Genetic Algorithm for Optimizing and Arrangement of Queue in Virtual Networks

Heru Sukoco, and Koji Okamura

Abstract—**Network virtualization provides methods that simplify resource management, deal with connectivity and bandwidth constraints, and enable virtual network-connected machines through either a common layer 2 or layer 3 IP network of TCP/IP protocol suite. This paper presents an alternative optimization solution in maintaining the packets queue on a virtual router by using Genetic Algorithm. The objective of this research is to reduce a cost of network resources such as memory and time processes on a router. It is important for a company with limited cost when implements a network infrastructure.**

We defined a crossover probability and a mutation probability to 0.90 and 0.05. In our experiments, we still have unsatisfied results due to the average and the standard deviation of fitness values and slots needed in queues are 58,238.10 & 139,575.45 and 106.20 & 82.51, respectively. These values should be better by defining an appropriate fitness function in our next experiments.

We still continue the research by examining an appropriate fitness function, Genetic Algorithm as a classifier, dynamic slot size, and also comparing with current queue managements such as first-in first-out and weighted fair queue.

Index Terms—**network virtualization, resource management, queuing management, genetic algorithm, crossover, mutation.**

I. Introduction

In this paper, we present a genetic algorithm used in network virtualization (NV) technology as one of the solutions to achieve optimization in queuing management and packet selection. The paper contains six sections organized as follows: section Introduction introduces the research background, motivation, related works, problem statements and contribution.

We introduce the description and function of the genetic algorithm and afterwards describe the case study and its results in section Case Study and section Result and Discussion, respectively. Finally, some conclusions and the next works are drawn in section Conclusion and Future Works.

A. Background

Nowadays, we are living in the midst of information technology (IT) revolution involving the generation, processing, and transmission of information. A particular organization may run a network comprising several network technologies such as Ethernets, Token Rings, Asynchronous Transfer Mode (ATM), and the Internet.

The internet is being the most popular network technology in the world, grows exponentially, and widely used by users around the world in their daily activities [1]. Any organization possessing and connected to such networks would like to be able to rely on orderly deployment, use, and upgrade of the networking equipments, both hardware and software.

Since its born, it becomes more complex networks with very huge members serving many applications. The complexity of the Internet could be a problem for network providers when adopting a new architecture. Due to its multi-provider nature on the Internet technologies, implementing a new technology or modification of the existing infrastructure requires consensus among them. Alteration to the Internet architecture and deployment new network technology has become restricted and more difficult [2], [3].

NV has been proposed as a diversifying attribute of the Future Internet (FI) [4] and focused on developing new services to increase the virtual networks more effective [5]–[7]. Therefore, providing quality of service (QoS) guarantees such as bandwidth allocation and packet loss in virtual networks are imperative [5].

B. Motivation

NV has been popular not only in implementing new technology over the Internet but also in research topic that many researchers around the world are interested with it [8]. An appropriate queueing technique is needed to manage virtual network resources effectively. In [5], [9], [10] traditional scheduling algorithms such as weighted fair queuing (WFQ) and recursive round robin (RRR) are not capable of providing service in a virtual network due to traffics flow in aggregated and tunneled sessions, which cannot isolate the traffic of well behaved from misbehaved sessions.

Genetic algorithm (GA) has been applied to solve many problems in variety fields, including computer networks. It works best in a wide range of low to high epistasis [11]. GA offers an effective and optimal technique to scheduling problems, route selection, and queuing management. It is a stochastic process and uses a learning based (heuristic method) loosely on principles of biological evolution [12]–[14].

C. Related Works

NV is a technique for isolating computational and network resources through virtualization to allocate them to a logical network for accommodating multiple independent and programmable virtual networks [8]. It comprises a collection of virtual nodes and virtual links overlaid on a physical network

Manuscript received December 28, 2010; revised February 5, 2011. This work was supported in part by the Ph.D Award Scholarship of the Directorate General of Higher Education (DGHE), Department of National Education, Indonesia under Grant No.1724.2/D4.4/2008.

