2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

# A Method for Enhancing Robustness of MCMC-Based Autonomous Decentralized Control

Masaya Yokota† Yusuke Sakumoto‡ and Masaki Aida‡

†‡Graduate School of System Design, Tokyo Metropolitan University
6–6 Asahigaoka, Hino-shi, Tokyo 191-0065 Japan
Email: †yokota-masaya@sd.tmu.ac.jp, ‡{sakumoto, aida}@tmu.ac.jp

**Abstract**—ADC (Autonomous Decentralized Control) is being actively discussed to realize a scalable control scheme for large-scale and wide-area systems. The previous work has proposed the MCMC (Markov Chain Monte Carlo)-based ADC for indirectly controlling the entire system, but has not fully discussed for its adaptability against environmental fluctuations. In this paper, we discuss the impact of environmental fluctuation on the MCMC-based ADC, and propose a method to enhance the robustness of the MCMC-based ADC against severe environmental fluctuations. In particular, we design an adjustment method to absorb environmental fluctuations according to the outcome of our discussion. In addition, we apply the proposed method to the virtual machine placement problem in data centers. Through simulation experiment, we clarify the effectiveness of the proposed method for severe environmental fluctuation.

## 1. Introduction

In recent years, progress of ICT (Information and Communications Technology) systems are remarkable [1]. Many ICT systems are supported by large-scale and wide-area data centers to accommodate numerous user requests. For instance, Google has distributed a large number of servers composed of its data center around the world [2], and many ICT systems are using services provided by Google. It is expected such a data center is becoming larger and larger in order to improve computational capability and system reliability. Hence, a scalable control scheme should be realized to handle sustainably large-scale and wide-area systems like data centers.

To build a scalable control scheme for large-scale and wide-area systems, ADC (Autonomous Decentralized Control) is being actively discussed (see, e.g., [3–7]). In general, system controls are roughly categorized as ADC or CC (Centralized Control). Consider a system composed of numerous nodes distributed in a wide area. In a CC, the management node has to gather state information from the entire system, and controls intensively the state of all nodes in the system. Hence, CCs require a large amount of time to react for an environmental fluctuation in a large-scale and wide-area system, and does not handle such a system against dynamic environment. In an ADC, each node autonomously controls its state on the basis of local information. Hence, ADCs can quickly react for an environmental fluctuation. However, an autonomous node action in ADCs should be designed to be able to combine with the control of the entire system [3].

In [6], the authors have proposed an ADC based on MCMC (Markov Chain Monte Carlo) [8, 9], which is a method to generate a Markov process following a desired probability distribution of a statistical-mechanical variable (e.g., energy). In [6], on the basis of MCMC, the authors designed an autonomous node action to be able to combine with the control for the probability distribution of system performance variable that is an amount to quantify of a system state. In [7], the authors have proposed an advanced function of the MCMC-based ADC to adapt against simple environmental fluctuations, but have not fully discussed for its adaptability against environmental fluctuations. There are various kind of environmental fluctuations for actual systems, so it is crucial to discuss adaptability of the MCMC-based ADC and improve its robustness if necessary.

In this paper, we discuss the impact of environmental fluctuation on the MCMC-based ADC, and propose a method to enhance its robustness against severe environmental fluctuations. In particular, we design an adjustment method to absorb environmental fluctuations according to the outcome of our discussion. In the proposed method, each node autonomously adjusts its control parameter of the MCMC-based ADC. In addition, similarly to [6], we apply the proposed method to the virtual machine placement problem in data centers. Through simulation experiment, we clarify the effectiveness of the proposed method for severe environmental fluctuation.

This paper is organized as follows. Section 2 introduces MCMC-based ADC [6]. In section 3, we discuss briefly the impact of environmental fluctuation on the MCMC-based ADC, and design an its adjustment method to absorb environmental fluctuations. Section 4 details the experiments conducted for investigating the robustness of the proposed method in data center networks. Finally, in Section 5, we conclude this paper and discuss future works.

