the end of this process, we have the best chromosome based on its fitness function. This value represents a single point in the DSNSF-GA. We have to apply the clusterization using GA for each point in the graphic, so it will be repeated for 288 times, one point every five minutes.

B. ACODS - Ant Colony Optimization for Digital Signature

Nature has been inspiring men in creating solutions for human problems. Hence, a variety of biologically inspired approaches have appeared in various research fields such as engineering, computer science, medicine and economics. For example, the ants' ability to find the shortest path between their colony and the food source has inspired the creation of a very promising method called Ant Colony Optimization (ACO) metaheuristic [15].

Similarly to real ants, ACO agents are able to organize themselves using pheromone trail. They travel through the search space represented by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of all nodes and $\mathcal{E}$ is the edges set. Agents are attracted to more favorable locations to optimize an objective function, i.e., those in which the concentration of pheromone deposited by ants which previously went through the same path is higher.

According to Jiang *et al.* [16], the ant colonies' habits living in groups are essentially similar to the grouping of data. Algorithms based on the behavior of ants have natural advantages in the application of cluster analysis. Thus, we introduce the ACODS, a modification of the Ant Colony Optimization metaheuristic for DSNSF creation using the clustering approach.

In this paper, we assume that the edges of $\mathcal{G}$ are formed between the center of a cluster (centroid) and each element that will be clustered. ACODS runs iteratively and in each iteration, an agent constructs a solution. Although each ant has the ability to build a viable solution as well as a real ant can somehow find a path between the colony and food source, the highest quality solutions are obtained through cooperation between individuals of the whole colony [17].

Algorithm 1 shows the steps executed by ACODS for DSNSF creation. These activities are classified into 3 categories:

Build solutions: This step consists of the movement of ants concurrently and asynchronously by the states of the problem. It is determined by moving agents from one node to another neighbor in the graph structure.

Algorithm 1 – ACODS used for DSNSF creation

```
Require: Set of bits and packets, number of clusters
Ensure: $X$: Vector representing the normal behavior for bits
    and packet sets of a day arranged in 288 intervals of 5
    minute, i.e. the DSNSF
for $i = 1$ to 288 do
    for $t = 1$ to number of iterations do
        Create solution
        Evaluate solutions through the objective function (1)
        Update the pheromone trail
    end for
    Calculate the center of each cluster of the best solution
        found
    $X_i \leftarrow$ Average among the clusters
end for
return $X$
```

Local Search: It aims to test and evaluate solutions created by ants through a local search. In our model, we use the objective function to measure how good are the solutions built by the ants.

Pheromone Update: The pheromone trails are modified in this process. The trails' values can be incremented (when ants deposit pheromones in the edge or connections between the used nodes) or can be decremented. The increased concentration of pheromone is an essential factor in the algorithm implementation, since it directs ants to seek new locations more prone to acquire a near-optimal solution.

IV. Creating a DSNSF-GA

To create our DSNSF-GA, our data were separated in files, one per day. Every file has 86400 lines, which corresponds to the amount of bits and packets per second in each line. As we choose to generate the DSNSF-GA using data from every five minutes, we selected 300 lines to generate a single point in the graphic.

These lines were divided and later grouped in clusters according to Euclidean distance. Using the Silhouette method for interpretation and validation of clusters [18], best results were reached using $K = 3$ and the GA were used to optimize these distribution among the clusters. Each cluster has its centroid, which is the center of its cluster. As we have three centroids, we calculate the average among those three clusters, which in turn should be the point allocated in the middle of these clusters. At this stage we obtained one point in the DSNSF-GA.

The operation of DSNSF-GA is shown in Algorithm 2. The chromosomes are vectors, and they contain the value of the cluster centroids. The length of a chromosome is defined by $N * K$, where N is the number of dimensions of our search space and K is the number of clusters. As we have three clusters, we distribute flow data among those clusters. Each cluster has its centroid and these values will determine a single gene in the chromosome.

For the initial population, we have generated randomly forty chromosomes, and their values should be between the minimum and maximum flow data values. These chromosomes are used to generate new populations of individuals. The next action is to determine the fittest individuals in a population.

