NOLTA, IEICE

Paper

A simplistic dynamic circuit analogue of adaptive transport networks in true slime mold

Hisa-Aki Tanaka [1,2a)], *Kazuki Nakada* [2], *Yuta Kondo* [1], *Tomoyuki Morikawa* [1], and *Isao Nishikawa* [2]

[1] *Graduate School of Information Systems,*
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu 182-8585, Japan

[2] *Graduate School of Informatics and Engineering,*
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu 182-8585, Japan

a) *htanaka@uec.ac.jp*

Received June 6, 2015; Revised October 28, 2015; Published April 1, 2016

Abstract: This paper presents a simplistic dynamic circuit analogue for an adaptive transport network model in true slime mold by Tero et al. This circuit analogue model is derived from Tero's model through nontrivial simplification under certain assumptions, and it realizes less computational complexity through a reduction of the number of variables. Despite of its simplicity, systematic simulations confirm that the shortest path search task is efficiently accomplished with this model; (i) the shortest path is always identified, for various random networks; (ii) if there are multiple, competing shortest paths in the network, they are simultaneously identified; and (iii) for random deletions of a link in the shortest path, a new shortest path is quickly identified accordingly. The model proposed here is easily implemented on the circuit simulator SPICE for instance, and hence the path search time will be further reduced with certain numerical devices including automatic adaptive numerical integration schemes as well as an acceleration method proposed in the end of the paper.

Key Words: biology-inspired algorithm, shortest path search methods, nonlinear resistive circuits

1. Introduction

A pioneering study by Tero et al. [1, 2] has recently uncovered a new shortest path search method, motivated by observation of the amoeba-like organism (plasmodium) of the true slime mold *Physarum polycephalum* [3] (hereafter abbreviated to *Physarum*). They showed that their method (*i.e.*, Tero's model) identifies the shortest path and, according to systematic numerical simulations and analysis, this ability is robust to temporal changes in the network topology [1, 2]. Later on, some of their results

DOI: 10.1587/nolta.7.86

are mathematically proved by Miyaji and Ohnishi [4], where a kind of 'local' Lyapunov function is cleverly constructed for planar graphs. This result in [4] enabled further analytical investigations. For instance, convergence to shortest paths are proved for all graphs [5], and for a time-discretization of the Tero's model [6].

Incidentally, Aono's group has explored a more difficult combinatorial optimization; the traveling salesman problem (TSP), using *Physarum* [7]. Their systematic experiments showed that *Physarum* finds a high quality solution for the (8-city) TSP with a high probability. Thus, *Physarum*-inspired algorithms and their physical implementation are expected to explore new, primitive but sophisticated computing machinery.

In this paper, we propose a nontrivial circuit analogue of Tero's model [1, 2], which is directly implemented on circuit simulators such as SPICE. From systematic investigations of this circuit, we numerically verify the following patterns for all the instances considered here[1]: (i) the shortest path is always identified, (ii) multiple, competing shortest paths are simultaneously identified, and (iii) a new shortest path is identified immediately after a link is lost in the shortest path. Also, the required times for this shortest path search method are systematically investigated. As far as we know, such properties by a simplistic shortest path search algorithm have never been reported before.

Some results in this paper were presented in NOLTA'07 [8]. The present paper carefully reexamines all the simulations and further investigates the above item (iii) of the presented method, and hence it adds new and original contents. The remainder of this paper is organized as follows. In Section 2, we briefly review the mathematical model by Tero et al. [1, 2] and construct a circuit analogue for it. In Section 3 we investigate the path search time and the path search process in the proposed circuits. Then, the discussion and conclusions are presented in Section 4.

2. Mathematical model for transport network in *Physarum polycephalum* and its dynamic circuit analogue

Systematic experiments demonstrate that adaptive transport networks in the true slime mold *Physarum* have a shortest path search ability [3]. Motivated by this observation, Tero et al. [1, 2] constructed a simple mathematical model for adaptive transport networks in *Physarum* (Tero's model, below). They numerically and analytically investigated how their model reaches the shortest path. In this section, we review Tero's model and construct a nontrivial circuit analogue for it. Implementation of this circuit on SPICE is also considered.