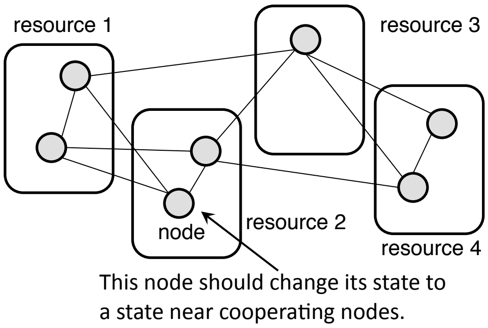


Figure 1: An example of the system model with $n = 7$ and $N = 4$

## 2. MCMC-Based ADC

### 2.1. System Model

We first introduce a system model for the MCMC-based ADC. Consider a system consisting of $n$ nodes. We assume that $N$ resources in the system are shared among nodes, and system performance depends on assignment of the resources for nodes. The state of a node corresponds to the assigned resource for the node, and we denote the state of node $i$ by $x_i \in \{1, ..., N\}$. System state $\boldsymbol{X}$ is given by the combination of all node states, $(x_1, \cdots, x_n)$. We assume that each node cooperates with other nodes to process a common task. Hence, the system should assign same or near resources to cooperating nodes for efficiently processing tasks. Figure 1 shows an example of the system model with $n = 7$ and $N = 4$. In this figure, we draw links among cooperating nodes. We define cooperating degree $r_{ij}$ between nodes $i$ and $j$, and state distance $d(k, l)$ between states $k$ and $l$. Then, we formulate the performance of such a state assignment for nodes $i$ and $j$ by the product of $r_{ij}\, d(x_i, x_j)$, and the total performance for the entire system by the following system performance variable

$$M(\boldsymbol{X}) = \sum_{i=1}^{n} \sum_{j \in \chi_i} r_{ij}\, d(x_i, x_j), \tag{1}$$

where $\chi_i$ is the set of nodes cooperating with node $i$ in the system. By decreasing $M(\boldsymbol{X})$, we can improve the performance of the state assignment considering cooperation among nodes. However, if $M(\boldsymbol{X})$ is too small, nodes only use a few resources, so the node concentration in the state assignment is too high. Hence, $M(\boldsymbol{X})$ should be adjusted suitably for different uses.

### 2.2. Node Action

In the MCMC-based ADC [6], node $i$ changes its state $x_i$ to another state $x'_i$ with state transition probability $T_i(x_i \to x'_i)$, which is designed by MCMC. Specifically, $T_i(x_i \to x'_i)$ is given by

$$T_i(x_i \to x'_i) = \begin{cases} \dfrac{1}{|\phi_{x_i}|} \exp[-\alpha\, \lambda\, \Delta M_i(x_i \to x'_i)] \\ \qquad\qquad \text{if } \Delta M_i(x_i \to x'_i) < 0 \\ \dfrac{1}{|\phi_{x_i}|} \exp[-(1-\alpha)\lambda\, \Delta M_i(x_i \to x'_i)] \\ \qquad\qquad \text{otherwise} \end{cases}, \tag{2}$$

where $\phi_k$ is the set of states that are able to be transited from state $k$, $\lambda$ is the control parameter of the MCMC-based ADC, and $\alpha$ is a positive constant ($0 \le \alpha \le 0.5$). $\Delta M_i(x_i \to x'_i)$ is the amount of change of $M(\boldsymbol{X})$ with respect to the state transition of node $i$, and is given by

$$\Delta M_i(x_i \to x'_i) = \sum_{j \in \chi_i} r_{ij} \left[ d(x'_i, x_j) - d(x_i, x_j) \right]. \tag{3}$$

If nodes uses state transition probability $T_i(x_i \to x'_i)$ by Eq. (2), system performance variable $M$ follows the probability distribution

$$P(M) = \frac{G(M) \exp(-\lambda M)}{\sum_{Y \in \Omega_M} G(Y) \exp(-\lambda Y)}, \tag{4}$$

where $G(Y)$ is the number of system states if the system performance variable is equal to $Y$, and $\Omega_M$ is the set of all possible values of system performance variable $M$. According to Eq. (4), if $\lambda = 0$, $P(M)$ is proportional to $G(M)$. In this case, the MCMC-based ADC is equivalent to the control where each node selects its state at random. In addition, as control parameter $\lambda$ increases, each node controls its state to lead to the emergence of smaller $M$. Therefore, the MCMC-based ADC can adjust $M(\boldsymbol{X})$ by changing $\lambda$.

## 3. Adjustment Method of Control Parameter $\lambda$ to Absorb the Impact of Environmental Fluctuations

In this paper, we consider adaptability to be able to retain performance (i.e., node concentration) of state assignments against environmental fluctuations. This adaptability is crucial for realizing robustness because high node concentration causes heavy load of a few resources, may lead to clash of the system.