Algorithm 2 – using GA to create the DSNSF

```
Require: Set of bits and packets collected from historic
    database, number of: clusters, initial population and iterations
Ensure: $X$: Vector representing the bits and packets of a day
    arranged in 288 intervals, which means 5 minutes, i.e. the DSNSF
for $i = 1$ to 288 do
    Initialize population $\rho$
    for $t = 1$ to number of iterations do
        Compute fitness for $\rho$
        Select the fittest using the Roulette Wheel
        Apply crossover to generate a new population $\tau$
        Apply mutation if necessary
        Evaluate solutions through the objective function (1)
    end for
    Calculate the center of each cluster of the best solution
    found (fittest)
    $X_i \leftarrow$ average among the clusters
end for
return $X$
```

As previously described, we used a Roulette Wheel technique to determine the best chromosomes. What will determine if an individual is or is not fitted, is the shorter distance between all points and their respective cluster centroid. If this distance is lower in an individual than in others, it means the data inside that cluster are well organized, i.e., there are more points closer of its central point in a cluster than in others. In our approach, we used the sum of the three clusters distance to determine the fittest individuals which will procreate.

To yield new generations, individuals must mate between each other. As in nature, the fittest individuals have a greater probability of generating a new offspring, who will generate new ones and so on. When two parents generate a new individual, they will combine their genetic material to a new progeny. This probabilistic process of breeding a new population and combining genetic material is the crossover. This is the key process in a Genetic Algorithm based search, because at this moment the exchange of genes will occur and our population will be diversified. Since two parents reproduce, they will switch genetic material and their children will be a copy of both merged chromosomes. It will assure the population diversity. For our purpose, the exchange of chromosomes will improve the solution, where we are finding the shorter total distance in a chromosome. An important part of the crossover process is to define the crossover point, as we have to choose between a single point crossover or a multi point crossover. A single point crossover is indicated when there is a larger population and we choose that technique to divide our chromosomes. This single point is a random point which divides two chromosomes in four (each of them in two), and combines a couple of those to generate a new one [19].

For a initial population of ρ individuals, we choose $\tau = \rho/2$ corresponding to the fittest in these population to generate a new population. These new population will be mixed to the previous ones and will breed others. The Roulette is span to choose τ, which will generate other children. Each one of these iteration is called generation. Since there is no ideal stopping criteria, usually a fixed iteration number is used. The process of generating new populations will repeat for a fixed number of times which we have set to sixty [20].

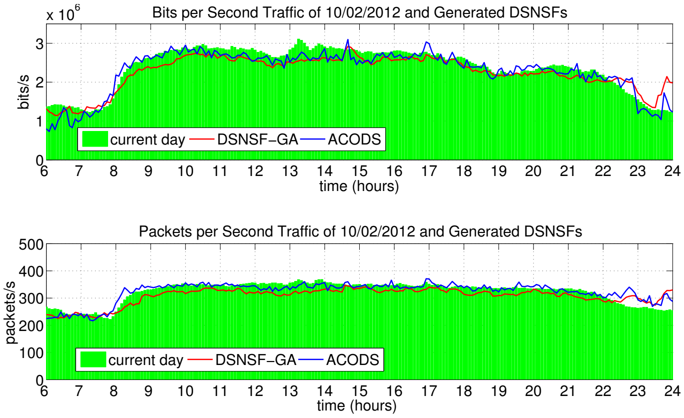

Figure 2. DSNSF-GA and ACODS for 2nd of October

Each chromosome undergoes a mutation probability which is a fixed number. Mutation allows the beginning and the preservation of genetic variation in a population by introducing other genetic structures modifying genes inside the chromosome. We establish a mutation tax of 5%. When the mutation occurs we choose a mutation point, called MP, which will be the point where we are going to change its value. This new value is calculated using $New_i = Old_i + (\delta \times Old_i)$, where New_i is the new individual, Old_i is the old individual, δ is the randomic number $0 < n < 1$ which determine if mutation will or will not occur. The new chromosome will be used to generate a new offspring.

At the end of all these processes we obtained the best population. From this, we choose the best individual, which is the chromosome with the shortest sum of distance between each point in the cluster and its respective centroid. We calculate now the mean among the three cluster centroids. This number will represent a unique point in the DSNSF-GA, as we choose to illustrate a five minute interval in the graphic.

The process of acquiring each value will be repeated for other 288 times to create a graph representing one day in a week. By using data from three days to generate this single point, we now have a network signature of them, or the DSNSF-GA.