H. Sukoco is with the Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka-shi, 819-0395 JAPAN e-mail: heru@itslab.csce.kyushu-u.ac.jp.

H. Sukoco is also with the Department of Computer Science, Faculty of Mathematics and Natural Sciences, Bogor Agricultural University, Bogor, 16680 INDONESIA e-mail: hsrkom@ipb.ac.id.

K. Okamura is with the Department of Advanced Information Technology, Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka-shi, 812-8581 JAPAN e-mail: oka@ec.kyushu-u.ac.jp.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

infrastructure [4], [5] which all of arriving packets will be forwarded. The research related to QoS in managing network virtual resources has been limited.

[15], [16] have been proposed the concept of the link sharing and packet scheduling algorithm on a physical link. The methods enable a physical link into several logical links to transmit traffics with different priority. GA has been used as a technique for constrained optimization and it can be improved with enclosing information about the scheduling problem represented by the use of heuristics. Scheduling problems introduced by [17]–[19] showed that the GA produced much better results in terms of quality of solutions in a multiprocessors and parallel system. The population size and the number of generations depend on the number of tasks.

The scheduling problems in virtual networks arrived when the packets received from many nodes in aggregated or tunneled packets. The scheduling of integrated best effort and real-time traffics in a virtual network and the complex real-time scheduling based on Genetic Algorithms have been discussed in [5] and [14], respectively.

D. Problem Statement

Today many providers use NV technologies to support their business and deploy their services to the customers. If the providers could not guarantee QoS, they may miss critical customers who could result in lost revenue, service level agreement (SLA) penalties, lost business, and further damage to the quality reputation [1].

A router which is specialized in forwarding and routing packets from source to destination has a particular queue length and capacity. These packets have varying packet sizes according to the protocol used by them. We want all of packets on virtual networks could reach the destination host by using network virtualization technologies. Due to aggregated and tunneled traffics come from virtual nodes to a router, typical queuing disciplines such as first-in first out (FIFO), weighted fair queuing (WFQ), and weighted round robin (WRR) have some problems [5], [9], [10].

We will use GA in maintaining packets arrived into a router to help us manage and improve our virtual network resources.

E. Contribution

Maintaining network resources on virtual network infrastructure are terribly important. Our research proposed a case study of managing queues on virtual networks based on GA. In addition, we also defined a WFQ-like model in implementing GA-based queuing discipline.

Finally, we evaluated our GA-based queuing model and performance to get an optimal arrangement of packets in slots of the queue and a minimal number of queue length on typical router technology. It is crucial to use minimal queues as possible on a router to reduce costs in using network resources such as memory and process time.

II. Genetic Algorithm

The usual form of GAs was described by Goldberg [20]. GA is a method for solving both constrained and unconstrained optimization problems that is based on a mechanism of natural selection on natural genetics. Among the evolutionary techniques, GAs are the most extended group of methods representing the application of evolutionary tools that rely on the use of a selection, crossover, mutation operators, and replacement by generations of new individuals [21].

GA is one of the heuristic methods based on the fittest individual surviving in a population. It is a tremendous tool for solving a variety of optimization problems that are not well suited for standard optimization algorithms. Figure 1 below outlines a tree structure of natural techniques among other well-known search procedures [11], [21].

The GA uses three main types of operations, i.e. selection, crossover, and mutation at each step to create the next generation from the current population. The flowchart of GA procedure is shown in Figure II while the following terminologies describing the GA's concepts are shown below:

1) **Population** is a set of chromosomes (individuals).
2) **Chromosome** expresses a possible solution to the problem as a string (a set of bits).
3) **Fitness function** results a fitness value from a chromosome as an input and returns a higher value for better solutions.
4) **Selection method** determines how the individuals, called parents, are selected for breeding from the initial population that contribute to the new population at the next generation.
5) **Crossover operation** determines how parents combine to produce offspring (children) for the next generation.
6) **Mutation operation** apply random changes to individual parents to form children.

