

A Node-Weight Equalization Problem with Circuit-Based Computations

Yoichi Sakai*, Kiyoshi Nakayama[†] and Norihiko Shinomiya*

*Dept. of Information Systems Science, Soka University, Tokyo 192-8577, Japan

[†]Dept. of Computer Science, University of California, Irvine, CA 92697-3435, USA

Email: {shinomi, kiyoshi.nakayama}@ieee.org

Abstract—Circuit-based computation has contributed to designing efficient, highly-scalable, and reliable distributed control systems. As in load-balancing problems, uniform assignment of node weights is an important function in this domain. Intuitively, iterative computation for balancing workloads based on graph circuit structure seems to always converge to an optimal solution. However, no previous literature provides a proof that circuit-based computation for load balancing guarantees to average all the weights in a given mesh network topology. We therefore analyze the property of circuit-based computations for a node-weight equalization problem (NWE) whose objective is making the average deviation of graph weights converge on zero. This paper provides a graph theory-based verification of circuit-based computations for NWE, and a way to shorten the computation time required to solve the NWE.

I. INTRODUCTION

Since the birth of Kirchhoff's voltage and current laws, graph and circuit theory have played an important role in designing efficient and reliable circuit systems, even for complicated non-planar graphs. Beyond the area of circuit systems, which are also often called loops, tie-sets, cycles, or rings, the theory of circuits has contributed to designing distributed networks for simplified, resilient, and scalable operation [1]–[4]. More recently, an autonomous control technique for efficient management of a new smart grid based on circuit theory has also been under development [5]–[7]. Attention has been focused on mesh topologies because of the flexibility, efficiency, and resiliency they provide, and their familiarity through similarity with the information networks that form the Internet [8]. Further development and verification of circuit theory is therefore inevitably important for the design of sophisticated systems in many fields.

This paper deals with balancing network loads in a distributed system, abstracted in this paper as assigning weights to nodes. We examine properties of the balancing problem on the basis of circuit-based computations and perform simulation experiments for graph theory-based verification.

II. DEFINITION

For a given biconnected, undirected graph $G = (V, E)$ with vertices $V = \{v_1, v_2, \dots, v_n\}$ and edges $E = \{e_1, e_2, \dots, e_m\}$, let T and \bar{T} respectively be a spanning tree and a cotree of G , where $\bar{T} = E - T$. Focusing on a subgraph $G_T = (V, T)$ of G and an edge $l = (a, b) \in \bar{T}$, there exists only one elementary path P_T whose origin is b and terminal is a in G_T . Then, a

fundamental circuit that consists of the path P_T and the edge l is uniquely determined as $C(l) = (a, l = (a, b), P_T(b, a))$. It is known that $\mu = |\bar{T}|$ fundamental circuits exist in G , and they are called a *fundamental system of circuits*. A fundamental system of circuits $\mathbb{C}_B = \{C_1, C_2, \dots, C_\mu\}$ guarantees coverage of all vertices and edges in a biconnected graph G .

Let a load $W(v_i)$ be a real number assigned to each node v_i , which we refer to as a *weight* on the node. In this paper, balancing node loads is called the *Node-Weight Equalization Problem (NWE)*, the objective of which is making the average deviation of weights of a graph converge on zero. A set of nodes belonging to a fundamental circuit C_i is represented by $V(C_i)$. Let \bar{W}_i be the average value of weights on nodes that belong to C_i . Here, averaging all the weights in a circuit C_i from $W(v_i)$ to \bar{W}_i is defined as *circuit equalization* of $W(v_i)$. The issue to be analyzed in NWE is whether all the weights in G can converge on \bar{W} by repeating a procedure that picks a circuit C_i and averages its weights. As G is a biconnected, simple graph, each node belongs to more than one circuit. Therefore, any circuit has more than one adjacent circuit unless G itself is a circuit. Here, “adjacent” is defined as having more than one node shared with other circuits.

III. PROOF OF CONVERGENCE BY CIRCUIT EQUALIZATION

This section briefly introduces a proof of the convergence property in a biconnected graph. The following is the proposition to be proved:

Proposition: For a given biconnected, undirected graph $G = (V, E)$ with a weight $W(v)$ assigned to each node $v \in V$ and an arbitrary positive real number ϵ , iteration of circuit equalization makes the average deviation of G less than ϵ .