V. Tests and Results

We used a real data set for DSNSF creation. These data were collected from State University of Londrina (UEL), and exported using sFlow pattern. Afterwards, these flows are divided in files containing 86400 lines, where each line has the value corresponding to one second in a day. One file is for packets and another one is for bits. As we have two methods to compare, it is important to emphasize that we have set the same number of clusters and iterations for both, ACODS and DSNSF-GA.

In Proença *et al.* [6], a DSNS was created using data of each day based on the history of its previous weeks. This technique was also discussed by Assis *et al.* [21] and Lima *et al.* [22]. Each of them have used a Digital Signature of Network Segment to represent the network behavior efficiently. For our purpose, we used only data between 06:00 and 24:00

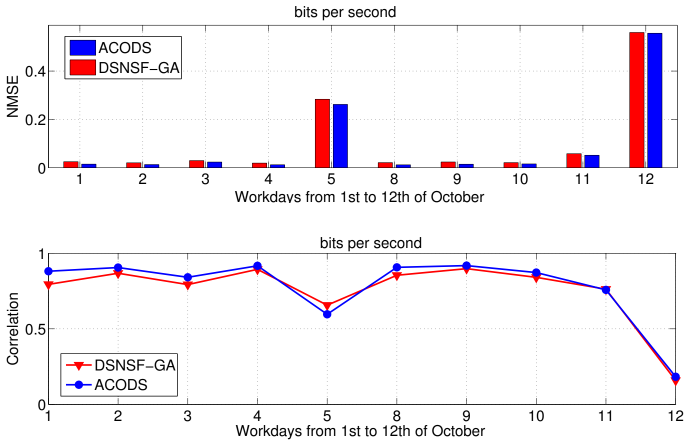

Figure 3. NMSE and CC for bits per second for DSNSF-GA and ACODS

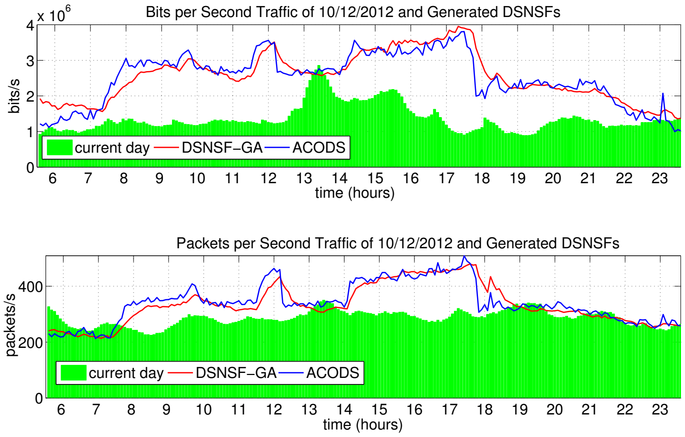

Figure 5. DSNSF-GA and ACODS for 12th of October (national holiday)

since we utilized data from a University and their working hours are between 07:00 and 23:00. It is important to inform that the 12th of October is a national holiday in Brazil, and this is the reason for a different traffic behavior during that day. We decided to keep this day to demonstrate the ability of adaptation of the methods to similar situations.

Figure 2 shows the observed traffic of both, DSNSF-GA and ACODS methods for bits and packets per second. The figure represents the interval described before, October 2nd, 2012, where the green color is the current day, and there are two lines, one for DSNSF-GA and another for ACODS. As shown by this figure, both of them are able to characterize network traffic, displaying the usual observed traffic, the increase and decrease in usage following the same pattern, and also a greater use of network resources during the periods from 07:00 to 12:00 hours and from 13:00 to 23:00 hours.

To measure the accuracy of each method on DSNSFs generation, we decided to adopt two different techniques: Correlation Coefficient (CC) and Normalized Mean Square Error (NMSE).

NMSE compares the mean of a series against certain predicted values. If the NMSE is less than 1, then the forecasts are doing better than the current traffic, but if it is greater than 1, then the predictions are doing worse than the current traffic. The CC measures how suitable a model is, resulting in values varying from -1 to 1. A positive value indicates total correlation, and values close to 0 mean bad or no correlation between data and the adjustable model.

