2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Control of Selfish Routing Based on Replicator Dynamics with State-Dependent Tax

Takurou Misaka†, Takafumi Kanazawa†, and Toshimitsu Ushio†

†Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Email: misaka@hopf.sys.es.osaka-u.ac.jp

Abstract—Selfish behaviors of users can increase average data transmission latency in computer networks. A selfish routing game is a simple model of the selfish behaviors and its replicator dynamics has been proposed. To reduce the inefficiencies due to the selfish behaviors, a control method to impose latency on each transmission path as a state-dependent tax has been proposed. Several properties of the control method have been discussed using the model based on the replicator dynamics. In this paper, we propose a routing algorithm based on replicator dynamics modeling an effect of the state-dependent tax. Moreover, we show the effectiveness of the proposed algorithm using a network simulator.

1. Introduction

In large computer networks, there exist several inefficiencies caused by selfish behaviors [1], [3]. To resolve such inefficiencies, a mechanism for packet control is needed. Recently, several game theoretical methodologies have been paid much attention to model and resolve such a problem [1]. A selfish routing game is a simple model of selfish behaviors in networks [5]. Its replicator dynamics has also been proposed [4], [6].

We consider a single-commodity network with a source and a sink. We suppose that there is a fixed flow demand from the source to the sink, and users select paths so as to minimize their own latencies. Such selfish behaviors cause several inefficiencies [1]. Braess's paradox is widely known as an example of such inefficiencies [2]. In Braess's paradox, an equilibrium flow achieved by selfish behaviors differs from the minimum latency flow.

On the other hand, inefficiencies due to players' selfish behaviors in social systems have also been studied. To model and resolve conflicts between a payoff of each player and the total payoff of players, replicator dynamics with a subsidy and a capitation tax has been introduced [8]. Its application to the selfish routing game has also been studied [9]. In this model, an additional latency is imposed on each transmission path as a state-dependent tax to control the selfish routing. Several properties of the model have been shown.

In this paper, we consider a stabilization problem of the target state. We propose a routing algorithm based on replicator dynamics modeling an effect of the state-dependent tax. The latency of the target state is unaffected by the state-dependent tax in the proposed algorithm. Using a graph, we show that the target state can be stabilized by the proposed state-dependent tax.

2. Preliminaries

2.1. Selfish Routing

We consider a single-commodity network $G = (V, E)$ with a source vertex s and a sink vertex t, where V and E are sets of vertices and edges, respectively. Suppose that there is a fixed flow demand and it is routed from the source s to the sink t. Let P be the set of paths s-t. A flow $x = (x_{p_1}, \ldots, x_{p_n})^T$ is a nonnegative vector, where x_{p_i} is the amount of flow routed over the path $p_i \in P$ and n is the number of elements in P. For a flow vector x, we define the flow on an edge $e \in E$ by $x_e = \sum_{p_i \ni e} x_{p_i}$. Let $l_e(x_e)$ be the latency on the edge $e \in E$ and we assume that it is a nonnegative, continuous, and nondecreasing function. The latency $l_{p_i}(x)$ of $p_i \in P$ is given by $l_{p_i}(x) = \sum_{e \in p_i} l_e(x_e)$, and the average latency $\bar{l}(x)$ of a flow vector x is given by $\bar{l}(x) = \sum_{p_i \in P} x_{p_i} l_{p_i}(x)$. For simplicity, we assume the total amount of the fixed flow demand is equal to 1, that is, $\sum_{p_i \in P} x_{p_i} = 1$.

Suppose that users select paths in P in order to minimize the latency. A selfish routing game is a simple model of such a selfish behavior in networks [1]. Its replicator dynamics has been proposed as follows [4], [6]: for all $p_i \in P$,

$$\dot{x}_{p_i} = x_{p_i}(\bar{l}(x) - l_{p_i}(x)). \tag{1}$$

In the selfish routing game, a flow vector x is said to be a Nash flow if $l_{p_i}(x) \leq l_{p_j}(x)$ holds for every pair of paths p_i and p_j with $x_{p_i} > 0$ [3].