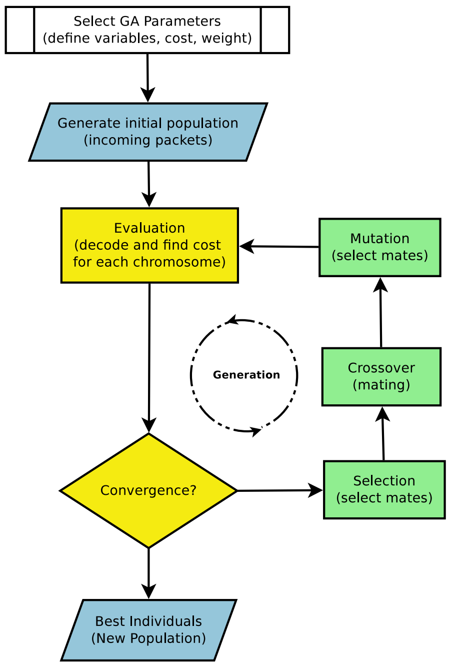

Fig. 2. Flowchart of Generation Algorithm

Note: GA represents solutions as binary strings (called chromosomes) and creates a random population of candidate solutions. Afterwards, it selects two parents from the population and crosses them to create two offsprings and sometimes mutate the children. The process will be repeated for the next selections, crossover, and mutation, until GA reaches a solution based on a desired level of fitness.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

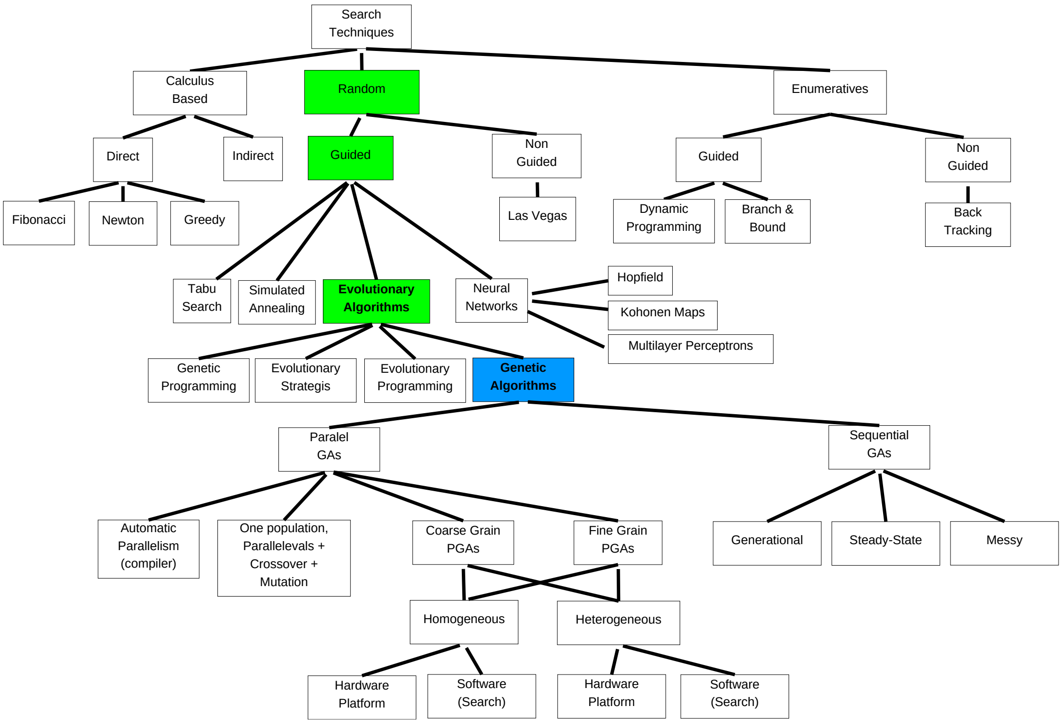

Fig. 1. The tree structure of well-known search procedures and techniques [21]

The GA commonly starts with an *initial population*, i.e. an initial set of random solutions, which are represented by chromosomes and repeatedly modifies a population of individual solutions. The chromosomes evolve through successive iterations, called *generations*. During each generation, the solutions represented by the chromosomes of the population are evaluated using some measures of fitness. At each step, the GA selects individuals at random from the current population to be parents and uses them to produce the offspring (children) for the next generation towards an optimal solution as a better new population.

