Network power saving based on Pareto optimal control with evolutionary approach

Yosuke Akishita, Yuichi Ohsita, Masayuki Murata
Graduate School of Information Science and Technology, Osaka University
Email:{y-akishita,y-ohsita,murata}@ist.osaka-u.ac.jp

***Abstract*—The power consumption of networks has been increasing as the service over the Internet becomes popular, and has become a serious problem. Many methods to reduce the power consumption by shutting down unnecessary network devices following the environmental changes have been proposed. These methods consider only simple objectives such as the number of powered-on nodes and the maximum link utilization. However, multiple complex objectives such as delay and reliability should be also considered in the actual network. In this paper, we propose a network power saving method that handles multiple complex objectives, following the environmental changes. In this method, we store the candidate network configurations, and evolve them, following the environmental changes. Then, we select the network configuration from the candidate network configurations. We combine two approaches to evolve the network configurations. The first approach is based on Pareto optimal, and evolves the network configurations so as to be close to the Pareto optimal solutions, considering multiple objectives. Another approach is based on the diversity of the network configurations. By storing the diverse network configurations, we can handle the significant environmental changes. We evaluate our method by simulation, and demonstrate that our method reduces the power consumption without violating the constraints, following the traffic changes. In addition, we also demonstrate that our method can keep the connectivity in case of failures, and recover the performance and the small power consumption soon after the failure occurs.**

***Index Terms*—Network power saving, Evolutionary approach, Pareto optimal solutions, Pareto front, Multi-objective evolutionary algorithms**

I. Introduction

Network traffic has been increasing as the service over the Internet such as streaming and cloud service becomes popular [1], and the power consumption of networks has also been increasing [2], [3]. The power consumption has become one of the important problems in networks.

The power consumption of networks can be saved by shutting down unnecessary network devices, following the changes in the traffic demands; when the traffic demands is small, only a small number of nodes are required to be powered on to accommodate the traffic. On the other hand, when the traffic demands becomes large, more nodes should be powered on to accommodate the traffic without congestion.

Many methods to reduce the power consumption have been proposed [4]–[6]. The method proposed by Amaldi et al. sets the OSPF link weights so that the energy consumption is minimized by solving the Mixed Integer Linear Problem (MILP) periodically [4]. Chiaraviglio et al. also formulated the MILP that minimizes the number of powered-on nodes under the constraints that the full connectivity should be kept and maximum link utilization should be less than the predefined threshold [5]. They also proposed a heuristic method to solve the problem.

These methods consider only simple objectives such as the number of powered-on nodes and the maximum link utilization. However, the network should satisfy multiple more complex objectives. One of the important objectives is the reliability. The reliability may be degraded by powering-off links. For example, the network, some of whose nodes are shut down, can be disconnected in case of failures. In this case, the network service becomes unavailable until the connectivity is repaired by powering up links or nodes. But powering up the nodes takes a time.

The performance is another important objectives, which have large impacts on the application. There are several objectives related to performance, such as delay and jitter and so on. Some of these objectives are non-linear and complex, compared with the maximum link utilization.

Though there are multiple objectives, we can use them as the constraints when minimizing the power consumption; for example, a network is configured so as to minimize the power consumption under the constraints that the delay should be less than a predefined threshold and the number of distinct paths on the powered-on paths should be larger than the predefined value. By using objectives as the constraints, we can formulate an optimization problem, considering such multiple complex objectives. However, it takes a long time to solve the optimization problem, because the optimization problem includes as many binary variables as the number of links, and the objective functions are non-linear and complex.

In this paper, we propose a new method to control a network to save the energy consumption, which can handle complex multiple objective functions, following the environmental changes. In our method, we store candidate network configurations and evolve them so as to follow the environmental changes in the network. Then, we select the network configuration that minimizes the energy consumption and satisfies the requirements from the candidate network configurations. By evolving the network configurations from the previous ones, we can obtain the suitable network configuration fast.