A sequence of circuits C_1, C_2, \dots, C_μ has the following characteristics:

- 1) For any r ($1 < r \leq \mu$),
 $V(C_1) \cup V(C_2) \cup \dots \cup V(C_{r-1}) - V(C_r) \neq \emptyset$,
 $(V(C_1) \cup V(C_2) \cup \dots \cup V(C_{r-1})) \cap V(C_r) \neq \emptyset$,
 $V(C_r) - V(C_1) \cup V(C_2) \cup \dots \cup V(C_{r-1}) \neq \emptyset$,
- 2) $V(C_1) \cup V(C_2) \cup \dots \cup V(C_\mu) = V$

This sequence of circuits exists in G from the characteristics of a biconnected graph. Circuit equalization is conducted according to the sequence order. Fig.1 shows the equalization process. In Fig.1, each C_r has one condition $\epsilon^{(r)}$, which is

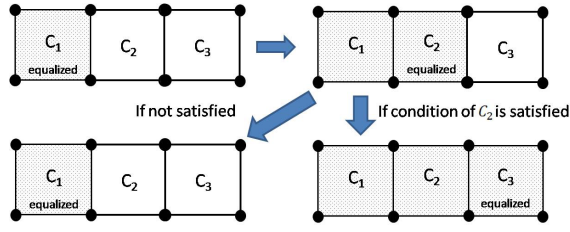


Fig. 1. The equalization process

Algorithm 1 The procedure of convergence on $\epsilon^{(\mu)}$

```

i:=1;
 $\epsilon' := \epsilon^{(\mu)} / \prod_{k=3}^{\mu} [\{(a_2-2)2^{k-1}+4\} / \{(a_2-2)2^{k-3}+2\}^2]$ 
while  $i \leq \mu$  do
  Converge  $C_i$ 
  if average deviation of  $\bigcup_{k=1}^i C_k < \epsilon'$  then
     $i := i + 1$ ;
     $\epsilon' := \epsilon' [\{(a_2-2)2^{i-3}+2\}^2 / \{(a_2-2)2^{i-1}+4\}]$ ;
  else
     $i := i - 1$ ;
     $\epsilon' := \epsilon' [\{(a_2-2)2^{i-1}+4\} / \{(a_2-2)2^{i-3}+2\}^2]$ ;
  end if
end while

```

decided by the following equation:

$$\epsilon^{(r)} = \epsilon^{(2)} \prod_{k=3}^r [\{(a_2-2)2^{k-3}+2\}^2 / \{(a_2-2)2^{(k-1)}+4\}] \quad (1)$$

Here, a_2 is the number of nodes of $C_1 \cup C_2$. $\epsilon^{(r)}$ in (1) is the worst value that exceeds the average deviation of C_1, C_2, \dots, C_r . Thus, the intended ϵ is $\epsilon^{(\mu)}$. When $\epsilon^{(r)}$ is satisfied, C_{r+1} is equalized. If $\epsilon^{(r)}$ is not satisfied, C_{r-1} is equalized as described in Algorithm 1. Such processes realize equalization of C_1, C_2, \dots, C_{μ} within the error $\epsilon^{(\mu)}$. From (1), if ϵ^2 is configured as

$$\epsilon^{(2)} = \epsilon^{\mu} \prod_{k=3}^{\mu} \frac{(a_2-2)2^{k-1}+4}{\{(a_2-2)2^{k-3}+2\}^2}, \quad (2)$$

then the average deviation of the graph can be less than $\epsilon^{(\mu)} (= \epsilon)$. From (2) above, the proof is complete. Details of this proof are discussed in [9].

IV. ANALYSIS ON SPEED OF CONVERGENCE

We now see that any biconnected graph can equalize the node weights. According to (2), $\epsilon^{(2)}$ depends on the number of circuits in the graph, μ , and the number of nodes of $C_1 \cup C_2$, a_2 . $\epsilon^{(r)}$ increases exponentially with a_2 and, especially, r . $\epsilon^{(2)}$ must therefore be quite small if the number of circuits μ becomes large.

We next present methods to speed up reduction of the average deviation of a graph, focusing on which circuits to equalize first.