III. Case Study

We are interested in network virtualization technology used in Future Internet, the replacement of current Internet technology. A research on implementation of GA as one of the alternative algorithms for managing virtual network resources is a challenging issue. We are proposing a new queue management for optimizing and managing packets arrived on a router queue. Our research uses a simulation to examine our model.

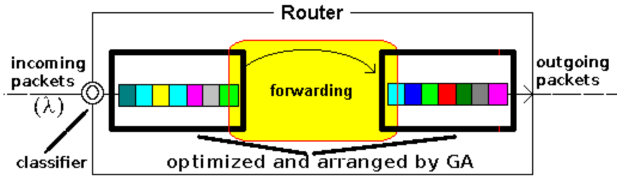

Fig. 3. Forwarding packets and queue management in a router

We use common queue models as well as Cisco's queue model and Java Genetic Algorithm Project as shown in Figure 3 and mentioned in [22], respectively. Java Genetic Algorithm Package (JGAP) developed by JGAP Project is a Genetic Algorithms and Genetic Programming component provided as a Java framework. It is also important to know some terms of the genetic algorithm compared with other classical optimization algorithms as shown in Table I.

TABLE I
Comparison of Natural Evolution and Genetic Algorithm Terminology [21]

Natural Evolution	Genetic algorithm
Chromosome	String
Gene	Character (or Feature)
Allele	Feature value
Locus	String position
Genotype	Structure or coded string
Phenotype	Parameter set, a decoded structure

A virtual network has virtual nodes overlaid on the physical nodes. We illustrated the virtual network and its queue model as shown in Figure 4. Packets transmitted by any sources will be classified and processed by GA. It defines and arranges the packets to get an optimal condition based on fitness value. Fitness value is needed to decide when a router has an optimal condition, called a solution.

We use the GA's pseudo-code, as shown in Algorithm 1, to impelement our case study. The algorithm consists of initialization, iterative variation and selection. Over iterations of random variation and selection, the population can be made to converge to optimal solutions by examining fitness values. The effectiveness of GA depends on the variation and selection operators as applied to a chosen representation and initialization.

A. Genetic Information

Genetic information of an individual is stored in a chromosome. The chromosome represents a potential solution and is divided into multiple genes. It consists of sequences of