When evolving the candidate network configurations, we combine two approaches that can handle multiple objective functions. The first approach is based on Pareto optimal. Pareto

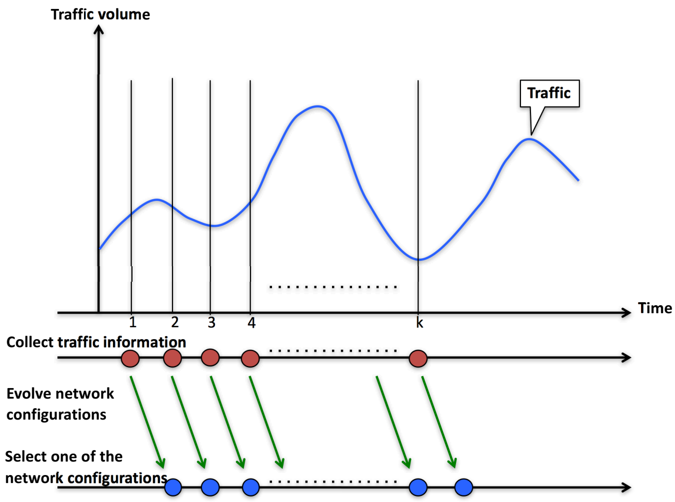


Fig. 1. The overview of our method

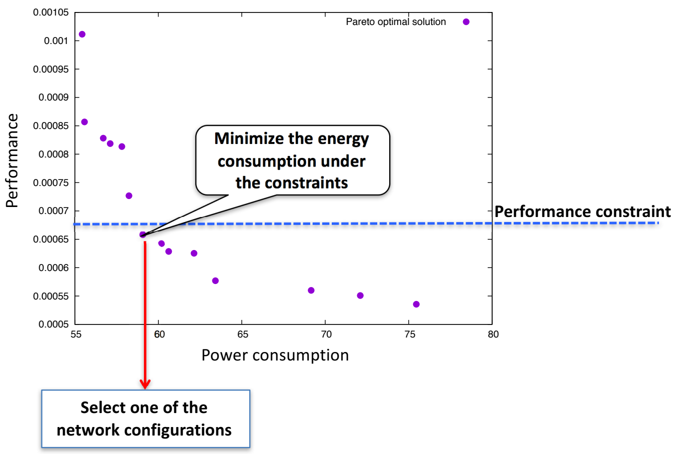


Fig. 2. Selection of the network configuration

optimal solutions are the solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives. A suitable solution considering multiple objectives is a Pareto optimal solution. The Pareto optimal solutions can be obtained by an evolutional algorithm called Multi-Objective Evolutionary Algorithms (MO-EA) [7]–[12]. In this paper, we calculate the Pareto optimal solutions by using this algorithm.

Another approach is based on the diversity of the network configurations. By storing diverse network configurations, we can find the suitable configuration in the set of the stored network configurations even when the significant environmental changes occur.

The rest of this paper is organized as follows. In Section 2, we propose a method to save the energy consumption. We evaluate our method in Section 3. The conclusion is drawn in Section 4.

II. Network Power Saving with Evolutionary Approach

A. Overview of our method

In this section, we propose a method to control the network to save the power consumption, considering multiple complex objectives. To follow the environmental changes, we need to calculate the suitable network configuration periodically. However, it takes a long time to solve the optimization problem including multiple complex objective functions.

In our method, we store the candidate network configurations and evolve them following the environmental changes, instead of solving the optimization problem periodically. Figure 1 shows the overview of our method. In this method, we periodically collect traffic information. Then, we evolve the candidate network configurations based on the collected traffic information. Finally, we select one of the network configurations that minimizes the energy consumption and satisfies the requirements as shown in Figure 2. By continuing these steps, our method follows the environmental changes.

B. Evolution of network configurations

In this section, we explain how we evolve the network configurations. In this paper, we use two approaches. The first one is based on the Pareto optimal. The problem we solve is a multi-objective optimization problem. The Pareto optimal solutions are the solutions of the multi-objective optimization problem, which are solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives as shown in Figure 3. That is, setting the network based on one of the Pareto optimal solutions, we can control the network, considering all of the objectives.

Another one is based on the diversity. By holding the diverse network configurations, we can find the suitable solutions even when the significant environmental changes occur.

In our method, we combine both of approaches. That is, we store both of the network configurations evolved by the approach based on the Pareto optimal and those by the approach based on the diversity.