A. Classification of Circuit

Any circuit C_i can be classified as one of the following four types:

Type A: C_i whose $W(v_i)$ is equal to or greater than \bar{W} , and has at least one v_i where $W(v_i) > \bar{W}$

Type B: C_i whose $W(v_i)$ is equal to or less than \bar{W} , and has at least one v_i where $W(v_i) < \bar{W}$

Type C: C_i whose $W(v_i)$ is equal to \bar{W}

Type D: C_i that has at least one v_i and v_j , where $W(v_i) > \bar{W}$ and $W(v_j) < \bar{W}$, respectively

Equalizing a circuit C_i of Type A, B, and C gives the following equation:

$$|\bar{W} - \bar{W}_i| = \frac{\sum_{v_i \in V(C_i)} |W(v_i) - \bar{W}|}{|V(C_i)|} \quad (3)$$

Equalizing a circuit C_i of Type D gives the following equation:

$$|\bar{W} - \bar{W}_i| < \frac{\sum_{v_i \in V(C_i)} |W(v_i) - \bar{W}|}{|V(C_i)|} \quad (4)$$

When a circuit of Type A, B, or C is equalized, the average deviation of the whole graph remains the same [3] [4], but when a circuit of Type D is equalized, the average deviation decreases. To reduce the average deviation of a whole graph, therefore, a circuit of Type D should be equalized.

B. Reduction of Average Deviation by Equalizing Circuit Type D

Let V_i^s be a set of nodes in C_i of Type D where for $v \in V(C_i)$, $W(v) < \bar{W}$. Let V_i^l be a set of nodes in C_i where for $v \in V(C_i)$, $W(v) > \bar{W}$. A is then defined as the total value of differences between $W(v)$ of V_i^s and \bar{W} , and B as the total value of differences between $W(v)$ of V_i^l and \bar{W} , as follows:

$$A = \sum_{v \in V_i^s} |W(v) - \bar{W}| \quad (5)$$

$$B = \sum_{v \in V_i^l} |W(v) - \bar{W}| \quad (6)$$

By these definitions of (5) and (6), the average deviation before equalizing the circuit of Type D is $(A+B)/|V(C_i)|$, and the average deviation of the circuit after equalization is $|A-B|/|V(C_i)|$. Therefore, the difference of the average deviations before and after equalization is as follows:

$$\frac{A+B}{|V(C_i)|} - \frac{|A-B|}{|V(C_i)|} = \begin{cases} \frac{2A}{|V(C_i)|} (A < B) \\ \frac{2B}{|V(C_i)|} (A > B) \end{cases} \quad (7)$$

Shortly, the average deviation within a circuit decreases by the lesser of $2A/|V(C_i)|$ or $2B/|V(C_i)|$ when equalizing C_i of Type D. Therefore, the average deviation of an entire graph by equalizing Type D circuits decreases as follows:

$$\frac{2 \times \min\{\sum_{v \in V_i^s} |W(v) - \bar{W}|, \sum_{v \in V_i^t} |W(v) - \bar{W}|\}}{|V|} \quad (8)$$

From Eq. (8), equalizing Type D circuits is effective because they have larger variance of weights. Although it has been discussed that equalizing Type D circuits decreases average deviation, it is not proven that the deviation approaches zero. However, no counterexample where average deviation approaches a nonzero value other has been found, so convergence to zero is assumed in an arbitrary biconnected graph.

C. Circuit Selection

Let $Q(C_i)$ be a variable of C_i determined by the *Circuit Evaluation Function* (CEF) $p(C)$. We consider the following CEFs:

- $p(C) = \text{Random}$
 $Q(C_i)$ is given by a random function $p(C)$.
- $p(C) = \text{Type D}$
 $Q(C_i)$ is a Boolean value where $Q(C_i) = 1$ if C_i is Type D; otherwise $Q(C_i) = 0$.
- $p(C) = \text{Type D \& AD}$
 $Q(C_i)$ is a combination of a Boolean value and the average deviation of C_i , where $Q(C_i) = 1$ if C_i is Type D and $Q(C_i) = 0$ otherwise.
- $p(C) = \text{Delta}$
 $Q(C_i)$ is δ , the difference between the maximum weight and the minimum weight in C_i .
- $p(C) = \text{AD}$
 $Q(C_i)$ is the average deviation of C_i .

In the case where $p(C) = \text{Type D}$, a circuit is selected randomly from circuits of Type D. Similarly, a circuit with maximum average deviation is selected from circuits of Type D in the case of $p(C) = \text{Type D \& AD}$. To reduce the number of equalizations, it should be considered which type of circuit best contributes to reduction of the average deviation of G . As mentioned in Section IV-B, the selected circuit should be of Type D and have the largest average deviation. Under these conditions we define the CEFs $p(C)$ above, which are likely to select such an effective circuit.