2.1 Mathematical model for transport networks in *Physarum*

We start by reviewing the mathematical model for adaptive transport networks in *Physarum* per [1, 2]. This living network initially contains numerous thin 'tubes' filled with nutrient-transporting liquid that sense and respond to the environment. If we set two 'foods' on N_1 and N_2 in the network, as shown in Fig. 1(a), *Physarum* tends to connect N_1 and N_2 with a shortest path of tubes by adaptively growing (or degrading) the thickness of these tubes [3]. The dynamics of network adaptation in *Physarum* [1, 2] is modeled by experimental results and physiological insights at the molecular level

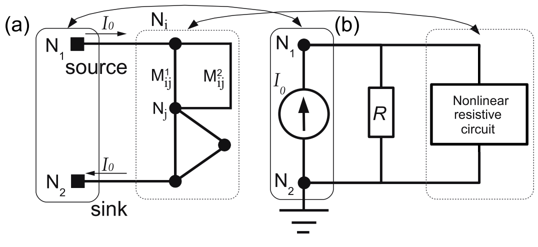

Fig. 1. Network architecture in (a) adaptive transport networks [1, 2], and (b) time-dependent, nonlinear resistive circuits in the proposed method.

[1] We have used 60 instances of randomly generated network with various sizes for shortest path search. See Section 3.1 for more details.

(for detail, see [2]), which are summarized as follows.

First, the tube connecting nodes N_i and N_j are denoted as M_{ij}. If there are multiple tubes connecting the same pair of nodes, we distinguish these tubes as M_{ij}^1 and M_{ij}^2, as shown in Fig. 1(a). The total numbers of tubes and nodes are M and N, respectively. Then we consider the flux Q_{ij}, which passes through M_{ij} from N_i to N_j. As Poiseuille flow is assumed in the tube, the flux $Q_{ij}(t)$ is given by

$$Q_{ij}(t) = \frac{D_{ij}(t)}{L_{ij}}[p_i(t) - p_j(t)], \tag{1}$$

where $p_i(t)$ represents the pressure at N_i, L_{ij} represents the length of M_{ij}, and $D_{ij}(t)$ is the conductivity of M_{ij}. As the total amount of liquid is conserved in the network, Kirchhoff's law holds at each node except at nodes with 'foods':

$$\sum_{i=1}^{N} Q_{ij} = 0, \ (j \neq 1, 2). \tag{2}$$

Assuming N_1 and N_2 act respectively as the source and the sink, the following two equations hold:

$$\sum_{i=1}^{N} Q_{i1} + I_0 = 0, \ \sum_{i=1}^{N} Q_{i2} - I_0 = 0, \tag{3}$$

where I_0 is the flux from the source node and Q_{11} and Q_{22} are assumed to be 0. In accordance with Refs. [1, 2], we set this I_0 to be a positive constant.

To model the adaptation of tubular thickness to the flux Q_{ij}, the conductivity D_{ij} is assumed to change in time as

$$\frac{d}{dt}D_{ij} = f(|Q_{ij}|) - rD_{ij}, \tag{4}$$

where $f(Q)$ is a certain increasing function with $f(0) = 0$, and r is a positive constant. From this equation, the conductivity D_{ij} decreases to 0 by itself, but it increases when a certain amount of flux Q_{ij} is retained in the tube M_{ij}. For simplicity, we assume the function f as $f(|Q_{ij}|) = \alpha|Q_{ij}|$, where α is a positive constant. Then, the equation for D_{ij} is simplified as

$$\frac{d}{dt}D_{ij} = \alpha|Q_{ij}| - rD_{ij}. \tag{5}$$

Thus, $D_{ij}(t)$, $Q_{ij}(t)$, $p_i(t)$, and $p_j(t)$ at time t are determined by Eqs. (1), (2), (3), and (5). More precisely, Eqs. (1), (2), and (3) contain $M + N$ linear equations with $M + N$ variables (Q_{ij} and p_i) and Eq. (5) contains M nonlinear equations with M variables (D_{ij}), which are well-defined. Then, $D_{ij}(t)$ of l.h.s. in Eq. (5), a function of $D_{ij}(t)$, $Q_{ij}(t)$, and $p_i(t)$, is obtained by Eq. (5).

Note that as p_i appears in the form of $p_i - p_j$ in Eq. (1), there remains an arbitrariness in the value of p_i. To remove this arbitrariness, we assume $p_2(t) \equiv 0$ naturally. This assumption is reasonable because the pressure p_2 always becomes 0 at the sink node N_2, as shown in the liquid transport network of Fig. 1(a). In addition, α and r in Eq. (5) are set to 1 without loss of generality after rescaling p_i and t.

