Efficient Scheduling Algorithms on Bandwidth Reservation Service of Internet using Metaheuristics

Tomoyuki Hiroyasu
Department of Life and Medical Sciences
Doshisha University
Kyoto, Japan
tomo@is.doshisha.ac.jp

Kozo Kawasaki
Graduate School of Engineering
Doshisha University
Kyoto, Japan
kkawasaki@mikilab.doshisha.ac.jp

Michihiro Koibuchi
Graduate School of Engineering
National Institute of Informatics
Tokyo, Japan
koibuchi@nii.ac.jp

Shigeo Urushidani
Graduate School of Engineering
National Institute of Informatics
Tokyo, Japa
urushi@nii.ac.jp

Mitsunori Miki
Department of Science and Engineering
Doshisha University
Kyoto, Japan
mmiki@mail.doshisha.ac.jp

Masato Yoshimi
Department of Science and Engineering
Doshisha University
Kyoto, Japan
myoshimi@mail.doshisha.ac.jp

***Abstract*—Network services that dynamically allocate bandwidth resources, such as QoS and layer-1 bandwidth-on-demand (BoD), are increasingly required to advanced Internet backbones, such as science information networks (SINET) in Japan. In this paper, we propose scheduling algorithms for BoD service which allocate parts of full bandwidth dedicated to specific users according to their requests in advanced Internet backbones. The scheduling algorithms maximize the number of accepted requests, fairness, or the total bandwidth in BoD. Simulation results show that the proposed algorithms achieve high utilization of network resources and user's fairness, compared with a simple random-based algorithm.**

***Keywords*-Bandwidth-on-Demand service, SINET3, scheduling algorithm, Internet backbone**

I. Introduction

Quality of Service (QoS) is essential for satisfying various types of Internet-application requirements by making the best use of the limited bandwidth. In particular, layer-1 bandwidth-on-demand (BoD) service, that dynamically allocates a certain amount of bandwidth to specific users according to their requests, is received attention as the ideal QoS service in Internet backbone, and it has been employed in science information networks (SINET) in Japan[1]. SINET3 is the Japanese academic backbone network for more than 700 universities and research institutions witch emphasizes some service aspects: VPN, QOS, and bandwidth on demand. In this service, when the user requires a certain amount of bandwidth between an arbitrary pair of source and destination nodes, the part of full bandwidth is temporally assigned to the user, just like dedicated links in traditional telecommunication. The scheduler is thus required to efficiently and fairly allocate network resources, and it strongly affects user satisfaction. However, at present, a simple algorithm such as first-come first-serve (FCFS) policy has been employed[2], which causes imbalanced utilization of network resources and unfairness. There has been active research regarding the efficient operation of the subject network resources in packet communication on large-scale networks[5][6]. However, There are few research focus of Layer-1 BoD. In this paper, we propose sophisticated scheduling algorithms, that decide whether each request should be accepted or not, in order to make the best use of networks resources in Internet. The priority of user requests is optimized to each objective, such as fairness, or maximum total bandwidth used.

II. Layer-1 BoD Service in Internet

The layer-1 BoD services are dynamic layer-1 resource allocation services dynamically triggered by users and artfully on advanced Internet backbones by using a BoD server. Some applications, such as non-compressed high-definition videos, need huge fixed end-to-end bandwidths to transfer constantly flowing data.

To accommodate the Layer-1 BoD service, the network has to flexibly assign network resources to it in response to user requests. Its users utilize this on-demand capability to selectively connect to several sites and specify the necessary bandwidth.

A. Layer-1 BoD Services and Server Functions

For ease of understanding, we explain an example of Layer-1 BoD services and server's functions implemented in SINET3[2]. Layer-1 BoD services rely on a layer-1 BoD server through which users can directly request a layer-1 path setup via their web browsers. The user calls for the service as request which is specified by some attributes : source node, destination node, requested start time, re-

quested duration, bandwidth requirement, and route option "minimum delay" or "unspecified" by web interface.

After receiving the requests, at scheduling intervals the BoD server calculates the appropriate path routes, performs admission control, and schedules the accepted layer-1 paths to be set up. If some requests non-scheduled fail to admission because of capacity of links, BoD server examines rejection requests from non-scheduled requests set we call "Queue".

