INTERNATIONAL JOURNAL OF CIRCUIT THEORY AND APPLICATIONS
Int. J. Circ. Theor. Appl. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cta.1818

An autonomous distributed control method based on tie-set graph theory in future grid

Kiyoshi Nakayama[1,*,†], Norihiko Shinomiya[2,†] and Hitoshi Watanabe[2]

[1]*Dept. of Computer Science, University of California, Irvine, CA 92697-3435, USA*
[2]*Dept. of Information Systems Science, Soka University, Hachioji-shi, Tokyo 192-8577, Japan*

SUMMARY

In future power grids where electricity flows bidirectionally, the essential problem is to maximize the total efficiency of distributed energy resources. In complicated and large-scale systems such as modern power distribution networks, maximizing the efficiency of the entire system as a whole is extremely difficult. To solve the global optimization problem of such a complex network, this paper proposes an efficient distributed control method for future grid on the basis of tie-set graph theory, where a tie-set is a set of all the edges in a loop of a graph. On the basis of tie-set graph theory, global optimization of an entire network can be realized as a result of local optimization in μ-dimensional liner vector space, where μ is the nullity of the underlying graph of a power network. Although each tie-set has its limited local information, an entire network is gradually optimized in an orderly manner because of the theoretical basis of a tie-set graph. Simulation results of several thousand-node networks demonstrate balanced allocation of dispersed energy resources and thus effectiveness of the proposed method. Copyright © 2012 John Wiley & Sons, Ltd.

Received 1 September 2011; Revised 26 February 2012; Accepted 31 March 2012

KEY WORDS: smart gird; distributed control; graph theory; renewable energy; power distribution; loop-based management; tie-set graph

1. INTRODUCTION

Information and network technologies are now being used throughout power grids. This represents the most profound electrical power revolution in a century. Power grid has been composed of central generating stations and electromechanical power delivery systems operated from control centers. But now, the system is transforming itself into a smart grid that integrates a multitude of distributed energy resources, uses solid state electronics to manage and deliver power, and employs automated control systems [1–3]. Under the background, energy networking projects have been conducted around the world as seen through the works [4,5], where local grid with intelligent power switches is introduced.

The future power grid will invariably feature rapid integration of alternative forms for energy generation; thus, electricity flows bidirectionally [6]. Harvesting energy from the environment is feasible in many applications to ameliorate the energy limitations in sensor networks. In the paper [7], an adaptive duty cycling algorithm that allows energy harvesting sensor nodes is proposed to autonomously adjust their duty cycle according to the energy availability in the environment. The renewable energy resources applications will offset the dependence of fossil fuels, provide green power options for atmospheric emissions curtailment, and put in policy the provision of peak load shaving.

*Correspondence to: Kiyoshi Nakayama, Department of Computer Science, University of California, Irvine, CA 92697–3435, USA.
†E-mail: kiyoshi.nakayama@ieee.org; shinomi@soka.ac.jp

This will require new optimization for energy resources that are inherently distributed with interconnection standards and operational constraints. Targeting such a future grid, controllers to handle a bidirectional power flow have been proposed [8], taking advantages of power converters able to sustain a bidirectional power flow [9]. In future power networks where electricity flows bidirectionally, one of the most essential problems is to maximize the efficiency of distributed renewable energy resources. To maximize the efficiency of distributed energy resources, any futile power generation derived from fossil fuels such as diesel engine oil should be reduced by effectively taking advantage of natural energies. Effective use of renewable energies can be realized by distributing electricity in a balanced manner so that all the storage systems store the same amount of power. This issue leads to a balancing problem of electric energies among dispersed power storage systems. If dispersed renewable energy resources are homogeneously allocated, there will be no need to generate excessive and futile electricity; thereby, energies can be conserved, and thus, emissions of CO_2 are also reduced [10]. In addition, the power-generating cost can be cut down.

However, the amount of electricity generated by natural energies fluctuates depending on weather conditions. Therefore, stable and reliable electricity distribution becomes difficult especially in a large-scale network. Integration of renewable resources including those natural energies from consumer premises should advance global energy sustainability [11]. For this reason, development of an autonomous distributed architecture that can realize balanced allocation of dispersed renewable energy resources is inevitable to conduct high-reliable operation of future power distribution networks.