2.2 Dynamic circuit analogue of the network

A circuit analogue for Tero's model is constructed as in Fig. 1(b). First, we regard the flux Q_{ij} as the current I_{ij}, the pressure p_i as the voltage V_i, and D_{ij}^{-1} as the unit resistance R_{ij}. The change of variables is: $p_i \equiv V_i$, $D_{ij}^{-1} \equiv R_{ij} = V_{ij}/I_{ij}$, and $Q_{ij} = (D_{ij}/L_{ij})V_{ij} = I_{ij}/L_{ij}$, in which $V_{ij} \equiv V_i - V_j$. Then, the equivalent equations to (1), (2), (3), and (5) are obtained for $I_{ij}(t)$, $V_{ij}(t)$, and L_{ij}. The circuit analogue of Eqs. (1), (2), and (3) is clear and trivial, since these equations represent Ohm's law and Kirchhoff's law. On the contrary, the circuit analogue of Eq. (5) is nontrivial, because it is not clear how to implement an Eq. (5) analogue with simple circuit elements. However, if we assume that the time evolution of V_{ij} is relatively slow, compared with that of I_{ij}, a reduction becomes possible as follows.

First, from Eqs. (1) and (5), the following equation is directly obtained after setting α and r to unity:

$$L_{ij}\frac{d}{dt}\left(\frac{I_{ij}}{V_{ij}}\right) = |I_{ij}| - L_{ij}\frac{I_{ij}}{V_{ij}}, \tag{6}$$

where L_{ij} is a positive constant. If V_{ij} evolves much slower than I_{ij} (*i.e.*, Assumption 1; the validity of this assumption is numerically verified later in Section 3.2), then the following is valid for a certain (infinitesimally) short time span,

$$\frac{d}{dt}I_{ij}(t) = \frac{V_{ij}}{L_{ij}}|I_{ij}(t)| - I_{ij}(t), \tag{7}$$

where V_{ij} is safely approximated as a constant parameter and its values is determined with $I_{ij}(t)$ from Ohm's law and Kirchhoff's law (*i.e.*, Eqs. (1), (2), and (3)) at each time span. Then, the solution of I_{ij} is given by

$$I_{ij}(t) = I_{ij}(0)\exp\Big[\Big(\frac{V_{ij}}{L_{ij}} - 1\Big)\, t\Big] > 0, \tag{8}$$

or

$$I_{ij}(t) = I_{ij}(0)\exp\Big[\Big(\frac{-V_{ij}}{L_{ij}} - 1\Big)\, t\Big] < 0, \tag{9}$$

respectively for a positive or negative initial value of $I_{ij}(0)$. Now the variable $D_{ij}(t)$ in Eq. (5) is eliminated, and Eq. (5) is reduced to Eqs. (8) and (9) via Eq. (6). This reduction makes our path search process computationally more effective, since the number of variables (I_{ij} and V_i) is now $M+N$. This $M+N$ is reduced from the original $2M+N$ variables (Q_{ij}, D_{ij}, and p_i), and the reduced number M $(= 2M + N - M - N)$ is often huge when the links are dense.

Here we note, in identifying the unknown shortest path in the network, the final direction of the current on each link is not given in advance. This requires some modifications for Eqs. (8) and (9). A simple but nontrivial construction is obtained by superimposing both currents Eqs. (8) and (9):

$$I_{ij} = I_0\exp\Big[\Big(\frac{V_{ij}}{L_{ij}} - 1\Big)\, t\Big] - I_0\exp\Big[\Big(\frac{-V_{ij}}{L_{ij}} - 1\Big)\, t\Big], \tag{10}$$

in which I_0 and $-I_0$ respectively correspond to $I_{ij}(0)$ in Eqs. (8) and (9). In this modification, both directions of currents are implicitly assumed in each link, which tolerates the uncertainty of the final currents directions. In conjunction with this uncertainty, I_{ij} in Eq. (10) becomes 0 at $t = 0$ since we have set $I_{ij}(0)$ in Eqs. (8) and (9) as I_0 and $-I_0$ respectively. Therefore, this time-dependent, nonlinear voltage-dependent current source I_{ij} is well-defined for $t \geq 0$, and can be stably simulated. The identified shortest path is characterized as follows. In Eq. (10), I_{ij} converges to I_0 or $-I_0$ if and only if V_{ij} converges to L_{ij} or $-L_{ij}$ respectively[2], and otherwise I_{ij} dies out to 0. This implies all $|I_{ij}| \to I_0$ on the shortest path, which is consistent to Eqs. (2), (3) (Kirchhoff's law) and it is verified in all simulations in this study.