1) Evolution based on Pareto optimal:

a) Pareto optimal solutions and Pareto front: In multi-objective optimization problems, it is impossible to obtain a complete optimal solution to all of the given objective functions, because the objective functions compete with each other. Therefore, the Pareto solutions are obtained. x^* is a Pareto optimal solution when there is no x that satisfies Eqs. (1) and (2).

$$f_i(x) \leq f_i(x^*) \ \forall i = 1, ..., p \tag{1}$$

$$f_i(x) < f_i(x^*) \ \exists i \in \{1, ..., p\} \tag{2}$$

where $f_i(x)$ is ith objective function. In general, there are a number of Pareto optimal solutions, and these solutions form surface which is called Pareto front.

b) Calculation of Pareto optimal solutions based on the evolutional algorithm: Pareto optimal solutions can be obtained by an evolutional algorithm called MO-EA [13]. In MO-EA, solutions are coded as a gene. MO-EA evolves the

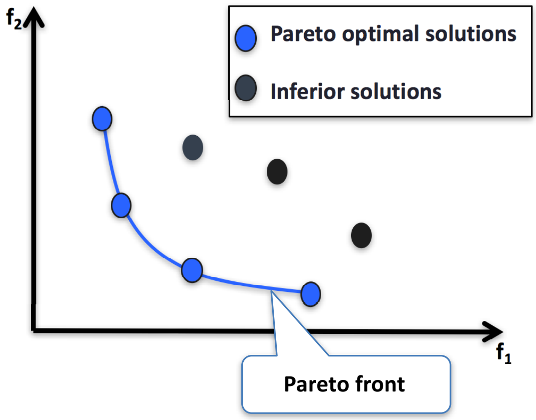


Fig. 3. Pareto optimal solutions

genes by using the mutation and crossover operators so that the genes approach to the Pareto optimal solutions.

MO-EA performs the following steps.

1) Initialization : generate N individuals, and denote the set of individuals as P.
2) Evaluation : rank the individuals based on the non-dominated sort, and calculate the density in each rank.
3) Generation of offspring : generate N offspring by manipulating genes (selection, crossover, and mutation), and denote them as Q
4) Replacement of the old solutions : update P to N solutions selected from $P \cup Q$, and save the solutions in rank 1 to Pareto archive.
5) Finish of determination : End when the end conditions are satisfied. Then, the solutions in Pareto archive form the Pareto front. Go back to Step.3 otherwise.

MO-EA ranks the individuals by using non-dominated sort [13]. When a solution A is superior to another solution B in all objective functions, the solution A *dominates* the solution B. The non-dominated sort ranks the solutions based on the number of dominating solutions. The non-dominated sort procedures are summarized as follows.

1) Initialize n to 1.
2) For each solution, count the number of the solutions dominating the solution, and the number of the solutions dominated by the solution.
3) For each solution that is not included in the lists, if the number of solutions dominating the solution is 0, add it to the lists F_n.
4) For each solution, subtract 1 from the number of the solutions dominating the solution if the solution is dominated by the solutions in F_n.
5) End if the number of the solutions that are not included in any lists is 0. Otherwise, increment n and go back to Step 2.

After the above steps, the list of F_n includes the solutions of the rank n.

Priorities between the individuals with the same rank are determined by using crowding distance [13]. Crowding distance c_x is a metric indicating the density of the individuals near the individual x, and is defined by

$$c_x = \begin{cases} \infty & \text{max or min on any criteria of } x \\ \sum_m \frac{f_m(I_m^{\text{next}}(x)) - f_m(I_m^{\text{prev}}(x))}{f_m^{max} - f_m^{min}} & \text{otherwise} \end{cases}$$

where $I_m^{\text{prev}}(x)$ is the individual whose value of the mth objective function is the largest among the individuals whose values are smaller than the value of x, and $I_m^{\text{next}}(x)$ is the individual whose value of the mth objective function is the smallest among the individuals whose values are larger than the value of x. f_m^{max} is the maximum value of the mth objective function, and f_m^{min} is the minimum value of mth objective function. By selecting the solution with a large c_x, we select the solutions that are different from the other solutions.

c) Evolution based on Pareto Optimal in our method: In this paper, we evolve the network configurations based on the MO-EA. That is, we perform the following step several times in each time slot.

1) Evaluation : rank the network configurations based on the non-dominated sort, and calculate the density in each rank.
2) Generation of offspring : generate offspring by manipulating network configurations
3) Replacement of the old solutions : update the set of network configuration, and save the solutions in rank 1 to Pareto archive.

Finally, we store the solutions in Pareto archive for the next time slot.

In the above steps, our method generates new offspring by selection, crossover and mutation as follow.