In the above definition of the selfish routing game, a packet routing mechanism is not given explicitly. So, we propose Algorithm 1 as a selfish routing algorithm based on replicator dynamics. In this algorithm, we assume that the end-to-end latency of each path is observed by the sink and the source receives it as a message. Based on the received message, the source increases flows routed over paths with lower latency than the average and decreases flows routed

over paths with higher latency than the average within min and max.

Algorithm 1 Selfish Routing Algorithm

▷ n is the number of paths from s to t.
▷ ϵ is the change ratio of flow.
▷ s is the packet size.
▷ d is the flow demand.
▷ x_{p_i} is a current weight of path i.
▷ $l_{p_i}(x)$ is a end-to-end latency of path i.

$max \leftarrow \frac{d+(1-n)s}{d}$
$min \leftarrow \frac{s}{d}$
$\bar{l}(x) \leftarrow \sum_{i=1}^{n} x_{p_i} l_{p_i}(x)$
for $i = 1$ to $n-1$ **do**
  $x_{p_i} \leftarrow x_{p_i} + \epsilon x_{p_i}(\bar{l}(x) - l_{p_i}(x))$
  if $x_{p_i} > max$ **then**
    $x_{p_i} \leftarrow max$
  end if
  if $x_{p_i} < min$ **then**
    $x_{p_i} \leftarrow min$
  end if
end for
$x_{p_n} \leftarrow 1 - \sum_{i=1}^{n-1} x_{p_i}$

2.2. Braess's Paradox

In the selfish routing game, there exist several inefficiencies due to selfish route selections of users. *Braess's paradox* is one of the well-known examples in such inefficiencies [7]. Intuition may suggest that an additional edge with zero or sufficiently low latency reduces the average latency of the flow in the network, but it is incorrect in some cases and the average latency may increase due to players' selfishness. That is called Braess's paradox.

3. Control of Selfish Routing

3.1. Replicator Dynamics with A Subsidy and A Capitation Tax

To reduce inefficiencies in social systems, Kanazawa *et al*. have proposed replicator dynamics with a subsidy and a capitation tax [8]. Its application to the selfish routing game has also been studied [9]. Let x^* be the target state. In this model, we stabilize the target state imposing additional delay $c(x)$ on each path as the capitation tax and removing delay $\beta x^*_{p_i}/x_{p_i}$ from path p_i as the subsidy where β is the total amount of subsidies. Since $c(x) - \beta x^*_{p_i}/x_{p_i}$ should be nonnegative and as small as possible, we set $c(x) = \max_{p_i \in P}\{\beta x^*_{p_i}/x_{p_i}\}$. Thus, the latency of each path $p_i \in P$ with the subsidy and the capitation tax is given by

$$l_{p_i}(x) + \max_{p_i \in P}\left\{\beta \frac{x^*_{p_i}}{x_{p_i}}\right\} - \beta \frac{x^*_{p_i}}{x_{p_i}}. \qquad (2)$$

Suppose that every path is used for the packet transmission at least in the initial state, that is, $x_{p_i}(0) > 0$ is assumed for any path $p_i \in P$. By this assumption, within any finite-time interval, $x_{p_i} > 0$ holds for all $p_i \in P$. Since Eq. (2) is always greater than or equal to $l_{p_i}(x)$ for all $p_i \in P$ with equality if and only if $x = x^*$, it is well-defined as a latency function in the selfish routing game and the latency of the target state is unaffected by the additional latency. Thus, replicator dynamics of the selfish routing with a state-dependent tax is given by

$$\dot{x}_{p_i} = x_{p_i}(\bar{l}(x) - l_{p_i}(x)) + \beta(x^*_{p_i} - x_{p_i}). \qquad (3)$$

Let the linearization system of Eq. (1) at $x = x^*$ be $\dot{x}_p = J_0 x$, and be $\lambda_{0i}, i = 1, ..., n$ as eigenvalues of J_0, where n is the total number of paths. It has been proved that an equilibrium point of Eq. (1) is also an equilibrium point of Eq. (3), and the target state $x = x^*$ of Eq. (3) is locally asymptotically stable if $\beta > \max_i\{\Re(\lambda_{0i})\}$, where $\Re(\lambda_{0i})$ is the real part of λ_{0i} [8].