The BoD server directs the source layer-1 switches to establish layer-1 paths toward the destination layer-1 switches at timing the requested start time.

III. Scheduling algorithm for Layer-1 BoD

A. Scheduling requirement

Layer-1 BoD is a system that assumes an unspecified number of users. Therefore, when multiple users request the bandwidth for the same usage link between communication bases, for the same date and time, not all users may have the network reserved as requested. Under such conditions, a request scheduling method that determines which users' requests are accepted or rejected becomes important. The specific requirements are described below.

- Maintain the overall throughput at a high value
  By maintaining a constant number of request acceptance trials as well as the rate of usage of each link, high service throughput should be maintained. To do this, it is very important to make the correct decision in choosing which user requests to accept, and how to provide the network resource to that accepted request.
- All users use the network resources equally
  In Layer-1 BoD, multiple users share the bandwidth from the link, which is a part of the network resource. Therefore, we should avoid the situation where the resource is constantly occupied by a particular user making the link unavailable to other users.

Considering the above two points, this paper presents a scheduling algorithm that aims to use links efficiently and to maintain a high level of satisfaction for each user.

B. Fitness Function for Evaluating the Schedule

When considering the scheduling algorithm, we introduced a $Fitness$ to evaluate the algorithm. The $Fitness$ is set as the following equation.

$$Fitness = \sum_{i=1}^{n} \left(W_i^{-1}(t) \times \rho_i\right)$$

$$W_i(t) = 0.5(dt/h) \times W_i(t - dt) + \{1 - 0.5(dt/h)\} \times \rho_i$$

$$\rho_i = D_i \times T_i \quad | \quad D_i < C$$

The above equation is a method for calculation of $Fitness$ when there are n users. When bandwidth requested by user i is D, and requested duration (requested end time - requested start time) is T, network usage ρ of the user i is defined as the product of D and T. C is available bandwidth of links of straight route from source node to destination node during request time. To utilize the network efficiently, it may be suggested that a scheduling scheme that maximizes the sum of network usage for each user is required. However, scheduling that maximizes the network usage for each user will create a bias in users that can use the network, and may cause a situation where fairness cannot be maintained. Therefore, the weight W is incorporated into the scheduling system to consider the users' past network usage.

We have incorporated W as a parameter with reference to $Condor$[7], a job scheduling system that aims to use computing resources efficiently and equally in a distributed computing environment. $Condor$ allocates resources equally to each user, and therefore uses a priority determining mechanism that considers the user's past resource usage[8]. This mechanism is very effective in equally allocating the resource to Layer-1 BoD users. W is the weight of a user i at time t, and this value depends greatly on the weight W of user i from that user's last scheduled time and the network usage ρ is obtained using time t. This value increases when the user uses the network, and decreases with time since the last usage. The rate at which the value decreases can be determined using half-life period h. $Fitness$ is determined as the sum of the product of each user's network usage and its weight. Here, we aimed to achieve request scheduling that can maintain a high value of this $Fitness$. By doing so, each user will be able to utilize the network efficiently and fairly.

C. Algorithm scheduling strategy

In this paper, we propose the scheduling algorithm shown in Figure 1 that satisfies the above requirements.

This scheduling algorithm determines the request's order to be handled at the process marked with (*), and whether to accept or reject each request is determined according to the request's order. In addition, routing rules that determine which link is used between the start and end points specified by the request selects the path with minimum latency available at the time the request is being processed. Therefore, the request's order determined in the process marked with (*) has a very large impact on the evaluation.

The following section describes the optimization method for determining the request's order in the process marked with (*).

D. Ordering optimization

We applied algorithms shown below which optimize requests ordering to our scheduling algorithm.

1) Requests order optimization using GA
   Genetic Algorithms (GA)[9][10] use multiple "individuals" that express data (optimal solution candidates) as analogs of genes in a process modeled on biological evolution, and selects individuals with high

$Fitness$ f preferentially to search for a solution by repeating operations, such as crossover (recombination) and mutation. $Fitness$ f is obtained from a $Fitness$ function.