Fig. 2 shows a flow chart of the equalization procedure, which is as follows:

- 1) Select one C_i with maximum $Q(C_i)$
- 2) Equalize C_i
- 3) Update the $Q(C_i)$ of circuits adjacent to the equalized circuit
- 4) Check if the average deviation of G is less than ϵ

When $p(C) = \text{Random}$, there is no update procedure of $Q(C_i)$ in adjacent circuits because $Q(C_i)$ is not configured. The only difference in the procedures other than Random is how to compute $Q(C_i)$, such as using δ and AD.

V. SIMULATION AND EXPERIMENTS

This section verifies the nature of convergence in different circuit selection methods using a simulator written in Java. We developed this simulator on a computer with the Microsoft

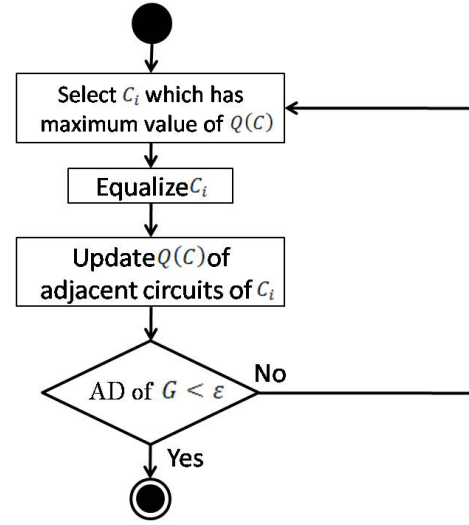


Fig. 2. Flow chart of overall processes

Windows 7 operating system and a Core 2 Duo 3.00 GHz CPU.

Simulation networks $G = (V, E)$ are randomly created with 100 to 1000 nodes, where the number of edges and circuits are approximately $2 \times |V|$ and $|V|$, respectively. The average value of degrees of nodes is similar to other simulated networks. The initial weight $W_0(v)$ of each node is randomly assigned between 0.0 and 100.0. In these experiments ϵ is 1.0, and the procedure iterates until the average deviation of G is less than ϵ . Twenty experiments are conducted for each configuration of $p(C)$, and the average number of equalizations N_{Total} and the average computational time CT are calculated.

A. Experimental Results and Analysis of N_{Total}

Fig.3 shows the average number of equalizations conducted until the average deviation of G becomes less than ϵ . As seen in Fig.3, there is a slight difference between Random and Type D, although as mentioned previously Type D contributes to reduction of the average deviation of G . The same tendency can be seen among Type D & AD, Delta, and AD. In those cases, it seems that selecting a circuit only from among Type D circuits does not reduce the total number of equalizations. Our analysis indicates that most circuits are of Type D in a graph where many circuits are connected to each other, where many Type D circuits in G always exist during the equalization process. This gives a large degree of freedom when selecting a Type D circuit, which becomes similar to selecting a circuit at random.

However, there is a large difference between groups of Random and Type D, and those of Type D&AD, Delta, and AD. The latter group is designed to select a circuit that has the biggest variance for reduction of the average deviation of G . It is therefore effective to select circuits whose with large weight variance.

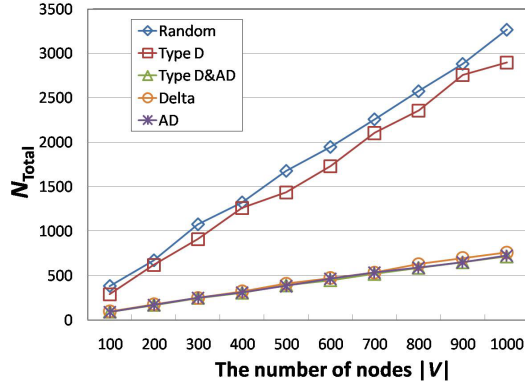


Fig. 3. Total equalizations until convergence of weights

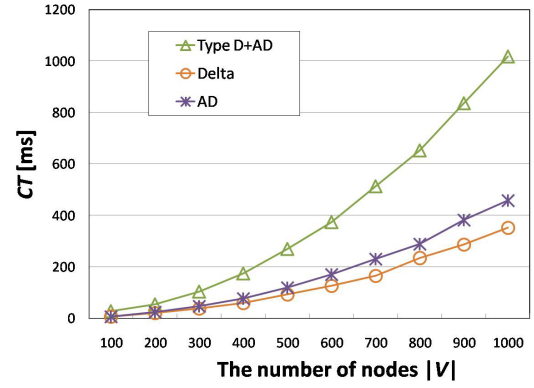


Fig. 4. Computation time CT

B. Experimental Results and Analysis on CT

Fig. 4 shows the computational time to convergence when selecting a circuit whose $p(C)$ is Type D & AD, Delta, and AD, as those CEFs require fewer equalizations for convergence. Computational time CT is given by the following equation:

$$CT = (T_{Slct} + T_{Eq} + T_{Upd}) \times N_{Total} \quad (9)$$

T_{Slct} : Computational time to find C with maximum $Q(C)$ in G

T_{Eq} : Computational time to equalize C

T_{Upd} : Computational time to update $Q(C)$ in adjacent circuits

Each element above corresponds to a step in Fig. 2. The computational complexity of T_{Slct} and T_{Eq} are the same for all CEFs. In addition, according to the result shown in Fig. 3, in this experiment N_{Total} does not vary with the CEF used. However, T_{Upd} heavily depends on the CEF, because the computation time to recalculate $Q(C_i)$ in adjacent circuits varies, depending on the $Q(C_i)$ used. Also, when the number of circuits in G increases the number of adjacent circuits of C_i also increases, making T_{Upd} large. Shortening T_{Upd} is required to keep CT low, as T_{Upd} occupies much of CT . The fastest convergence is realized when $p(C) = \text{Delta}$ (Fig.4), suggesting that calculation of δ of $Q(C_i)$ requires less time than other CEFs.

VI. CONCLUSION

We provided a theoretical proof and verification of a NWEP that has been practically demonstrated with distributed control based on circuit structure. As a result of the proof and verification, we showed that there is a way to equalize the weights with circuit-based computations in any biconnected graph. In addition, we analyzed the speeding up convergence by specific selection of equalized circuits through simulation and experiments. Results show that fast convergence is realized when circuits with the largest delta between the maximum and minimum weight are equalized first.

We are now developing a method of parallel selection of multiple circuits to facilitate the efficiency of circuit-based

computations. The proposed theory can be integrated with cloud-based optimization control for next-generation smart grids.

REFERENCES

- [1] G.Shen, and W.D.Grover, *Extending the p-Cycle Concept to Path Segment Protection for Span and Node Failure Recovery* IEEE Journal on Selected Areas in Communication, Vol. 21, No. 8, 2003.
- [2] Medard M, Barry RA, Finn SG, He W, Lumetta SS, *Generalized loop-back recovery in optical mesh networks*, IEEE/ACM Transactions on Networking 2002; 10(1):153-164.
- [3] N.Shinomiya *et al.*, "A theory of tie-set graph and its application to information network management," *International Journal of Circuit Theory and Applications*, John Wiley&Sons, Vol.29, No.4, pp.367-379(Jul.2001).
- [4] K. Nakayama, N. Shinomiya, H.Watanabe, *An autonomous distributed control method for link failure based on tie-set graph theory*, IEEE Transactions on Circuits and Systems-1, Regular Paper, 2012. DOI: 10.1109/TCSI.2012.2196109.
- [5] K. Nakayama, K. Benson, L. Bic, M. Dillencourt, *Complete Automation of Future Grid for Optimal Real-Time Distribution of Renewables*, To be appeared in IEEE International Conference on Smart Grid Communications (SmartGridComm), Oct, 2012.
- [6] K. Nakayama, N. Shinomiya, H.Watanabe, *An autonomous distributed control method based on tie-set graph theory in future grid*, International Journal of Circuit Theory and Applications, March, 2012. DOI: 10.1002/cta.1818.
- [7] EDISON, etc, *Irvine Smart Grid Demonstrate Project* <http://www.sustainability.uci.edu/Resources1/ISGDOverview.pdf>
- [8] S. Galli, A. Scaglione, Z. Wnag, *For the Grid and Through the Grid: The Role of Power Line Communications in the Smart Grid*, Proceedings of the IEEE, 2011, Vol.99, No.6, pp. 998 - 1027.
- [9] Y.Sakai, K.Nakayama, N.Shinomiya, "A property verification of node-weight equalization focusing on cycles of a graph," *proc.of Int'l — Conf.on ITC-CSCC*, P-T2-236 (Jul.2012).