3.2. Control Algorithm

In this paper, we assume that the source selects paths based on the received latency from the sink. The sink observes the latency of each path and sends it as a message periodically with a specified period. Therefore, the sink can control the source routing by sending fake messages which informs a latency differs from the observed one. To stabilize the target flow, we consider that the sink informs the latency added the state-dependent tax defined by Eq. (2) to the source as an observed latency. We propose Algorithm 2 as an implementation of the state-dependent taxation. It calculates the latency which is imposed the state-dependent tax on and the sink informs it to the source.

Algorithm 2 SDT (State-Dependent Taxation)

▷ n is the number of paths from s to t.
▷ β is the total subsidy.
▷ x_{p_i} is a current weight of path i.
▷ $x^*_{p_i}$ is a desirable weight of path i.
▷ $l_{p_i}(x)$ is a end-to-end latency of path i.

$maxsub \leftarrow 0$
for $i = 1$ to n **do**
  if $\frac{\beta x^*_{p_i}}{x_{p_i}} > maxsub$ **then**
    $maxsub \leftarrow \frac{\beta x^*_{p_i}}{x_{p_i}}$
  end if
end for
for $i = 1$ to n **do**
  $l_{p_i}(x) \leftarrow l_{p_i}(x) + maxsub - \frac{\beta x^*_{p_i}}{x_{p_i}}$
end for

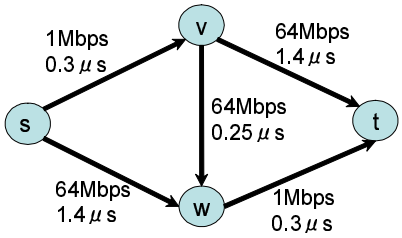

Figure 1: A simple example of networks.

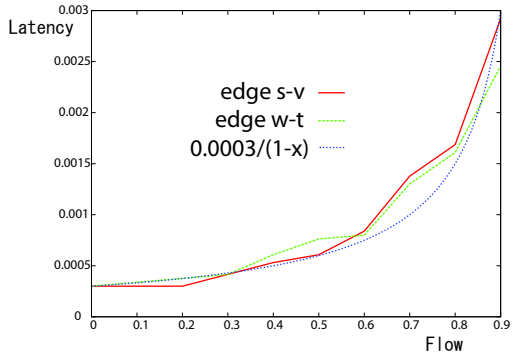

Figure 2: Latencies of s-v and w-t.

4. Simulation

4.1. Network Topology

In this section, we consider a graph shown in Fig. 1 which exhibits a Braess's paradox. Suppose that a user sends UDP packets from s to t. The flow demand is 1Mb and the packet size is 10 byte. The bandwidth of links s-v and w-t is 1Mbps and that of s-w, v-w, and v-t is 64Mbps. So, s-v and w-t may be bottleneck links. The propagation delay of links s-v and w-t is 0.3 μs, that of s-w and v-t is 1.4 μs, and that of v-w is 0.25 μs. We set the queue size of each edge to 100.

We identify link latency functions of the graph shown in Fig. 1. We employ an M/M/1 model as the latency function. Figure 2 shows relationships between the flow and the latency on edges s-v and w-t, and the indentification of the latency function of them. As shown in Fig. 2, the latency function of s-v and w-t is given by $l_e(x) \simeq \frac{0.3}{1-x}(\mu s)$, that of v-w is $l_e(x) \simeq 0.25(\mu s)$, and that of s-w and v-t is $l_e(x) \simeq 1.4(\mu s)$.