In this algorithm, the individuals are "requests order," and we selected the "order-based crossover[11]" method that is typically used when solving discrete problems, such as the traveling salesman problem, as a method of crossover. $Fitness$ function is the sum of satisfaction for all users. We introduced "Order Crossover" one of the fundamental techniques in order optimization as crossover operator, and "The Exchange Mutation Operator" as mutation operator.

2) Requests order optimization using LS
Local Search (LS)[15] is one of the simplest algorithm frameworks in approximation algorithms. LS is an optimization method that selects one of the solutions close to the current solution based on specific condition as a neighborhood solution.
Here, the neighborhood solution was created by applying 2-Change[13] to the current solution.
3) Requests order optimization using GR
The Greedy Algorithm (GR)[13][14] divides the elements in a problem into multiple partial problems, evaluates each partial problem individually, and takes the results in the order of the highest to lowest evaluation values. In GR, an element that has been selected will not be considered again.
4) Determining request order randomly
To compare the above algorithms, we used an algorithm that does not consider the $Fitness$ as such FCFS, and determines the order of requests in the queue randomly. SINET3 has adopted FCFS as an order of processing a request.

IV. Numerical experiment

Here, a simulator that imitates Layer-1 BoD was used to analyze what types of behavior are observed from each algorithm and what types of problem may arise from each. We examined which part should be focused upon to improve the scheduling algorithm for better scheduling.

A. Experimental environment

As a numerical experiment, scheduling was analyzed for each algorithm for a topology with multiple nodes and links as shown in Figure 2. In this environment, we attempted to confirm what types of request should be accepted for a link to operate the network link efficiently, and to confirm how a $Fitness$ affects the scheduling results. Parameters for GA and LS used here are shown in Table I. Each link has a capacity of 100 Mbps, and latency was set as 3 ms.

Data representing rules for users submitting requests are shown in Table II. The point where communication takes place is determined from the "Source node" and the