Even though the above construction is somewhat heuristic, the shortest paths (and the second shortest paths) for all 60 cases given in Section 3.1 are successfully obtained in this circuit, with SPICE[3]. This result and the underlying mechanism will be investigated in detail in the next section.

3. Path search process and elapsed times

Thus far, we have derived a dynamic circuit analogue from Tero's model [1, 2]. In simulating this circuit we set a constant current source ($\sim$ 10 A) between the nodes N_1 and N_2, and we connect a large resistance R ($\sim$ 100 kΩ) in parallel with the nonlinear current sources of Eq. (10), as shown in Fig. 1(b). The reason why this resistance R is introduced is understood as follows. Firstly, at $t = 0$ there is no current on every link because I_{ij} becomes 0 at $t = 0$. However, as we assumed a

[2] In Eq. (10) the speed of convergence in V_{ij} is fast enough, compared with the increasing speed of t, which is not yet proved but verified in all simulations in this study.

[3] An illustrative example on SPICE (netlist) is available from the following site: http://synchro3.ee.uec.ac.jp/netlist2015.pdf, which generates the data shown in Fig. 5(a).

constant current source between N_1 and N_2, there must be a bypass (a constant resistance) in parallel. Secondly, it is desirable that all the current should pass through only the shortest path eventually. So this constant resistance should be as large as possible. The most widely used circuit simulator SPICE is used for all simulations in this study, as its numerical results are reliable and they can be easily traced by many researchers.

3.1 Elapsed times for shortest path search

As we shall see in Section 3.2, typical path search processes consist of two stages: an early fast-evolving transient process and a final process slowly converging to an shortest path (which is shown later in Fig. 3(a) for a typical case). Thus, the shortest path is obtained immediately after this fast evolving stage in practice, which is judged by changing rates of the currents being less than a threshold ($\sim 1.0 \times 10^{-3}$ A/s for instance).

To investigate the averaged elapsed times required for the shortest path search, we have systematically generated random networks with N nodes and M $(= 4N)$ links for $N = 2^5, 2^6, 2^7, 2^8, 2^9$, and 2^{10}, respectively[4]. For each N, 10 networks are randomly generated and our shortest path search method is applied to each network. Also, Dijkstra's algorithm for finding the shortest path [10] is applied to the same networks to verify the advantage and the correctness of our path search results.

Figure 2 shows the elapsed times for our proposed method, where a comparison is made among elapsed times for simulations based on Dijkstra's algorithm (plotted with □) on a PC (Dell Dimension 8300, Pentium 4 3.2 GHz CPU), elapsed times on (virtual) circuits (⋄) on the SPICE simulator, and their averages (×). By comparing these two elapsed times, we can clarify the essential physics behind this path search process, which is not influenced by simulation details. Namely, we observe the following patterns;

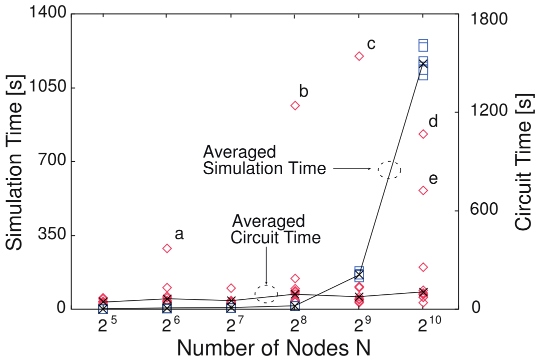

Fig. 2. Elapsed times for the proposed method. □ (blue) and ⋄ (red) respectively represent the elapsed time for numerical simulations on PC (Simulation Time) and the time for virtual circuits on the SPICE simulator (Circuit Time). × represents the average of both times for a given number of nodes N. Data sets 'a,' 'b,' 'c,' 'd,' and 'e' are exceptional slowly converging cases, as explained in the main text.