In this section, suppose that the sink observes the latency of each path every 0.1 seconds. Using the average of the observed latencies in 2.0 seconds, Algorithm 2 is executed, the latency is informed to the source, and Algorithm 1 is executed every 2.0 seconds. Let paths 1, 2, and 3 correspond to s-v-t, s-v-w-t, and s-w-t, respectively. In this case, we have

$$\dot{x}_{p_i} = x_{p_i}(\bar{l}(x) - l_{p_i}(x)) + \beta(x^*_{p_i} - x_{p_i})(i = 1, 2), \quad (4)$$

$$l_{p_1}(x) = 10^{-3}\left(\frac{0.3}{1 - x_{p_1} - x_{p_2}} + 1.4\right),$$

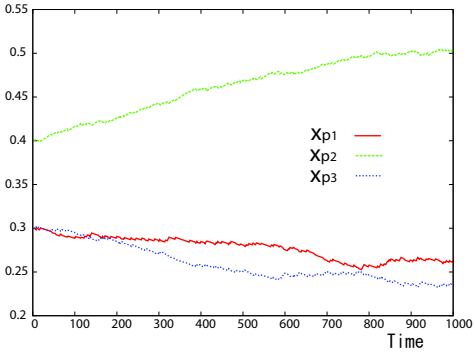

Figure 3: Simulation result ($\beta = 0$).

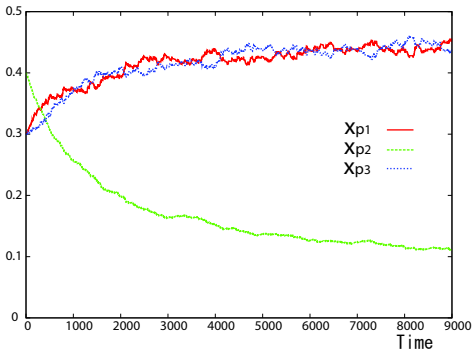

Figure 4: Simulation result ($\beta = 0.56$).

$$l_{p_2}(x) = 10^{-3}\left(\frac{0.3}{1 - x_{p_1} - x_{p_2}} + 0.25 + \frac{0.3}{x_{p_1}}\right),$$

$$\bar{l}(x) = x_{p_1} l_{p_1}(x) + x_{p_2} l_{p_2}(x) + (1 - x_{p_1})\left(\frac{0.3}{x_{p_1}} + 1.4\right),$$

as replicator dynamics of the selfish routing in the graph shown in Fig. 1.

4.2. Results and Discussions

We set the minimum latency flow $(1/2, 0, 1/2)^T$ to the target state x^*. From Sect. 2.1, the target state of Eq. 4 is stabilized with $\beta > 0.55$.

Figure 3 shows a simulation result without the tax ($\beta =$

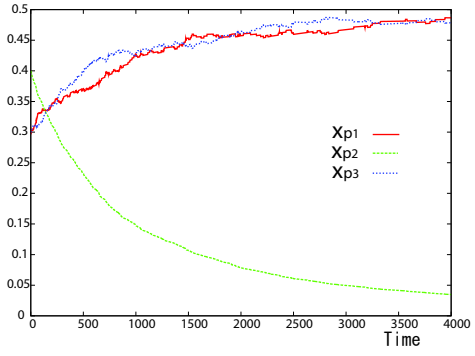

Figure 5: Simulation result ($\beta = 1.0$).

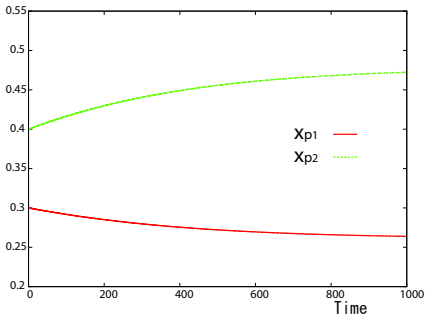

Figure 6: Transient behavior of Eq. (4) for $\beta = 0$.

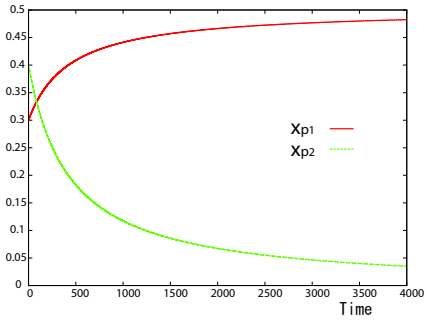

Figure 7: Transient behavior of Eq. (4) for $\beta = 0.56$.