Advent of smart meters makes it possible to visualize the deficiency and excess of electric power distributed among houses and factories, among others, and thus, autonomous distributed control becomes feasible by using such advanced meters. The main functions of a smart meter are as follows [12]:

1. Presence Function: The amount of electricity generated by solar panels, aerogenerators, among others, and consumed by users is monitored in real time. The digitalized value can be provided with electric operators or outside applications by wireless or cable communications.
2. Load Shedding Function: A smart meter is connected to electric household appliances and electrical equipment such as lights, air conditioners, TVs, and kitchen appliances through wired or wireless communication. ON/OFF equipment operations and temperature modulation of air conditioning systems can be controlled externally under the direction of an operating point such as an electric power company to adjust electric loads on a power grid.

Data and information obtained by smart meters are centralized on the Internet; thereby, smart metering is integrated into cloud computing [13]. Real-time monitoring by smart meters enables distributed systems to effectively control power grid. When there is insufficient power station capacity to supply the demand (load) from all the customers, the electricity system could become unstable, possibly resulting in a national blackout. To avoid this, load shedding has been conducted, and there are many methods for autonomous power balancing as seen in [14,15]. Our proposed method for power balancing focuses on balanced allocation of distributed energy resources from a perspective of graph theory without load shedding or load leveling.

Because the birth of the graph and circuit theory traced back to Kirchhoff's current and voltage laws (KCL and KVL) in 1845, its fundamental concept has been to study not the characteristics of individual elements but the nature of the overall structure of an entire system that leads us to the graph theory. In spite of abrupt change of technology, the graph and circuit theory has played an important role as an essential philosophy of the topological aspects in general systems. It has been applied to various fields of circuits and systems in formulating problems, solving the problems, and characterizing properties of the problems. In complicated and large-scale systems such as modern power distribution networks, maximizing the efficiency of the entire system as a whole is extremely difficult. To solve the global optimization problem of such a complex network in terms of overall systems control, this paper proposes an efficient distributed control method for future grid focusing on tie-set graph theory [16], where a tie-set is a set of all the edges in a loop of a graph. On the basis of tie-set graph theory, global optimization of an entire network can be realized as a result of local optimization in μ-dimensional liner vector space, where μ is the nullity of the underlying graph of a power network (Figure 1).

DOI: 10.1002/cta

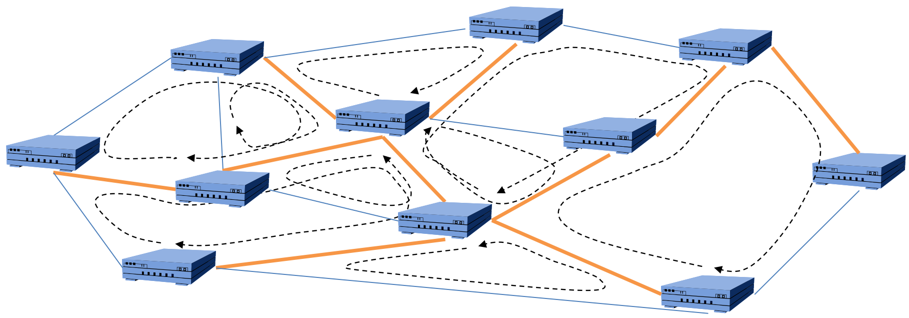


Figure 1. Autonomous control in μ independent fundamental tie-sets.

In distributed control, partition units of a network become important. In information networks, there are many techniques that focus on loops, which are often referred to as ring, cycle, or circuit structure, as partition units [17–19]. In addition, there are a variety of methods on covering a network with cycles [20–22]. Utilization of loop-based management for local units of a power grid is also gathering attention in electricity control. For example, the work proposes the reliable overlay topology design for the smart grid network focusing on ring topology [23]. A management method that realizes reducing voltage rise and power flow control focusing on loops in a smart grid has been proposed [24]. If power networks are also managed in local units defined by tie-sets, more flexible control for distributed energy resources is considered to be possible.

Many authors study on tie-sets and its applications from a variety of perspectives. In a previous work, the principal partition theorem focusing on loops of an information network is proposed for distributed network management [16,25]. The study introduces the tie-set graph theory that provides theoretical nature of tie-sets that underlie an information network. The paper [26] presents an efficient method of enumerating all minimal tie-sets connecting selected nodes in a mesh topological network. Among studies on tie-sets, it is also suggested by the works [27,28] that local optimization in a μ-dimensional liner vector space whose basis is a system of μ linear independent tie-sets leads to global optimization. A tie-set-based approach to network fault tolerance is also proved to be effective than traditional tree-based centralized approaches [29–31].