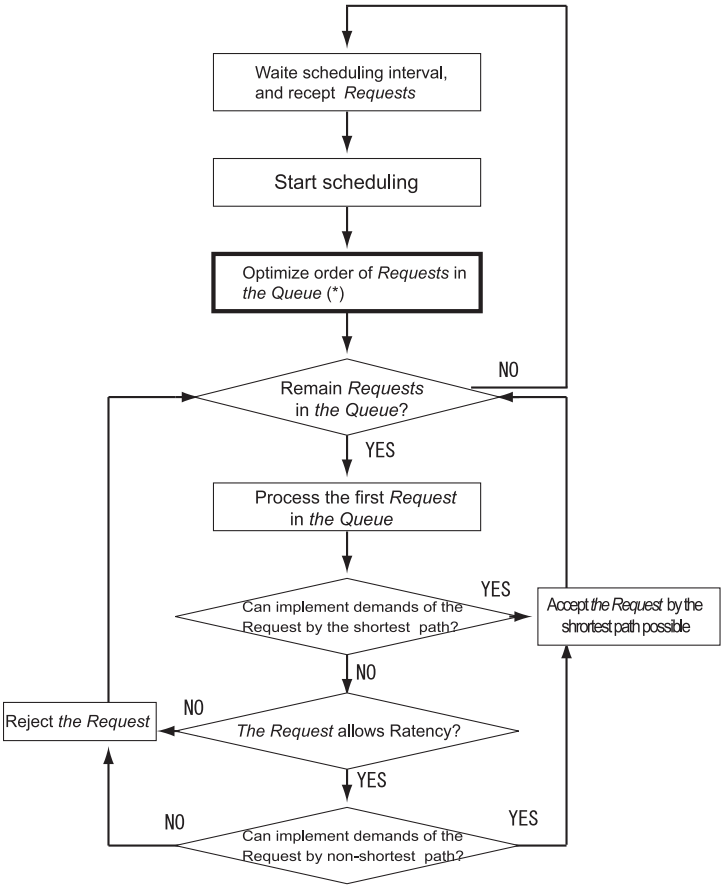

Figure 1. Proposed scheduling algorithm

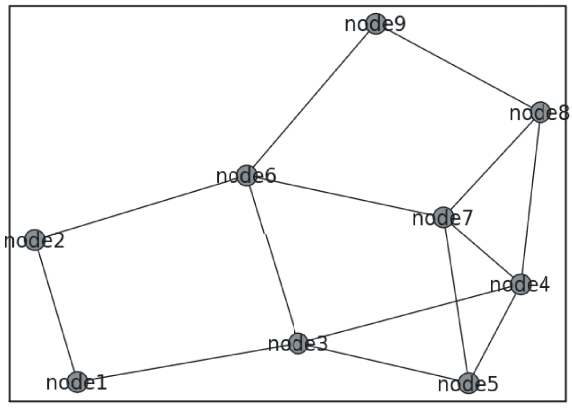

Figure 2. Topology in the numerical experiment

Table I
PARAMATERS

GA	Population size	20
	Generation num	200
	Crossover rate	0.7
	Mutation rate	0.3
Local Search	Trial num	4000

”Destination node”. ”Allowance latency” shows whether permission is given to accept a request even when the communication latency is not minimal. ”Bandwidth requirement” and ”Duration” indicate the bandwidth used during communication and the time of usage, respectively. ”Requested start time” is a numerical value that indicates after how many steps communication is started after submission. ”Wait interval” is the time interval between each submission. In this paper ”Wait interval” is common for all users to ensure scheduling. In addition, to facilitate analysis, users were divided into four groups according to their ”Bandwidth requirement” and ”Requested duration”. The dotted line in Table II indicates these groups. From the top, the groups are: users with both small ”Bandwidth requirement” and ”Requested duration” (GroupI), users with large ”Bandwidth requirement” but short ”Requested duration” (GroupII), users with both large ”Bandwidth requirement” and ”Requested duration” (GroupIII), and users with small ”Bandwidth requirement” and long ”Requested duration” (GroupIV).

Table II
USER'S RULE SUBMISSION REQUESTS

Group	UserName	Source	Destination	Allowance Latency	Bandwidth (Mbps)	LinkUp Time(step)	Communication Time(step)	Submit Interval(step)
	user0	node6	node9	FALSE	9	6	6	100
	user1	node9	node4	TRUE	8	8	7	100
	user2	node3	node8	TRUE	9	5	7	100
	user3	node1	node4	FALSE	10	8	2	100
I	user4	node2	node1	FALSE	6	6	11	100
	user5	node6	node4	TRUE	6	10	6	100
	user6	node4	node1	TRUE	10	10	12	100
	user7	node2	node9	FALSE	7	10	2	100
	user8	node7	node8	FALSE	9	10	7	100
	user9	node6	node8	FALSE	58	6	6	100
	user10	node5	node1	TRUE	60	9	9	100
	user11	node4	node7	FALSE	56	7	6	100
	user12	node2	node5	TRUE	51	7	9	100
II	user13	node5	node7	TRUE	50	8	9	100
	user14	node6	node4	TRUE	57	6	5	100
	user15	node7	node6	FALSE	56	8	12	100
	user16	node9	node3	TRUE	55	6	6	100
	user17	node9	node4	FALSE	60	9	12	100
	user18	node1	node5	FALSE	66	5	25	100
	user19	node4	node5	FALSE	80	7	24	100
	user20	node8	node4	TRUE	71	9	13	100
	user21	node5	node2	FALSE	76	9	15	100
III	user22	node1	node6	FALSE	63	9	22	100
	user23	node9	node1	TRUE	69	9	12	100
	user24	node1	node7	FALSE	73	5	23	100
	user25	node6	node8	FALSE	80	6	24	100
	user26	node7	node4	TRUE	65	7	19	100
	user27	node1	node7	FALSE	28	9	13	100
	user28	node6	node3	TRUE	15	9	8	100
	user29	node3	node4	TRUE	27	7	13	100
	user30	node4	node2	FALSE	17	7	11	100
IV	user31	node3	node8	FALSE	29	9	9	100
	user32	node7	node6	FALSE	30	10	11	100
	user33	node6	node3	FALSE	20	5	17	100
	user34	node1	node5	FALSE	26	9	12	100
	user35	node7	node3	TRUE	24	5	15	100

B. Results of experiment 1

Here, we confirm what type of evaluation was obtained through one time scheduling in the above experiment environment. Scheduling was performed 5 times for each algorithm, and the evaluation results are shown in Figure 4. The search results from GA and LS are shown in Figure 4. In addition, Table III indicates how routing was allocated for each request in each method in one trial. Table III shows the nodes through which the requests from each user pass in transmitting the information. The more nodes the request goes through, the more links it uses. The blank column indicates that the request from the user was rejected.