Selection and crossover: We select network configurations based on a tournament strategy; we select k network configurations randomly, and select the best individual from them by using rank and crowding distance. By performing the above steps twice, we select two network configurations. The network configuration includes the paths between nodes, and the information on the powered-on links. Then, we perform the crossover operation, which swaps randomly selected path in the selected two network configurations to generate new offspring.

Mutation: Mutation operation includes two mutation step; the path mutation and the link mutation. In the path mutation step, we select one path randomly, and select one node on the selected path randomly. Then, we generate a new path which includes randomly generated paths from the selected node to the destination node. In the link mutation step, we select a node randomly, and turn off the links connected to the selected node. Then, the paths are recalculated so as not to pass the selected node.

C. Evolution based on the diversity

Pareto front may become significantly different from the previous one if the significant environmental changes occur. In this case, the network configuration evolved from the solutions in the Pareto archive cannot provide the required performance. Therefore, we hold the network configurations evolved by the different strategy.

In this strategy, we evolve the network configurations so that the diverse network configurations are stored. This kind of network configurations are generated after completing the evolution based on the Pareto optimal. Then, we evaluate all the candidate network configurations using the metric $\mathrm{Ev}(x)$, which is defined by

$$\mathrm{Ev}(x) = \mathrm{Distance}(x) \times \mathrm{Sim}(x)$$

where $\mathrm{Distance}(x)$ is difference between x and a solution in the Pareto front, and $\mathrm{Sim}(x)$ is the similarity between x and the solutions in the set of the other stored network configurations. By selecting the solution with small $\mathrm{Ev}(x)$, we can store the diverse solutions that are close to Pareto front.

In this paper, $\mathrm{Distance}(x)$ is defined as the distance from x to the closest solution in Pareto front, and is calculated by

$$\mathrm{Distance}(x) = \min_{x' \in P} \sum_m (f_m(x) - f_m(x'))^2$$

where P is the set of the solutions on the Pareto front.

To define $\mathrm{Sim}(x)$, we use the maximum link utilization on each path as metrics. First, $S(x, x')$ is defined by

$$S(x, x') = \frac{1}{\sqrt{\sum_{i,j} (l_{i,j}(x) - l_{i,j}(x'))^2}}$$

where $l_{i,j}(x)$ is the maximum link utilization on the path from i to j when configuring the network configuration x. $S(x, x')$ becomes small when x and x' uses the different paths.

Then, $S(x)$ is defined by maximum value of $S(x, x')$. That is,

$$S(x) = \max_{x' \in R} S(x, x')$$

where R is the set of the solutions that are already stored. Finally, $\mathrm{Sim}(x)$ is defined by normalizing $S(x)$ so that $0 \leq \mathrm{Sim}(x) \leq 1$.

$$\mathrm{Sim}(x) = \frac{S(x) - \min_{x'} S(x')}{\max_{x'} S(x') - \min_{x'} S(x')}$$

III. Evaluation

A. Simulation Environment

1) Network topology: In our evaluation, we use a FatTree topology, which is a typical network architecture in data centers. We set the number of ports of each switch to 8 as shown in Figure 4; each pod consists of 16 servers and 2 layers of 4 switches. Each edge switch connects to 4 servers and 4 aggregation switches. Each aggregation switch connects to 4 edge switches and 4 core switches. Each core switch connects to 8 pods. In our evaluation, we set the power consumption of the links and switches based on the fact that the power consumption of nodes is 10 times as large as that of links [4]. Table I shows the power consumption and capacity of links and nodes used in our evaluation.

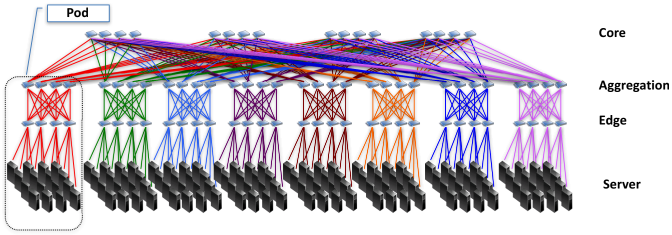


Fig. 4. Network topology

TABLE I
Simulation Environment

Power consumption of node	0.7 [kw]
Power consumption of link	0.07 [kw]
Maximum transmission speed at each port	1 [Gbps]

2) Traffic: In our evaluation, we generate two kinds of traffic.

Traffic Pattern A: This pattern includes daily traffic changes. In this traffic pattern, traffic are generated between selected server pairs and traffic between each node pair changes as shown in Figure 5.