(i) for all cases except for the five exceptional slowly converging cases 'a,' 'b,' 'c,' 'd,' and 'e,' the circuit finds the shortest path within around 100 s, irrespective of the number of nodes N, and

(ii) although the simulation times clearly show an increasing tendency (partly reflecting the time complexity of the numerical integration in the simulator), the average elapsed times (×) for the circuit do not apparently show such a tendency with respect to N. Nevertheless, the exceptional cases 'a,' 'b,' 'c,' 'd,' and 'e' deviate from the averaged circuit time significantly.

3.2 Shortest path search process

To investigate the mechanism behind the above observations (i) and (ii), we have numerically analyzed the time courses of path search processes in detail, by setting I_0 in Eq. (10) to 10 A. Figures 3(a)

[4] Those random networks are downloaded from the DIMACS challenge site [9], which is widely used by researchers of combinatorial algorithms and its validity has been tested by them.

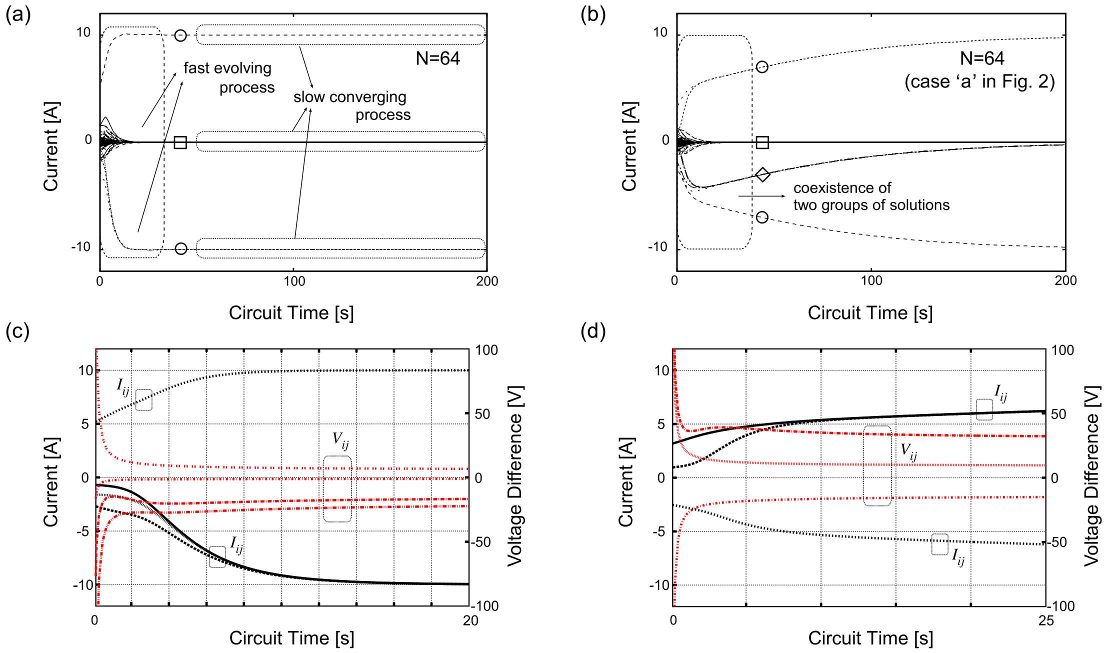

Fig. 3. Two typical path search processes in the proposed method. Horizontal axes (Circuit Time) represent the elapsed time in the circuits. I_0 is set to 10 A. (a) Fast converging case. ○ and □ represent $I_{ij}(t)$ for the shortest path and the other paths respectively. (b) Slowly converging case. ○, ◇, and □ respectively represent $I_{ij}(t)$ for the shortest path, the second shortest path, and the other paths. (c) Time evolution of the shortest path solution V_{ij} (red) and I_{ij} (black) for the fast converging case (a). (d) Time evolution of the shortest path solution V_{ij} (red) and I_{ij} (black) for the slowly converging case (b). [Note that right after $t = 0$ some V_{ij} become too large to be included in Figs. 3(c), (d), since all current (10 A) passes through the bypass resistance R (100 kΩ).]

and (b) respectively shows two typical examples from a fast converging case and a slowly converging case (*i.e.*, the data point ‘a’ in Fig. 2). In both figures, we plotted all the currents in the network, but they quickly coalesced into some groups.