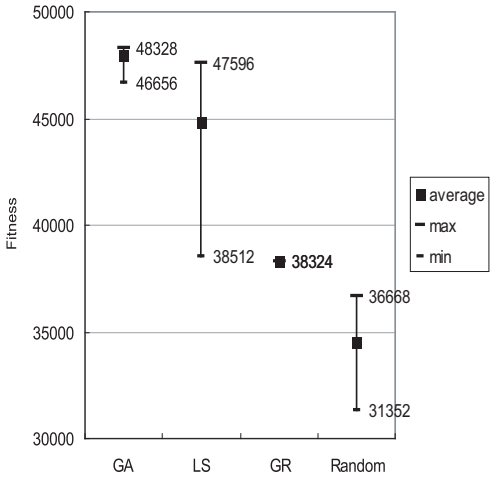

Figure 3. $Fitness$ for each method

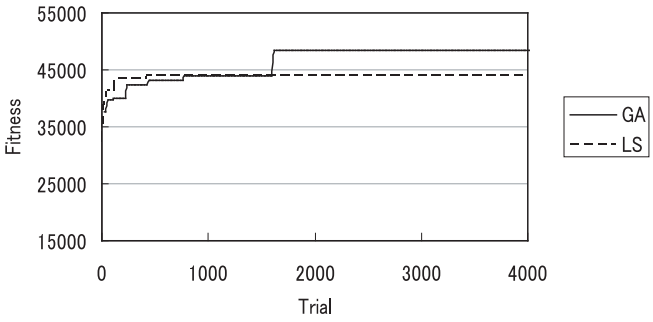

Figure 4. Search by GA and LS

As shown in Figure 3, the $Fitness$ is highest in the order GA, LS, GR, and Random. This experiment involved one time scheduling and thus past usage of each user was not considered. Therefore, the $Fitness$ equals the sum of network usage ρ for each user, and the results in Figure 3 indicate how well the network resources were utilized. Thus, algorithms with high $Fitness$ in this experiment were effectively utilizing the network resource.

In Figure 4, comparison of search from GA and LS indicated that LS converged into a local solution after approximately 500 trials, and subsequent solution exchange did not occur. GA converged into a local solution after approximately 800 trials with a $Fitness$ similar to the local solution obtained with LS, but after approximately 1700 trials, solution exchange did occur. However, no solution exchange occurred thereafter. These observations indicated that the environment used in this experiment was a problem

Table III
ROUTING FOR EACH METHOD

Group	User	GA	LS	GR	Random
	user0	node6, node9	node6, node9		node6, node9
	user1	node9, node8, node4	node9, node8, node4	node9, node8, node4	node9, node8, node4
	user2	node3, node4, node8	node3, node4, node8	node3, node5, node7, node8	node3, node1, node2, node6, node7, node8
	user3	node1, node3, node4			
I	user4	node2, node1	node2, node1	node2, node1	
	user5	node6, node3, node4	node6, node3, node4	node6, node7, node5, node4	node6, node9, node8, node4
	user6	node4, node3, node1		node4, node5, node7, node6, node2, node1	node4, node7, node6, node2, node1
	user7	node2, node6, node9	node2, node6, node9	node2, node6, node9	node2, node6, node9
	user8	node7, node8	node7, node8	node7, node8	node7, node8
	user9				
	user10				
	user11				
	user12				
II	user13	node5, node7	node5, node7		node5, node7
	user14				node6, node3, node4
	user15				
	user16	node9, node6, node3	node9, node6, node3		
	user17				
	user18	node1, node3, node5			
	user19	node4, node5	node4, node5	node4, node5	
	user20	node8, node4	node8, node4		
	user21		node5, node3, node1, node2		
III	user22	node1, node2, node6		node1, node2, node6	node1, node2, node6
	user23				node9, node8, node4, node5, node3, node1
	user24			node1, node3, node4, node7	
	user25	node6, node7, node8	node6, node7, node8	node6, node7, node8	node6, node7, node8
	user26	node7, node4	node7, node4	node7, node5, node3, node6, node9, node8, node4	node7, node4
	user27				
	user28	node6, node3	node6, node3		node6, node3
	user29	node3, node4	node3, node4	node3, node6, node9, node8, node4	node3, node5, node4
	user30	node4, node3, node1, node2	node4, node3, node1, node2		node4, node3, node1, node2
IV	user31				
	user32				
	user33	node6, node3	node6, node3		node6, node3
	user34			node1, node3, node5	
	user35	node7, node4, node3	node7, node4, node3	node7, node4, node3	node7, node4, node3