Traffic Pattern B: This pattern includes unexpected sudden traffic changes. For this pattern, we select the pod pair randomly, and add traffic between all server pairs in the selected pod pairs at the time slot 461 as shown in Figure 6. By using this pattern, we investigate whether our method follows such sudden traffic changes.

3) Compared methods: In this paper, we compare the following three methods.

Method with Diverse Solutions (w/ DS): This is our proposed method that evolves the network configuration from both of the Pareto archive and the archive storing the diverse solutions.

Method without Diverse Solution (w/o DS): This is a method that evolves the network configuration only from the solutions in the Pareto archive.

Random (R): This is a method that generate the Pareto optimal solutions by MO-EA method at each time slot, which generate initial solutions randomly at every time slot.

The parameters used in this evaluation are shown in Table II. R requires a longer calculation time than the method w/o DS, because R needs to generate the initial solutions. Similarly, the method w/ DS requires a longer calculation time than the method w/o DS because the method w/ DS holds the archives storing the diverse solutions. Considering this, we set the number of evolution in each time slot as shown in Table III. By these setting, all of three methods takes a similar time to obtain the solution in each time slot.

4) Objective functions and SLA requirements:

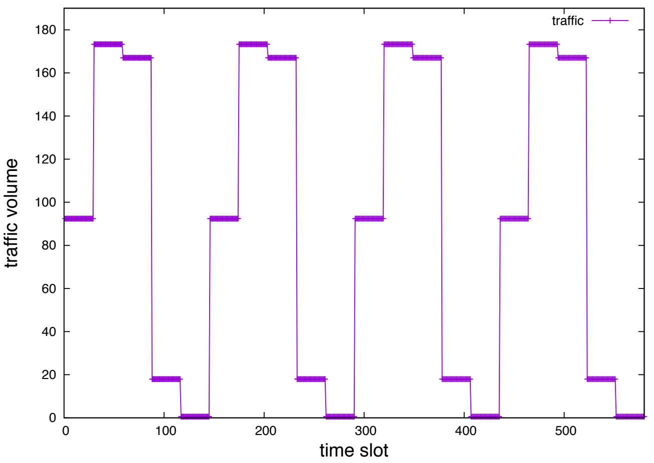


Fig. 5. Traffic fluctuation ; pattern A

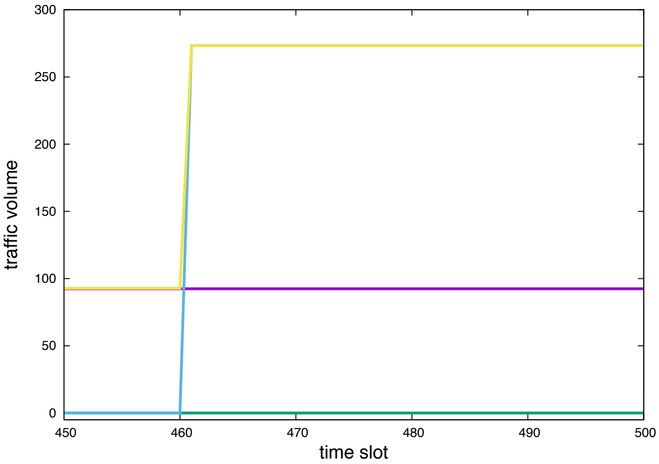


Fig. 6. Traffic fluctuation ; pattern B

a) Objective functions: In this paper, we use the following objective functions in each method.

Power Consumption: In this paper, the network power consumption is defined as the sum of the power consumption of the powered-on links and nodes, and is calculated by

$$E^{\mathrm{net}}(x) = \sum_{(i,j)\in L} E^{\mathrm{Link}} p_{i,j}(x) + \sum_{k\in V} E^{\mathrm{Node}} p_k(x) \tag{3}$$

where L is the set of links, V is the set of nodes, E^{Link} is the power consumption at each link, E^{Node} is power consumption at each node, $p_{i,j}(x)$ is a variable which is 1 when gene x uses link between i and j, 0 otherwise, and $p_k(x)$ is variable which is 1 when x uses node k, 0 otherwise.