We first discuss the impact of environmental fluctuations on state assignments of the MCMC-based ADC. We assume that $r_{ij}$ or the number of nodes, $n$, changes when an environmental fluctuation occurs. Such a fluctuation affects the value of $\Delta M_i(x_i \to x'_i)$, and would change node behavior by the node action using Eq. (2). For instance, if $\Delta M_i(x_i \to x'_i)$ of node $i$ is increased due to a fluctuation (i.e., node addition or increase in $r_{ij}$), node $i$ and cooperating other nodes $j \in \chi_i$ would select more near state (resource) as is the case when $\lambda$ is increased. As the result, nodes are concentrate to fewer resources. In this sense, such environmental fluctuations would change node concentration in state assignments by the MCMC-based ADC.

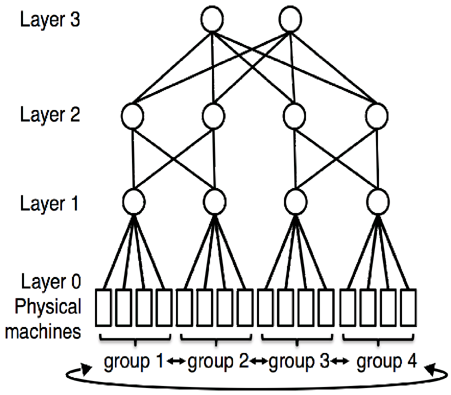


Figure 2: DCN topology

To retain the node concentration, each node should absorb the impact of environmental fluctuations. According to the above discussion, node $i$ should increase/decrease control parameter $\lambda$ used in Eq. (2) so as to cancel the increasing/decreasing amount in $\Delta M_i(x_i \to x_i')$ of node $i$ when a environmental fluctuation occurs. Namely, node $i$ adjusts own control parameter $\lambda_i$ by using

$$\lambda_i = \frac{K}{\sum_{j \in \chi_i} r_{ij}}, \qquad (5)$$

where $K$ is the control parameter of the adjustment method, and is used to provide direction for node concentration in state assignments.

## 4. Simulation Experiment

### 4.1. Experiment Model

In this section, similarly to [6], we apply the proposed method to the virtual machine placement problem in DCN (Data Center Networks). In this application, a node and a resource (a node state) correspond to a VM (Virtual Machine) and a PM (Physical Machine) used by a node, respectively. Then, $r_{ij}$ and $d(k,l)$ are the traffic rate between VMs $i$ and $j$ and the communication cost (i.e., the sum of communication costs in the shortest path) between PMs $k$ and $l$, respectively.

In experiment, we use a network topology shown in Fig. 2. In this network topology, PMs are placed in layer 0, and network equipments (e.g., network switch and router) are placed on the other layers. For scalability reason, PMs are divided into four groups, and the state transition of a VM in one time will be permitted only between PMs within the same or adjacency groups that are connected by a double-headed arrow.

To evaluate the effectiveness of the proposed method for improving the robustness against traffic rate fluctuations, we compare node concentrations when using the following two traffic rate settings:

Table 1: parameter configuration

| | |
|---|---|
| number of VMs, $n$ | 200 |
| number of PMs, $N$ | 16 |
| internal cost in a PM | 0.0001 |
| link cost | 0.1 |
| simulation time | 40000 |
| high traffic rate, $r_H$ | 10 |
| low traffic rate, $r_L$ | 0.1 |
| average number of high traffic VMs, $N_H$ | $0.1 \times N$ |
| $\alpha$ | 0 |

**traffic rate setting 1** Each VM communicates with a randomly-chosen $N_H$ (average) VMs with high traffic rate $r_H$, and other VMs with low traffic rate $r_L$.

**traffic rate setting 2** A half VMs communicates with a randomly-chosen $N_H \times 2$ (average) VMs with high traffic rate $r_H$ and others with low traffic rate $r_L$. Further the other half VMs communicates with a randomly-chosen $N_H$ (average) VMs with high traffic rate $r_H$ and other VMs with low traffic rate $r_L$.

If the node concentrations for traffic rate settings 1 and 2 are the same, we can clarify the robustness of the proposed method against traffic rate fluctuations in an indirect way.

In this experiment, we use the parameter configuration shown in table 1. At the start of each simulation run, we place $n$ VMs in a randomly-chosen PM. At each simulation time unit, a VM uses the node action to determine if it should change to another PM according to the MCMC-based ADC.

In order to evaluate the node concentration in a node assignment, we define coefficient of variation $\mathrm{CV}(\boldsymbol{X})$ by

$$\mathrm{CV}(\boldsymbol{X}) = \frac{N}{n}\sqrt{\frac{1}{N}\sum_{k=1}^{N}\left\{\left(\sum_{i=1}^{n}\delta_{x_i k}\right) - \frac{n}{N}\right\}^2}, \qquad (6)$$

where $\delta_{ij}$ denotes the Kronecker delta.