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

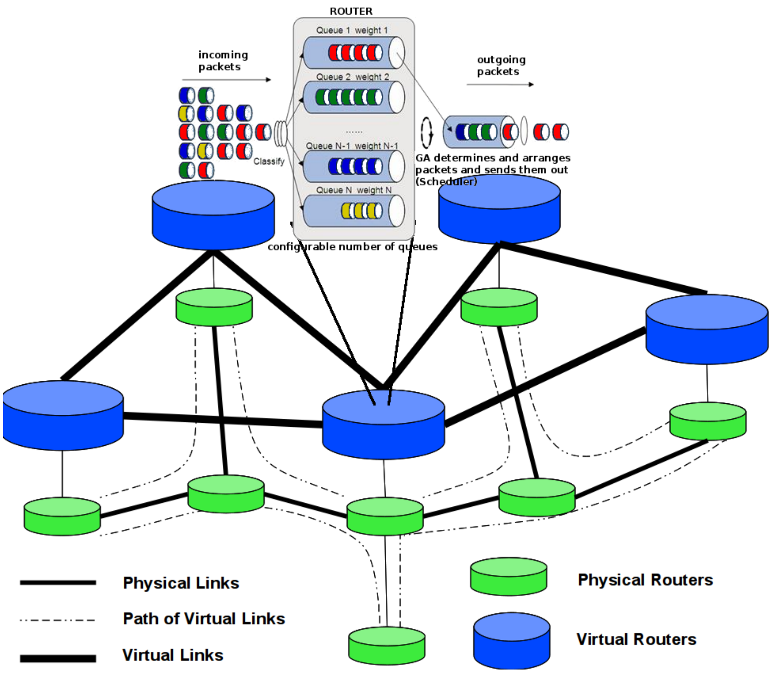

Fig. 4. Queue model in a virtual network

Algorithm 1 A Pseudo-code for Optimizing Scheduling using Genetic Algorithm

Require: $t \leftarrow 0$; $initialize\ P(0)$; $evaluate\ P(0)$
{**comment:** Get a new better population}
while $N \neq 0$ (not convergence) **do**
  {Get a generation}
  $t \leftarrow X \mid 1$
  $select\ P(t)\ from\ P(t-1)$
  $alter\ P(t)$
  $evaluate\ P(t)$
end while

positive integers that represent the packet IDs transmitted to a router. Each locus of the chromosome represents a packet sequence in a router. Locus is a unique position on a set of all the genes of a specific individual.

In our case study, chromosomes are exposed to multiple genetic operators that represent mating and mutation. Afterwards, they are chosen for the next generation during a natural selection phase based upon their fitness. Given a list of n slots and m packets, chromosomes are ordered list of pairs of slots (s_i) and packets (p_j) defined as (s_1,p_1),(s_2,p_2),(s_3,p_3),..,(s_{n-1},p_{m-1}),(s_n,p_m) where $1 \leq s_i \leq n$ and $1 \leq p_j \leq m$. Slots and packets could be vary in size during their transmission and process in a router. A slot can transmit at least one packet at a time. In our illustration as shown in Table II, a slot can process 3 packets at once as long as the slot capacity is available.

B. Initialization

We initialize the population by generating a random procedure in an appropriate range of values. These values are chosen based on some experimentation and can vary based on the specific problem. In our case study, we use a population size of 250 and a total number of 5000 evolutions. Thus, the population initially consists of 250 chromosomes as potential solutions.

C. Fitness Function

The fitness function interprets a measure of how optimal that solution is relative to other potential solutions. The objective of our case is to produce superior solutions by examining the fitness function. The fitness function is the most important part in GA to determine which populations potentially make into the next evolution. The fitness function for our case study can be shown in Equation 1.

$$\mathbf{F_i} = \left(\sum_{i=1}^{n} S_i \right) \times N_s \tag{1}$$

where:
F_i = fitness value of the i-th chromosome
S_i = available capacity of the i-th slot
N_s = the number of slot needed by a router

Our solution is to reduce processing and transmitting packets by regulating the use of slot numbers in queues as small as possible. GA will evaluate the packets and put them in the slots. A solution can be reached after GA has the

TABLE II
EXAMPLE OF CHROMOSOMES

Packets	Slots
1	1
2	1
3	2
4	2
5	2
6	3
7	4
8	4

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

fewest number of slots as shown in Equation 2.

$$\textbf{Solution} = \min(F_i) \quad (2)$$

IV. Result and Discussion

GA will process the population repeatedly until the convergent value is reached. Our research used a simulation method that was performed with Java's technology $version1.6.0_20, OpenJDK64 - BitServerVM19.0 - b09$ on Intel Core i5 processor and running on Linux Ubuntu 10.10 with kernel 2.6.35-24. In all the simulations, the crossover probability and mutation probability are set to 0.90 and 0.05, respectively. Those are typical option values used in GA.

Each experiment is terminated when all the chromosomes have converged to the same value (solution). We determine the termination criteria defined by the number of evolutions and the loop will end when the highest number of evolutions is reached or when the optimal number of slots needed is reached. We evaluated our problems by running simulations with these criteria and results, as shown in Table III.