Reliability: In this paper, we set the metric of the reliability based on the number of distinct paths on the powered-on links, because we can keep connectivity in case of failures by preparing more distinct path. We define the reliability by

$$R(x) = \frac{1}{\min_{i,j} r_{i,j}(x) + \alpha \sum_{i,j} r_{i,j}(x)}$$

where $r_{i,j}$ is the number of distinct paths between i and j, and α is a parameter. In this evaluation, we set α to $\max_x \sum_{i,j} r_{i,j}(x)$.

Performance: In this paper, we use delays between devices as the metric of performance. That is, the objective function of performance $P(x)$ is defined by

$$P(x) = \max_{i,j} D_{i,j}(x)$$

where $D_{i,j}(x)$ is a delay between i and j when the network configuration x is set, and is calculated by

$$D_{i,j}(x) = \sum_{(s,d)\in q_{i,j}(x)} d_{(s,d)}(x)$$

where $q_{i,j}(x)$ is the set of links on the path from i to j, and $d_{(s,d)}(x)$ is the delay between link s and link d. In this paper, we assume that the delay is modeled by M/M/1 model. That is, $d_{(s,d)}(x)$ is calculated by

$$d_{(s,d)}(x) = T_s \frac{\rho_{s,d}(x)}{1-\rho_{s,d}(x)}$$

where T_s is the average processing times of a packet in each link, and $\rho_{s,d}(x)$ is utilization of link s–d.

b) SLA requirements: In this evaluation, we set the following requirements.

- Delay between any server pairs must be less than 250[μs].
- At least 2 distinct paths should be provided between any server pairs.

TABLE II
PARAMETERS OF MO-EA

Parameter	Value
Number of saved configurations	30
Crossover ratio	0.5
Mutation ratio	0.5

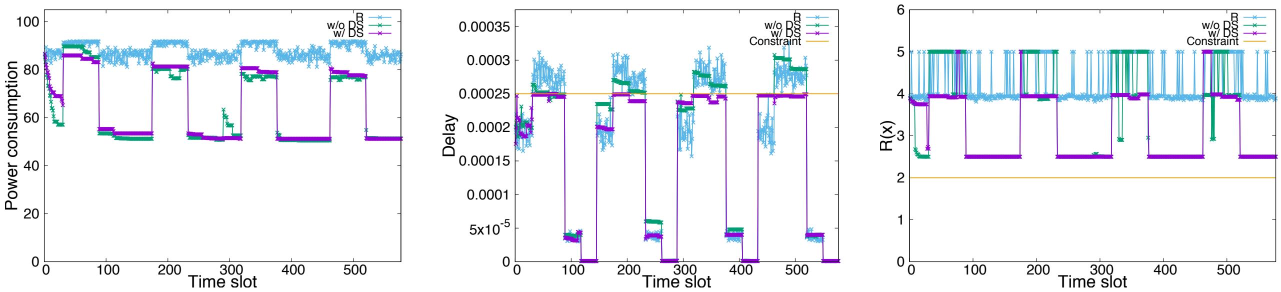
TABLE III
NUMBER OF GENERATIONS FOR EVALUATION

w/o DS	R	w/ DS
50	49	45

B. Results

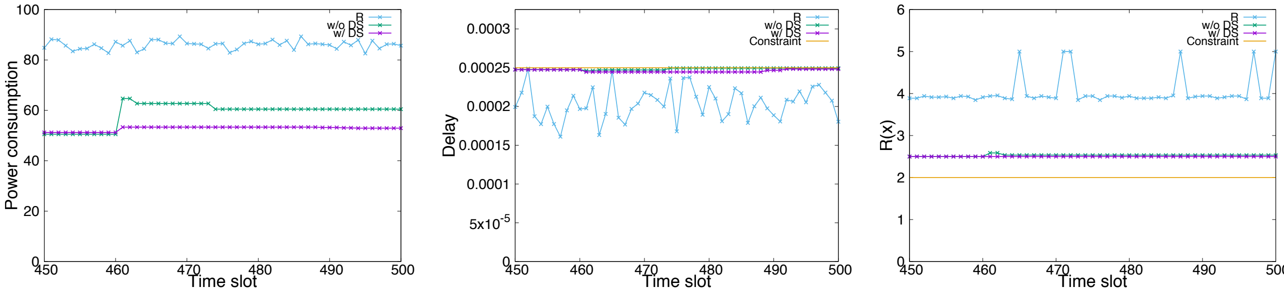
1) In the case without failures : Figure 7 shows the results in the case of traffic pattern A without failures.