0), where the horizontal axis is time, and the vertical axis is the amount of flow routed over each path. As shown in Fig. 3, the flow of path 2 increases and the flows of paths 1 and 3 decrease. Figure 4 shows a simulation result with state-dependent tax ($\beta = 0.56$). The target state in Eq. 4 is stabilized for $\beta = 0.56$. As shown in Fig. 4, the flows of paths 1 and 3 increases and the flow of paths 2 decrease. However, the target state isn't stabilized. Figure 5 shows simulation result with state-dependent tax ($\beta = 1.0$), and in this case, the target state is stabilized. From these simulation results, we find that β may be larger than the theoretical value to stabilize the target state. This may be due to discretization of replicator dynamics, noises, and errors in identifying latency functions. Figures 6 and 7 show transient behavior of Eq. (4) for $\beta = 0$ and $\beta = 0.56$. As shown in these figures, the stable states of Figs. 3 and 5 are

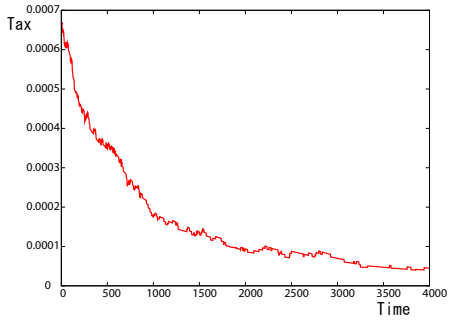

Figure 8: Transient behavior of the state-dependent tax ($\beta = 1.0$).

almost same as those of Figs. 6 and 7. Figure 8 shows the transient behavior of the state-dependent tax for $\beta = 1.0$, where the horizontal axis is time, and the vertical axis is the value of the state-dependent tax. As shown in Fig. 8, the state-dependent tax converges near the origin. In the steady state, the average latency for $\beta = 0$ is 2.6(μs) and the average latency for $\beta = 1.0$ is 1.7(μs). Therefore, the price of anarchy is $\frac{2.6}{1.7} \simeq 1.53$ and we can resolve the paradox.

5. Conclusions

In this paper, we proposed a routing algorithm based on a selfish routing and we also proposed a flow control algorithm based on a state-dependent tax. Moreover, we showed the effectiveness of the proposed algorithm using a network simulator.

Our future work is to clarify the reason why β must be larger than the theoretical value to achieve the stabilization and extend our work to multiple source-sink network.

Acknowledgments

This research was supported in part by KAKENHI (No. 21760330) and "Global COE Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] T. Roughgarden, *Selfish Routing and the Price of Anarchy*, MIT Press, 2005.

[2] D. Braess, "Über ein paradoxen aus der verkehrsplanung," *Unternehmensforschung*, vol.12, pp.258–268, 1968.

[3] T. Roughgarden and É. Tardos, "How bad is selfish routing?," *Journal of the ACM*, vol.49, no.2, pp.236–259, 2002.

[4] S. Fischer and B. Vöcking, "On the evolution of selfish routing," *Proceedings of 12th Annual European Symposium on Algorithms (ESA)*, pp.323–334, 2004.

[5] J. G. Wardrop, "Theoretical Aspects of Road Traffic Research," *Proceedings of the Institute of Civil Engineers*, no.2, pp.325–378, 1952.

[6] J. W. Weibull, *Evolutionary Game Theory*, MIT Press, 1995.

[7] R. Cole *et al.*, "How Much Can Taxes Help Selfish Routing?," *Journal of Computer and System Sciences*, vol.72, no.3, 2006.

[8] T. Kanazawa *et al.*, "Replicator Dynamics with Pigovian Subsidy and Capitation Tax," *Nonlinear Analysis: Theory, Methods and Applications*, doi:10.1016/j.na.2008.11.072 (in press).

[9] T. Kanazawa *et al.*, "A Control Method of Selfish Routing Based on Replicator Dynamics with Capitation Tax and Subsidy," *Proceedings of 3rd IEEE Multiconference on Systems and Control*, pp.249-254, 2009.