TABLE III
The results of simulation for arrangement and optimizing queue

	Ev	Pop	Pckt	SS	FV	NS
	2,250	50	125	4	306.199	49
	3,750	50	150	6	4,569.923	94
	1,750	100	150	4	503.058	58
	750	100	150	4	430.430	47
	4,750	100	300	6	1,188.395	94
	9,750	100	300	10	135,123.753	201
	$\geq$ 10,000	300	300	4	436,992.673	298
	$\geq$ 10,000	300	300	6	656.789	79
	5	300	300	15	575.915	
	2750	300	1000	15	1953.040	109
AVG	4,575.50	170.00	307.50	6.30	58,238.10	106.20
SD	3,924.08	113.53	255.51	3.59	139,575.45	82.51

Note:
Ev = Evolution
Pop = Population
Pckt = Number of Packets
SS = Slot Size
FV = Fitness value
NS = Number of slots

To yield a solution in the simulation, we define initial values to the parameters. We set initial values of the evolution value, the population size, packet size, and slot size to 0-12000, 50-300, 125-1000, and 4-15, respectively.

Our experiment results still show unstable values in the evolution and fitness values that have the standard deviation 3,924.08 and 139,575.45 respectively. Those values indicate high variations in the evolution and fitness values to get a solution. These problems are subjects to discuss for getting a better fitness function.

V. Conclusion and Future Works

This paper presented a genetic algorithm for evaluating and solving queue management to reduce cost of network resources especially on a router such as memory and time processes. Our results showed that the fitness function should be precisely well-defined for getting better result in performing queue management.

By defining a crossover probability and mutation probability to 0.90 and 0.05, our experiments result the average (AVG) and standard deviation (SD) of fitness values and slots needed are 58,238.10 & 139,575.45 and 106.20 & 82.51, respectively. These values are subject to define in the future works by defining a better fitness function.

In the future, we will continue the research by evaluating the fitness function and implementing dynamic slot size to get complexity and performance of a router in a real implementation. We also would compare our results with current queue managements implementing on a router such as FIFO/FCFS and WFQ. It is important to compare with other queue management on a router to achieve results of proposing an alternative queue management on a router. Finally, we will use GA as a classifier for arrived packets on a router to identify the priority of process.

Acknowledgment

The authors would like to thank the reviewers for constructive comments and suggestions, the members of Okamura Laboratory, Department of Advanced Information Technology, Kyushu University, Japan and the members of NCC Laboratory, Department of Computer Science, Bogor Agricultural University (IPB), Indonesia for valuable help and support in discussion.

References

[1] Sukoco, H., Y. Hori, Hendrawan, and K. Sakurai, *Towards a Fairness Multimedia Transmission Using Layered-Based Multicast Protocol*, IEICE TRANSACTIONS on Information and Systems Vol.E93-D No.11, pp.2953-2961, Special Section on Architectures, Protocols, and Applications for the Future Internet, ISSN: 1745-1361, 0916-8532, Nov. 2010.

[2] Turner, J.S. and D.E. Tylor, *Diversifying the Internet*, In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'05), Vol.2, DOI:10.1109/GLOCOM.2005.1577741, Dec. 2005.

[3] Anderson, T., L. Peterson, S. Shenker, and J. Turner, *Overcoming the Internet impasse through virtualization*, IEEE on Computer, Vol.38 No.4, pp.34-41, DOI 10.1109/MC.2005.136, ISSN: 0018-9162, April 2005.

[4] Chowdhury, N.M.M.K and R. Boutaba, *A Survey of Network Virtualization*, Computer Networks, Elsevier, Vol.54 pp.862-876, DOI:10.1016/j.comnet.2009.10.017, 2010.

[5] Garg, R. and H. Saran, *Scheduling Algorithms for Bounded Delay Service in Virtual Networks*, In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'99), Vol.2, pp.1318-1322, DOI:10.1109/GLOCOM.1999.829986, ISBN: 0-7803-5796-5, August 2002.

[6] Vicente, J., *Genesis: Toward Programmable Virtual Networking*, In Proceedings of OPENSIG'98 Workshop on Open Signaling for ATM, Internet and Mobile Networks, DOI:10.1.1.70.8838, October 1998.