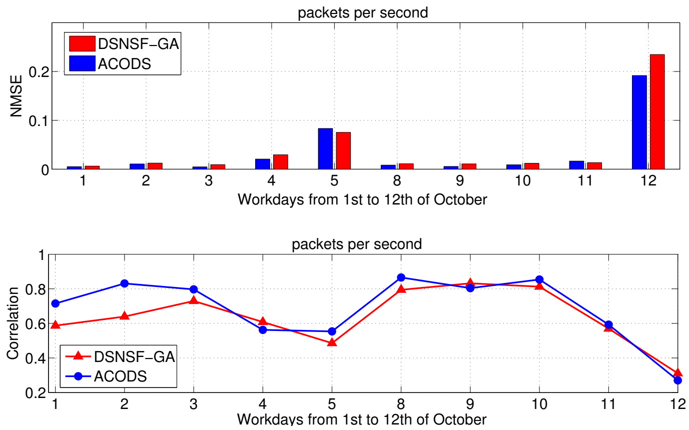

Figure 4. NMSE and CC for packets per second for DSNSF-GA and ACODS

Figure 3 indicates that there is correlation between the observed traffic and the predictions generated by the characterization from DSNSF, as the values are close to 1, both in DSNSF-GA and ACODS. The 12th of October has a bad correlation, and we can see in Figure 5 that the traffic was lower than predicted. This day was a national holiday and there was a little activity at the University. Also in Figure 3, we observe values up to 0.7, meaning that the real traffic and the DSNSF are correlated, except on 12th of October again and on 5th of October.

We investigated what the cause was for the bad correlation 5th of October. The network administrator discovered, analyzing log files, that there was an input HTTP traffic in large scale, caused by students applying for new classes of postgraduate courses. The University was offering 53 new classes that semester, and it was not only the first day but also the only way to apply for a class via the Internet. Figure 4 shows NMSE and CC for packets per second with the same results, indicating that both methods present good error rates, achieving averages NMSE values of 0.4 including the national holiday, and less than 0.1 excluding it. In addition to this, we get an average correlation of 0.75 with the holiday and 0.8 without it.

Both methods are able to characterize network traffic efficiently, as we can see a small difference between the predicted traffic and the real traffic in a normal day. Although we have no threshold to distinguish anomalous from normal behavior, we can see in figures that when good values were observed for CC and NMSE , a normal traffic pattern was also observed. Abnormal values, after investigation, were found to be derived from anomalous traffic.

A. Computational complexity

The methods computational complexity are presented as asymptotic notation based on amount of executed instructions. Initially, the ACODS algorithm partitions a set E of data by

K centers of N dimensions, resulting in $O(EKN)$. Using the population of ants to assist the search of the best centers for the collation of data and, as all the ants are compared with each other in pursuit of the final solution, a quadratic complexity is added, ensuring $O(EKNM^2)$. Taking the number of iterations T into account as stopping criterion of the algorithm, we have a final complexity of $O(EKNM^2T)$. Although a maximum of interactions T is defined, ACODS quickly converges to a solution.

Selection operator is executed by the Roulette Wheel, one of the simplest and most traditional methods. As the choice of the roulette slices is done by a linear search from the beginning of the list, each selection requires $O(\rho)$ because, on average, half of the list will be searched. In general, the roulette method uses $O(\rho^2)$ steps, since an offspring will be bred from the crossover between ρ parents. Crossover and mutation operations present linear behavior of $O(\rho)$. All these processes are executed for the N dimensions (bits and packets). Thereby, the activities number performed by DSNSF-GA during iterations are given by $O(EKNH^2T)$.

VI. Conclusion and future works

In terms of computational complexity, both methods use meta-heuristics algorithms to find an optimal solution.There are a number of iterations until a certain condition is reached, and both of them used the same value.

The DSNSF-GA method, introduced in this paper, uses the Genetic Algorithm technique to improve data clusterization and it characterizes network traffic using data collected by sFlow. To estimate network traffic, we organize data simulating the natural evolution process of the nature. Using natural operators like selection of the fittest, crossover and mutation we can obtain the best individuals in a population. We used the shortest distance among each point in a cluster and their respective centroid to determine who are the fittest, which represents a single point in the DSNSF-GA. These digital network signatures can be used, in the future, by network administrators to identify anomalous traffic in their environments, by comparing the real current traffic with the predicted traffic. As previously described, when we identified a flash crowd traffic caused by new classes of postgraduate students, a large input traffic was associated with the beginning of online applications, and the management of network resources could be done automatically.