Figure 7(a) shows the power consumption at each time slot. The vertical axis indicates power consumption and the horizontal axis indicates time slot. This figure indicates that the methods w/ DS and w/o DS follows the traffic variation; they reduce the power consumption when traffic volume is low, and turn up many devices to accommodate more traffic with a sufficiently small delay when traffic volume becomes



(a) Power consumption (b) Performance (c) Fault tolerance

Fig. 7. Evaluation values (Traffic = pattern A)



(a) Power consumption (b) Performance (c) Fault tolerance

Fig. 8. Evaluation values (Traffic = pattern B)

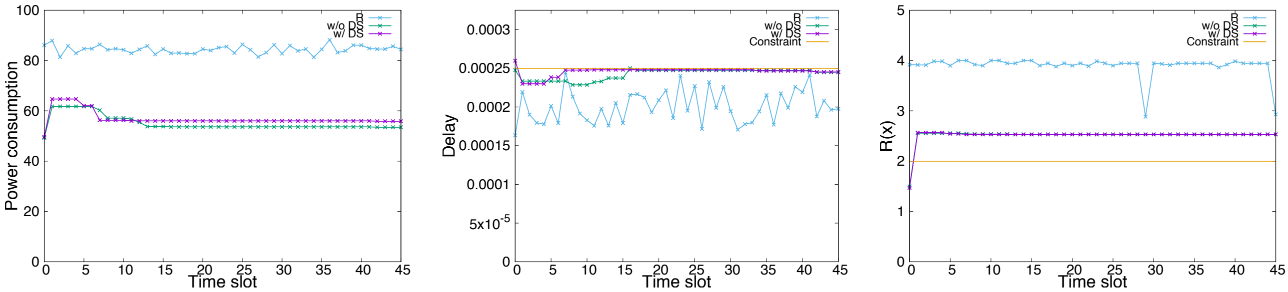
high. However, R cannot reduce the power consumption. This is because the method that does not use the network configurations in the previous time slot cannot obtain appropriate Pareto front that saves power consumption in 49 generations at each time slot. That is, this results demonstrate that evolution from network configurations in the previous time slot is efficient to follow the environmental changes.

Figure 7(b) shows delay and Figure 7(c) shows the number of distinct paths. The vertical axis indicates the maximum delay or $1/R(x)$. The horizontal axis indicates the time slot. In both figures, we plot the constraints of SLA as a line. These figures show that the method w/ DS satisfies all constraints at all time slots. However, the method w/o DS cannot satisfy the constraints at time slot 175, 318, and 461. This is because the method w/o DS removes the solutions that satisfy the constraints at time slot 175, 318, and 416, when the traffic volume is very small. When traffic volume is very small, the delay depends on the number of hops. As a result, the delay has only a small impact on selecting the network configurations to be stored in the Pareto archive, and the network configurations with the smaller energy consumption are stored. Therefore, the network configurations that accommodate more traffic are removed from the Pareto archive. On the other hand, the method w/ DS stores such network configurations that can accommodate more traffic. As a result, the method w/ DS satisfies the constraints.

We also evaluate the methods in the case of traffic pattern B. Figure 8 shows the results. Figures 8(b) and 8(c) show that all methods satisfy the requirements at all time slots. Figure.8(a) shows that power consumption becomes higher at time 461 in both of the methods w/ and w/o DS. This is because the sudden traffic changes causes the lack of the solutions satisfying the SLA constraints. As a result, the solution found in the evolved network configurations requires powering on a number of devices. However, the methods w/ DS decreases the power consumption immediately. This is because the methods w/ DS evolves the network configurations so as to save the energy consumption. On the other hand, the method w/o DS cannot reduce the energy consumption, compared with the method w/ DS. This is because the network configurations stored in the Pareto archive becomes significant different from the suitable network configurations. As a result, a large number of generations are required to achieve the suitable Pareto front. That is, the diversity in the stored solutions also helps the immediate adaptation to the sudden traffic changes.

2) In the case with failures: In this evaluation, we generate a failure at the randomly selected node at time slot 461. In our method, the traffic passing the failed node is immediately rerouted to the shortest paths on the network constructed of powered-on nodes. Then, at the next time slot, we evolve the network configurations, considering the failure of the nodes.