As we observe in Fig. 3(a), in a typical instance of $N = 2^6 = 64$, the shortest path (denoted by ○) can be already identified at around $t = 20$ s, while other paths (□) quickly disappear, namely their associated currents go to 0.

In contrast to such fast converging cases, we have investigated slowly converging cases ‘a,’ ‘b,’ ‘c,’ ‘d,’ and ‘e’ in Fig. 2, as follows. For all these five cases, we verified that two groups of solutions are initially competing in the path search process. As shown in Fig. 3(b), for the case ‘a’ in Fig. 2, two groups of solutions (○ and ◇) coexist for a certain period, and it is conjectured that the appearance of two such competing groups results in the slow convergence to the shortest path since the second shortest path, once formed at an early stage of the path search process, requires certain time to be removed. Actually, it is verified that these two groups of solutions correspond to the shortest and the second shortest paths in the network[5]. Figures 4(a), (b), (c), (d), and (e) shows a summary of these network structures showing the shortest and the second shortest paths for the cases ‘a,’ ‘b,’ ‘c,’ ‘d,’ and ‘e,’ respectively, where l denotes the distance between nodes. Note these two competing paths have path lengths that are quite close to each other. From the perspective of application, the above observation suggests some advantage of the proposed method (and possibly of Tero’s model [1, 2] as well) over the conventional combinatorial algorithms; competing multiple paths are identified simultaneously at an early stage of the path search process.

Now, we numerically verify the validity of Assumption 1 in Section 2.2 in all simulations for shortest paths here. In Assumption 1, we have expected V_{ij} evolves slower than I_{ij}, and it is partly true as shown in the typical cases of Figs. 3(c), (d). As observed in Figs. 3(c), (d), this assumption is violated

[5] By using Dijkstra’s algorithm [10], we have verified that these paths are the shortest and the second shortest paths, respectively.

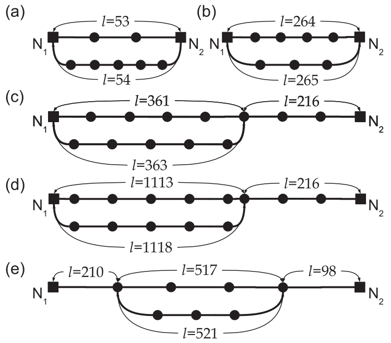

Fig. 4. Network structures for the cases (a), (b), (c), (d), and (e) with two competing shortest paths.

at a very early short span of the fast evolving process ($0 \leq t < 1$ s). However, we note this assumption is satisfied for all time after the above very early stage. This implies that the system always satisfies Assumption 1 if we regard the initial conditions of V_{ij} and I_{ij} as their values at $t = 1$ s for instance. And, this observation explains the situation how the derivation of Eqs. (8), (9) (and hence Eq. (10)) is validated, although the global convergence to the shortest path from any initial conditions is not clear from this argument.

3.3 Recovering process after random deletions of a link in the network

In Refs. [1, 2], Tero et al. showed that a new shortest path is quickly identified after random deletions of a link in the network. To verify that this ability is retained in our path search method, we have investigated the recovering process after random deletions of a link in the shortest path. Figures 5(a) and 6(a) respectively show two typical examples of the path search process for a random deletion of such a link at $t = 100$ s for instance, after the fast converging process shown in Fig. 3. In these examples, the shortest path ($\circ$) is already identified until $t = 100$ s, and this path disappears right after the deletion of a link on the shortest path. On the other hand, the new shortest path ($\diamond$; verified with Dijkstra's algorithm) emerges immediately after the deletion of a link in the shortest path[6]. Although here we only show the examples of $N = 2^5$ and $N = 2^6$ in Figs. 5 and 6 due to space limitation, the same pattern is observed in a total of all 60 instances for networks mentioned in Section 3.1. Thus, we expect that the proposed method can adapt quickly to a new shortest path, suggesting that the proposed method has certain robustness to temporal changes in the network, *i.e.*, resilient path finding ability, at least numerically.