In future works, we intend to increase the number of dimensions, including new flow data. This multidimensional approach will improve traffic characterization, as we will use more detailed information about the network behavior. Also, we plan to develop a model to establish a threshold for the DSNSF, to distinguish anomalous from normal behavior, being possible to identify, in real time, network anomalies.

Acknowledgment

This work was supported by SETI/Fundação Araucária and MCT/CNPq for Betelgeuse Project financial support. Also the authors would thanks São Paulo State Technological College (Fatec Ourinhos).

References

[1] W. Stallings, "Snmpv3: A security enhancement for snmp," Communications Surveys Tutorials, IEEE, vol. 1, no. 1, First 1998, pp. 2–17.

[2] B. Trammell and E. Boschi, "An introduction to ip flow information export (ipfix)," IEEE Communications Magazine, vol. 49, no. 4, 2011, pp. 89–95.

[3] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in Internet Measurement Conference, 2004, pp. 201–206.

[4] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Computer Networks, vol. 51, no. 12, 2007, pp. 3448–3470.

[5] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1992.

[6] M. L. Proenca Jr., C. Coppelmans, M. Bottoli, and L. Souza Mendes, "Baseline to help with network management," in e-Business and Telecommunication Networks. Springer Netherlands, 2006, pp. 158–166.

[7] M. Adaniya, M. Lima, J. Rodrigues, T. Abrao, and M. Proenca Jr., "Anomaly detection using dsns and firefly harmonic clustering algorithm," in Communications (ICC), 2012 IEEE International Conference on, June 2012, pp. 1183–1187.

[8] M. V. O. Assis, L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proenca Jr., "Holt-winters statistical forecasting and ACO metaheuristic for traffic characterization," in Proceedings of IEEE International Conference on Communications, ICC 2013, Budapest, Hungary, June 9-13, 2013, 2013, pp. 2524–2528.

[9] G. Fernandes, A. Zacaron, J. Rodrigues, and M. L. Proenca Jr., "Digital signature to help network management using principal component analysis and k-means clustering," in Communications (ICC), 2013 IEEE International Conference on, June 2013, pp. 2519–2523.

[10] L. F. Carvalho, A. M. Zacaron, M. H. A. C. Adaniya, and M. L. Proenca Jr., "Ant colony optimization for creating digital signature of network segments using flow analysis," in 31st International Conference of the Chilean Computer Science Society, SCCC 2012, Valparaíso, Chile, November 12-16, 2012, 2012, pp. 171–180.

[11] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," Pattern Recognition, vol. 33, no. 9, 2000, pp. 1455–1465.

[12] J. Xiaopei, W. Houxiang, H. Ruofei, and L. Juan, "Improved genetic algorithm in intrusion detection model based on artificial immune theory," in Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on, Jan 2009, pp. 1–4.

[13] H. Guo, W. Chen, and F. Zhang, "Research of intrusion detection based on genetic clustering algorithm," in Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, April 2012, pp. 1204–1207.

[14] M. Mitchell, An introduction to genetic algorithms. MIT Press, 1998.

[15] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5, 1999, pp. 137–172.

[16] H. Jiang, Q. Yu, and Y. Gong, "An improved ant colony clustering algorithm," in Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, vol. 6, oct. 2010, pp. 2368 –2372.

[17] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," Computational Intelligence Magazine, IEEE, vol. 1, no. 4, nov. 2006, pp. 28 –39.

[18] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," Journal of Computational and Applied Mathematics, vol. 20, no. 0, 1987, pp. 53 – 65.

[19] W. M. Spears and V. Anand, "A study of crossover operators in genetic programming," in ISMIS, 1991, pp. 409–418.

[20] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms," Pattern Recognition Letters, vol. 17, no. 8, 1996, pp. 825–832.

[21] M. V. d. Assis, J. J. Rodrigues, and M. L. Proenca Jr., "A seven-dimensional flow analysis to help autonomous network management," Information Sciences, vol. 278, no. 0, 2014, pp. 900 – 913.

[22] M. Lima, L. Sampaio, B. Zarpelão, J. Rodrigues, T. Abrao, and M. L. Proenca Jr., "Networking anomaly detection using dsns and particle swarm optimization with re-clustering," in Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, Dec 2010, pp. 1–6.