Figure 9 shows the results. In this figure, the vertical axis indicates the power consumption, delay, and the reliability, and the horizontal axis indicates the time slots after the failure occurs. Figures 9(b) and 9(c) show that the methods w/ DS and w/o DS cannot satisfy the SLA constraints, due to the



(a) Power consumption (b) Performance (c) Fault tolerance

Fig. 9. Evaluation values (Traffic = pattern A, Fault occurs at time 461)

reroute of the traffic without considering the SLA constraints. However, all methods satisfy the SLA constraints 1 time slots after the failure. This is because the network configurations are evolved, considering the SLA constraints. As a result, the set of candidate network configurations after the evolution includes network configurations that satisfy the requirements. But, the power consumption increased at the time slot 1. This is because the number of network configurations in the archive that satisfy the SLA requirements is small, though the evolution of the network configurations generates the network configuration satisfying the SLA constraints. Then, the power consumption also decreases by evolving the network configurations. Therefore, our method can also handle the failures, and achieves suitable network configuration.

IV. Conclusion

In this paper, we proposed a network power saving method that handles multiple complex objectives, following the environmental changes. In this method, we hold candidate network configurations and evolve them following the environmental changes in the network. Then, we select the network configuration that minimizes the energy consumption under the constraints.

When evolving the candidate network configurations, we combine two approaches. The first approach is based on Pareto optimal. Another approach is based on the diversity of the network configurations.

In this paper, we evaluated our method by simulation. The results show that our method w/ DS reduces the power consumption without violating the SLA constraints, following the traffic changes. In addition, even when a failure occurs, our method re-builds paths and gradually recovers the network configurations so that the energy consumption is minimized under the SLA constraints.

In this paper, we evaluate our methods in the FatTree topology. Our future work includes to evaluate our method in a more general network structure.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP16K00125.

References

[1] Cisco, "Cisco global cloud index:forecast and methodology, 2014-2019," Cisco Systems Inc., Tech. Rep., Oct. 2015.

[2] Van Heddeghem, Ward and Lambert, Sofie and Lannoo, Bart and Colle, Didier and Pickavet, Mario and Demeester, Piet, "Trends in worldwide ICT electricity consumption from 2007 to 2012," *Computer Communications*, vol. 50, pp. 64–76, Sep. 2014.

[3] Lambert, Sofie and Van Heddeghem, Ward and Vereecken, Willem and Lannoo, Bart and Colle, Didier and Pickavet, Mario, "Worldwide electricity consumption of communication networks," *Optics express*, vol. 20, no. 26, pp. B513–B524, Dec. 2012.

[4] Amaldi, E. and Capone, A. and Gianoli, L. G., "Energy-aware IP traffic engineering with shortest path routing," *Comput. Netw.*, vol. 57, no. 6, pp. 1503–1517, Apr. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2013.02.006

[5] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP network energy cost: formulation and solutions," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 2, pp. 463–476, Apr. 2012.

[6] Chiaraviglio, Luca and Mellia, Marco and Neri, Fabio, "Reducing power consumption in backbone networks," in *Proceedings of Communications, 2009. ICC'09. IEEE International Conference on*. IEEE, Jun. 2009, pp. 1–6.

[7] Kessaci, Yacine and Melab, Nouredine and Talbi, El-Ghazali, "A pareto-based metaheuristic for scheduling HPC applications on a geographically distributed cloud federation," *Cluster Computing*, vol. 16, no. 3, pp. 451–468, Sep. 2013. [Online]. Available: http://dx.doi.org/10.1007/s10586-012-0210-2

[8] Battiti, Roberto and Passerini, Andrea, "Brain–computer evolutionary multiobjective optimization: a genetic algorithm adapting to the decision maker," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 5, pp. 671–687, Sep. 2010.

[9] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, Sep. 2006.

[10] Fadaee, M and Radzi, MAM, "Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: a review," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 5, pp. 3364–3369, Jun. 2012.

[11] Ahmadi, Pouria and Dincer, Ibrahim and Rosen, Marc A, "Thermoeconomic multi-objective optimization of a novel biomass-based integrated energy system," *Energy*, vol. 68, pp. 958–970, Apr. 2014.

[12] Q. Wang, M. Guidolin, D. Savic, and Z. Kapelan, "Two-objective design of benchmark problems of a water distribution system via moeas: Towards the best-known approximation of the true pareto front," *Journal of Water Resources Planning and Management*, vol. 141, no. 3, p. 04014060, Jul. 2014.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, Apr. 2002.