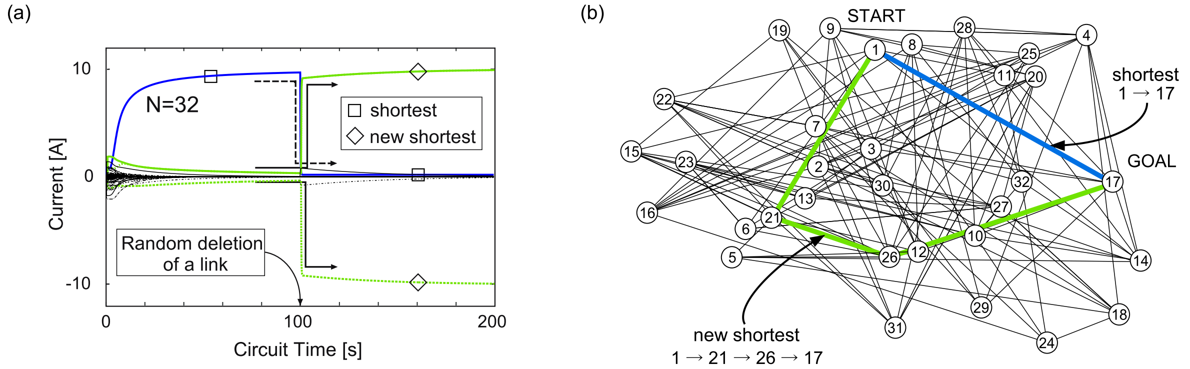

Fig. 5. A typical example with network size $N = 2^5 = 32$. START and GOAL represent the source node and the sink node, respectively. (a) Emergence of the new shortest path (*i.e.*, the second shortest path) after the deletion of a link at $t = 100$ s. [Note that all currents are 0 at $t = 0$, although it is difficult to recognize this from the graph due to their fast movement at the initial moment.] (b) The shortest path (blue) and the new shortest path (green) obtained in the network.

[6] In most cases where random deletions of a link do not destroy the second shortest link, this new shortest path is the second shortest path in the original network.

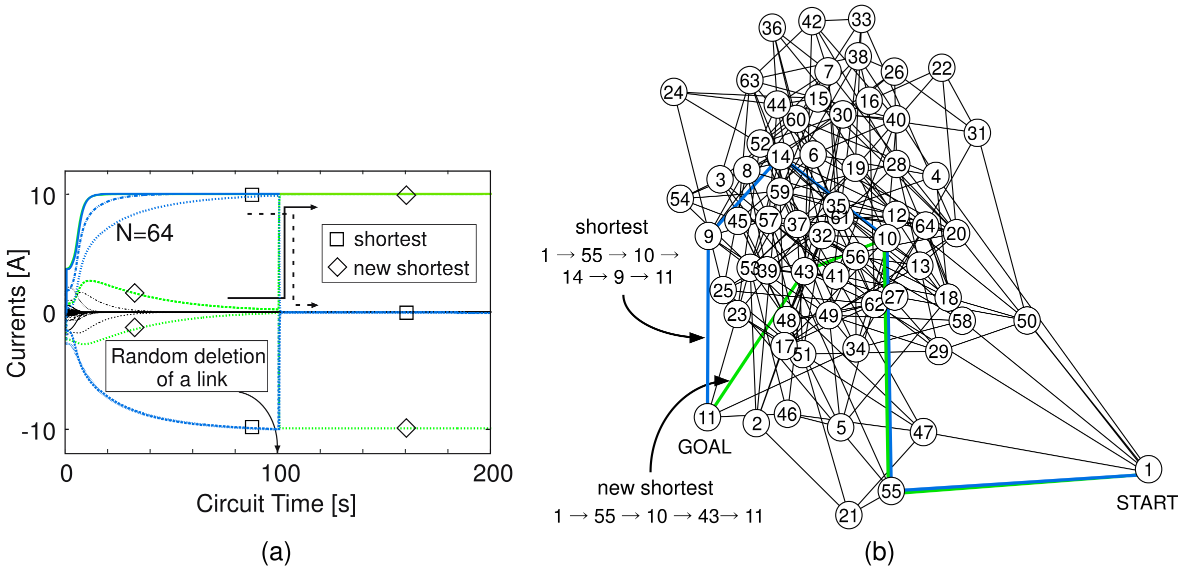

Fig. 6. A typical example with network size $N = 2^6 = 64$. (a) Emergence of the new shortest path after the deletion of a link at $t = 100$ s. [Note that all currents are 0 at $t = 0$, although it is difficult to recognize this from the graph due to their fast movement at the initial moment.] (b) The shortest path (blue) and the new shortest path (green) in the network.