where algorithms could easily fall into a local solution.

Table III indicates that GA and LS have relatively small numbers of links utilized by each user, while GR and Random have relatively large numbers of links used. Although in GR, users with a large value of ρ have the priority, GR has less accepted requests from the group with larger ρ value than GA. We believe this is because of the marked impact from routing at the time of request acceptance. The request from user26 can be communicated with a minimum of 1 link, but in GR, 6 links are used. This indicates that the resource allocation in GR is very inefficient. Random also handles requests with large ρ, such as request from users in GroupII, through inefficient routing. GA, on the other hand handles requests with large ρ from GroupIII using as few links as possible, thus utilizing the resources efficiently. This suggests that GA performs scheduling more efficiently compared to LS, GR, and Random. However, the number of accepted requests from users in GroupII was small in all algorithms. This suggests that with the $Fitness$ setting used in this experiment, when requests from users from GroupI I conflicted with those from users from other groups, the requests from GroupII users are unlikely to be accepted.

C. Results of experiment 2

Next, we conducted an experiment under conditions where multiple scheduling will occur in a single simulation to analyze whether past use of network resources will be considered when scheduling. This is to ensure the fairness for each user. Each user submits a request following a certain pattern. The time interval between submission from each user was common for all users, and was 100 steps, as indicated in Table II. The length of the simulation was 1200 steps. Thus, scheduling took place 12 times in this experiment.

Table IV shows the sum of accumulated network usage ρ from step 1 to 1200 in a user with the largest sum, with the smallest sum, median sum, and sum of all users. Table V shows the number of accepted requests and their sum, and the variance for all users.

Table IV
ACCUMULATED NETWORK USAGE(ρ)

Algorithm	Total	Max	Min	Median
GA	107191	21120	120	1635
LS	99609	15360	168	1605
GR	114972	23040	0	0
Random	94936	17280	0	1545

Table IV shows that the value of Total is the highest in GR, followed by GA, LS, and Random in this order. Thus, in terms of throughput, this is the order of good performance. However, the values of Min and Median are both 0 in GR. This indicates that there are users that were never accepted, and thus the request acceptance for users was greatly biased. The Random algorithm also had users whose requests were never accepted. In GA and LS, there were no users with 0 accepted requests, and the Median value was higher than those of GR and Random. Glancing at GA's Min, It looks low value. However, It cause particle request's size

Table V
NUMBER OF ACCEPTED REQUESTS

Group	User	GA	LS	GR	Random
	user0	12	12	0	12
	user1	12	12	12	12
	user2	12	12	12	5
	user3	6	9	0	2
I	user4	12	12	12	2
	user5	12	12	12	7
	user6	10	11	12	8
	user7	12	12	12	2
	user8	12	12	12	5
	user9	3	3	0	3
	user10	3	4	0	1
	user11	5	4	0	5
	user12	3	3	0	12
II	user13	10	9	0	3
	user14	4	4	0	0
	user15	4	4	0	5
	user16	5	5	0	4
	user17	6	6	0	2
	user18	3	2	0	4
	user19	11	8	12	6
	user20	5	5	0	3
	user21	3	2	0	12
III	user22	4	4	12	11
	user23	3	3	0	8
	user24	3	3	12	9
	user25	5	4	12	5
	user26	5	6	12	8
	user27	4	3	0	9
	user28	12	11	0	6
	user29	8	10	12	9
	user30	4	5	0	11
IV	user31	5	5	0	12
	user32	7	8	0	12
	user33	12	12	0	12
	user34	3	5	12	12
	user35	10	11	12	1
	Total	250	253	180	240
	variance	13.4	13.5	36.0	15.8