[7] Florissi, D., *The NetScript Approach to Programmable Networks*, In Proceedings of OPENSIG'98 Workshop on Open Signaling for ATM, Internet and Mobile Networks, October 1998.

[8] Nakao, A. *Network Virtualization as Foundation for Enabling New Network Architectures and Applications*, Invited Letter on Special Edition on New Generation Network towards Innovative Future Society, IEICE Transactions on Communication, Vol.E93-B No.3, pp.454-457, DOI:10.1587/transcom.E93.B.454, March 2010.

[9] Demers, A., S. Keshav, and S. Shenker, *Analysis and Simulation of a Fair Queueing Algorithm*, In Proceedings of ACM Symposium on Communications Architectures and Protocols (SIGCOMM'89), pp.3-12, DOI:10.1145/75246.75248, ISBN:0-89791-332-9, October 1990.

[10] Garg, R. and X. Chen, *RRR: Recursive Round Robin Scheduler*, In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'98), Vol.1, pp. 422-432, DOI: 10.1109/GLOCOM.1998.775767, ISBN: 0-7803-4984-9, 1998.

[11] Haupt, R.L. and S.E. Haupt, *Practical Genetic Algorithm*, 2nd Ed. ISBN 0-471-45565-2, John Wiley & Sons, Inc, 2004.

[12] Sheta, A., M. Salamah, H. Turabieh, and H. Heyasat, *Selection of Appropriate Traffic of a Computer Network with Fixed Topology Using GAs*, IAENG International Journal of Computer Science (IJCS), Vol.34 Issue:1, ISSN: 1819-9224 and 1819-656X, August 2007.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[13] Ahn, C. W. and R.S. Ramakrishna, *A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations*, IEEE Transactions on Evolutionary Computation, Vol.6 No.6, pp.566-579, DOI:10.1109/TEVC.202.804323, December 2002.

[14] Montana, D., M. Brinn, S. Moore, and G. Bidwell, *Genetic algorithms for Complex, Real-Time Scheduling*, In Proceedings of IEEE on Systems, Man, and Cybernetics, Vol.3 pp.2213-2218, DOI:10.1109/ICSMC.1998.724984, ISBN: 0-7803-4778-1, October 1998.

[15] Floyd, S. and V. Jacobson, *Link Sharing and Resource Management Models for Packet Networks*, IEEE/ACM Transactions on Networking, Vol.3 No.4, August 1995.

[16] Zhang, H. and J.C.R. Bennett, *Hierarchical Packet Fair Queueing Algorithms*, In Proceedings of ACM SIGCOMM'97, pp.143-156, September 1997.

[17] Ahmad, I., Y.K. Kwok, and M.Y. Wu, *Analysis, Evaluation, and Comparison of Algorithms for Scheduling Task Graphs on Parallel Processors*, In Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks, pp.207, ISBN: 0-8186-7460-1, IEEE Computer Society Washington, DC, USA, 1996.

[18] Kwok, Y.K., and I. Ahmad, *Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors*, ACM Computing Surveys, Vol. 31, Issue. 4, December 1999, ISSN: 0360-0300, ACM New York, NY, USA, 1999.

[19] Kaur, K., A. Chhabra, and G. Singh, *Heuristics Based Genetic Algorithm for Scheduling Static Tasks in Homogeneous Parallel System*, the International Journal of Computer Science and Security (IJCSS), Vol.4, Issue 2, pp.183 - 198, May 2010. ISSN (Online): 1985-1553, CSC Journals, 2010.

[20] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, ISBN-10: 0201157675, ISBN-13: 9780201157673, Addison-Wesley, 1989.

[21] Sivanandam, S.N. and S.N. Deepa, *Introduction to Genetic Algorithms*, Chap.2 pp.15-27, ISBN 978-3-540-73189-4, Springer-Verlag, 2008.

[22] JGAP Project, *Java Genetic Algorithms Package*, Available at http://jgap.sourceforge.net/.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)