### 4.2. Robustness against Fluctuation

We evaluate the effectiveness of the proposed method against traffic rate fluctuations. Figure 3 shows VM placement in each PM for different traffic rate settings when using the proposed method (i.e., the MCMC-based ADC with the adjustment method) and the previous method (i.e., the MCMC-based ADC without the adjustment method). According to Fig. 3, the previous method cannot retain the node concentration for different traffic rate settings. On the contrary, the proposed method can retain the node concentration. Therefore, we can confirm that the proposed method improves the robustness of the MCMC-based ADC for severe fluctuations in traffic rates.

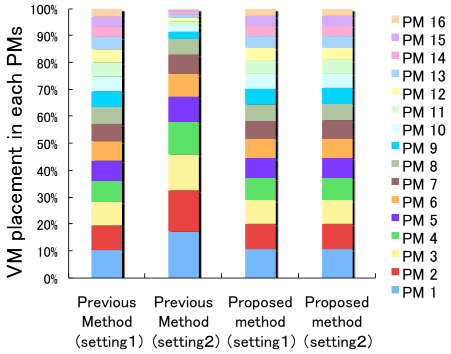


Figure 3: VM placements using traffic settings 1 and 2 in Proposed method and Previous method

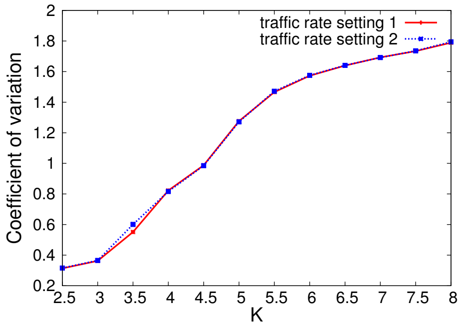


Figure 4: Coefficient of variation CV($\boldsymbol{X}$) for different values of parameter $K$ of the proposed method

Figure 4 shows coefficient of variation CV($\boldsymbol{X}$) for different values of control parameter $K$ of the proposed method. In this figure, we show the results for traffic rate settings 1 and 2. As the result shown in Fig. 4, CV($\boldsymbol{X}$) for different traffic rate settings are almost same regardless the value of control parameter $K$ of the proposed method.

From Figs. 3 and 4, the results for traffic rate settings 1 and 2 are the same. Hence, we clarify the robustness of the proposed method against traffic rate fluctuations in an indirect way.

## 5. Conclusion and Future Work

In this paper, we discussed the impact of environmental fluctuation on the MCMC-based ADC, and proposed a method to enhance the robustness of the MCMC-based ADC against severe environmental fluctuations. In particular, we designed an adjustment method to absorb the impact of environmental fluctuations according to the outcome of our discussion. In the proposed method, each node autonomously adjusts its control parameter of the MCMC-based ADC. In addition, similarly to [6], we applied the proposed method to the virtual machine placement problem in data centers. We performed the simulation experiment using the application, and clarified the effectiveness of the proposed method for severe environmental fluctuation.

As future work, we are planning evaluate the proposed method against several environmental fluctuations (e.g., the change in the number of nodes, and state distance).


## Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 24700072.


## References


[1] "2014 WHITE PAPER" http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2014/2014-index.html

[2] "Google Date Centers. " http://www.google.com/about/datacenters/inside/locations/index.html.

[3] M. Aida, "Using a renormalization group to create ideal hierarchical network architecture with time scale dependency," IEICE Transactions on Communications, vol. E95-B, no. 5, May 2012.

[4] V. A. Cicirello and S. F. Smith, "Ant colony control for autonomous decentralized shop floor routing," In Proceedings of ISADS 2001, pp. 383-390, Mar. 2001.

[5] K. Leibnitz, N. Wakamiya and M. Murata, "Biologically inspired self-adaptive multi-path routing in overlay networks," Communications of the ACM, vol 49, no. 3, pp. 62-67, 2006.

[6] Y. Sakumoto, M. Aida and H. Shimonishi, "An autonomous decentralized control for indirectly controlling system performance variable in large-scale and wide-area network," in Proceedings of NETWORKS 2014, pp. 1-7, Sep. 2014.

[7] Y. Sakumoto, M. Aida and H. Shimonishi, "An autonomous decentralized adaptive function for retaining control strength in large-scale and wide-area system," in Proceedings of GLOBECOM 2014, pp. 1958-1964, Dec. 2014.

[8] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," Biometrika, vol. 57, no. 1, pp. 97–109, 1970.

[9] W.R. Gilks, S. Richardson and D. J. Spiegelhalter, "MarkovchainMonte Carlo in practice, " vol. 2. CRC press, 1996.