composed by bandwidth and duration dependence. Even if the user with small availability has many requests accepted, he may be around 120. These indicate that while maintaining fairness for all users, the throughput is kept higher than other algorithms.

Table V shows that the number of accepted requests for all users was highest in LS, followed by GA, Random, and GR in this order. In GR, the same scheduling was repeated, and the users were divided into groups according to whether their requests were accepted or rejected. GA, LS, and Random were compared; the value of Total was the highest in LS, followed by GA, and then Random in this order. However, variance improved in the order GA, followed by LS, and then Random. It should be noted that the minimum acceptance number in GA was 3 times, while in Random, there was a user with 0 accepted requests in GroupII. These observations indicated that in GA and LS, even users in GroupII that had less chance of being accepted, will be accepted when multiple scheduling takes place because of the mechanism that considers fairness.

The observations described above indicated that $Fitness$ has an effect on both throughput and fairness during scheduling. However, the situation of GR with biased acceptance indicates that a weight setting that is more suitable for the environment is required.

V. CONCLUSION

In this paper, we proposed a scheduling algorithm for Layer-1 BoD, and introduced $Fitness$ as a method to search for solutions using this scheduling algorithm. The algorithm was compared to GA, LS, and GR in our scheduling algorithm's framework. The results indicated that scheduling with higher $Fitness$ for throughput and fairness would result in more efficient network operation. In future, we plan to improve the weight setting and develop an algorithm with better performance that can search for a better solution for throughput and fairness.

REFERENCES

[1] SINET3: http://www.sinet.ad.jp/sinet3/

[2] Shigeo Urushidani et al, *Design of Versatile Academic Infrastructure for Multilyaer Network Services*, IEEE Journal on Selected Areas in Communications, Apr.2009

[3] http://www.internet2.edu/

[4] http://akari-project.nict.go.jp/overview.htm

[5] Hidehiro Kobayashi et al, *Bandwidth Allocation using Genetic Algorithms*, Special Issue on Multimedia Communication and Distributed Processing1999

[6] Louis Anthony Cox,Jr.,et al, *Dynamic Anticipatory Routing In Circuit-Switched Telecommunications Networks*, Handbook of Genetic Algorithms, 1991, pp124-143

[7] A Bricker, M Litzkow, M Livny, *Computer Sciences Department, University of Wisconsin*, 1992

[8] D Thain, T Tannenbaum, M Livny, *Grid Computing: Making the Global Infrastructure a Reality*, 2002

[9] Holland, J.H, *Adaptation in Natural and Artificial Systems*, University of Michigan Press(1975)

[10] Morgan Kaufman, *A Genetic Algorithm for the Point to Multipoint Routing Problem with Varying Number of Request of Genetic AZgorithms 4*, 1993

[11] G. Syswerda, *Schedule Optimization Using Genetic Algorithms*, In Handbook of Genetic Algorithms. l. Davis, ed. Van Nostrand Reinhold, New York(1991)

[12] I. M. Oliver, D. J. Smith and J. R. C. Holland, *A study of permutation crossover operators on the traveling salesman problem*, Proc. Second Int. Conf. Genetic Algorithms and their Applications, pp. 224-230

[13] P Preux, EG Talbi, *Towards hybrid evolutionary algorithms*, International Transactions in Operational Research, 1999

[14] Cormen, Leiserson, and Rivest, *Introduction to Algorithms*, 1990, Chapter 16 "Greedy Algorithms" p. 329

[15] Christos Voudouris Edward P. K. Tsang, *Guided Local Search", Handbook of metaheuristics*, PP185-218