4. Discussion and conclusions

We have proposed a dynamic circuit analogue for the shortest path search method [1, 2] which shows unique dynamical characteristics compared with the original Tero's model [1, 2] as well as the classical Dijkstra's algorithm [10]. One of the interesting characteristics is that competing multiple shortest paths (*i.e.*, the shortest path and the second shortest path) are simultaneously identified during the path search process. This task is known to be difficult to achieve with Dijkstra's algorithm or its modifications [11]. Although the circuit analogue here is somewhat heuristic and all the results are obtained numerically, the systematic simulations thus confirm practical utility of the proposed method, *i.e.*, reduction of huge number of variables (*i.e.*, number of links in the network) as well as a nontrivial nonlinear dynamics behind the system.

The elapsed time for the numerical shortest path search process can be further shorten by making the time step larger in the numerical integration scheme during the slow converging process in Fig. 3(a), for instance. Also, it is worth considering to replace t in Eq. (10) with t^2 by regarding this t here as a time-dependent parameter, after the fast evolving process shown in Fig. 3(a). Although this modification is non-rigorous, our preliminary results show that it successfully leads to the shortest path and reduces the simulation time, suggesting one possible practical acceleration method. Such accelerations of the proposed method is now ongoing and will be reported elsewhere.

In addition, a proof concerning why the shortest path is obtained with our method is now under consideration. For instance, in relation to rigorous results in [4–6] and their references within, it is worthwhile to consider if our system or Eqs. (1), (2), (3) and Eq. (10) has a Lyapunov function. On the other hand, it is also worth investigating if the slow dynamics of the proposed system with Eq. (10) is reduced to the original Tero's model with Eq. (5) after the fast evolving process, by using mathematical techniques such as the method of multiple scales.

Acknowledgments

We thank Drs. A. Tero (Kyushu Univ.), T. Saegusa, T. Nakagaki (Hokkaido Univ.), and R. Kobayashi (Hiroshima Univ.) for stimulating discussions. We also thank Drs. T. Saito (Hosei Univ.), K. Yamamura (Chuo Univ.), H. Asai (Shizuoka Univ.), K. Okumura (Hiroshima Inst. of Tech.), and S. Umetani (Osaka Univ.) for helpful comments.

References

[1] A. Tero, R. Kobayashi, and T. Nakagaki, “Physarum solver: A biologically inspired method of road-network navigation,” *Physica A*, vol. 363, pp. 115–119, 2006.

[2] A. Tero, R. Kobayashi, and T. Nakagaki, “A mathematical model for adaptive transport network in path finding by true slime mold,” *J. Theor. Biol.*, vol. 224, pp. 553–564, 2007.

[3] T. Nakagaki, H. Yamada, and Á. Tóth, “Intelligence: Maze-solving by an amoeboid organism,” *Nature*, vol. 407, p. 470, 2000.

[4] T. Miyaji and I. Ohnishi, “Physarum can solve the shortest path problem on Riemannian surface mathematically rigorously,” *Int. J. Pure and Applied Mathematics*, vol. 47, no. 3, pp. 353–369, 2008.

[5] V. Bonifaci, K. Mehlhorn, and G. Varma, “Physarum can compute shortest paths,” *J. Theor. Biol.*, vol. 309, pp. 121–133, 2012.

[6] L. Becchetti, V. Bonifaci, M. Dirnberger, A. Karrenbauer, and K. Mehlhorn, “Physarum can compute shortest paths: convergence proofs and complexity bounds,” *Automata, Languages, and Programming*, pp. 472–483, Springer, Berlin Heidelberg, 2013.

[7] L. Zhu, M. Aono, S.-J. Kim, and M. Hara, “Amoeba-based computing for traveling salesman problem: Long-term correlations between spatially separated individual cells of *Physarum polycephalum*,” *BioSystems*, vol. 112, no. 1, pp. 1–10, 2013.

[8] Y. Kondo and H.-A. Tanaka, “An electric circuit analogue of a mathematical model for adaptive transport network in true slime mold,” *Proc. NOLTA’07*, pp. 521–524, Vancouver, Canada, 16–19 September 2007.

[9] DIMACS, Ninth DIMACS Implementation Challenge – Shortest Paths, 2006. http://www.dis.uniroma1.it/~challenge9/.

[10] T. Ibaraki, *Algorithms and Data Structures in C*, Shokohdo, 1999 (in Japanese).

[11] S. Umetani, private communications, 2015.