On the basis of that recent development of technologies in smart grid, the primary motivations of this study are listed as follows:

- Taking full advantage of distributed renewable energy resources even in a complex power network;
- Realizing reliable and sustainable distributed control for the future gird where generations of renewable energy resources are quite unstable depending on whether conditions;
- Cutting the cost of futile power generation by allocating renewable energies in a balanced manner;
- Reducing the generation of CO_2 by cutting power generation from fossil fuels;
- Accomplishing real-time demand response by autonomous control in μ-dimensional independent cycles;
- Minimizing the transmission and conversion loss when electricity is distributed;
- Making power distribution networks resilient on the basis of ring management from various failures or disasters.

All the issues are quite important for future smart grid management. In addition, this research also poses significant issues to be carried out in the future when incorporating practical issues into the actual power grid. Future development of this research leads to the innovation of business models in terms of not only the power grid companies but also various kinds of businesses including IT and network system fields.

In this paper, the amount of power, which is the algebraic sum of power production, consumption, inflow, and outflow monitored by smart meters of each node, is defined as a node value (NV). Power flow on a link between two nodes is defined as an edge flow. To realize efficient power management, the balanced allocation problem of distributed energy resources can be formulated as an NV

DOI: 10.1002/cta

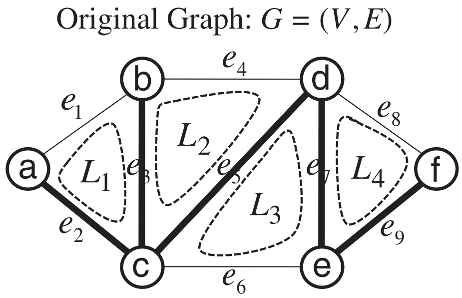


Figure 2. An example of a fundamental system of tie-sets.

homogenization problem that minimizes the variance of all the NVs. Our proposed distributed algorithm solved the NV homogenization problem with iterative computations in independent tie-sets. Simulation results also demonstrated the global optimization of several thousand-node networks.

2. TIE-SET GRAPH THEORY

2.1. Fundamental system of circuits and tie-sets

For a given biconnected and undirected graph $G=(V,E)$ with a set of vertices $V=\{v_1, v_2, \ldots, v_n\}$ and a set of edges $E=\{e_1, e_2, \ldots, e_m\}$, let $L_i = \{e^i_1, e^i_2, \ldots, e^i_k\}$ be a set of all the edges that constitutes a loop in G. The set of edges L_i is called a 'tie-set' [32]. Let T and $\bar{T}$ be a tree and a cotree of G, respectively, where $\bar{T} = E - T$. $\rho = \rho(G) = |T|$ and $\mu = \mu(G) = |\bar{T}|$ are called the *rank* and the *nullity*, respectively. A tree T on a graph $G=(V,E)$ is an ultranet set of edges that does not include any tie-set. In other words, for $l \in \bar{T}$, $T \cup \{l\}$ includes one tie-set. 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, an elementary circuit that consists of the path P_T and the edge l is uniquely determined as follows:

$$\begin{aligned} L(l) &= (a, l = (a,b), P_T(b,a)) \\ &= (a, l, v_0 = b, t_1, v_1, \cdots, t_h, v_h = a) \end{aligned} \tag{1}$$

In this way, a circuit determined by an edge $l = (a,b) \in \bar{T}$ and a path $P_T(a,b)$ in G_T is denoted as a 'fundamental circuit'. A simple circuit[1] can be expressed by a set of edges, so called a tie-set. The tie-set corresponding to a fundamental circuit regarding T is denoted as a 'fundamental tie-set' regarding T. It is known that μ fundamental circuits and tie-sets exist in G, and they are called a 'fundamental system of circuits' and a 'fundamental system of tie-sets', respectively. A fundamental system of tie-sets $\mathbf{L}_B=\{L_1, L_2, \ldots L_\mu\}$ covers all the vertices and edges in G as shown in Figures 1 and 2. Let V_i be a set of all the vertices contained in a fundamental circuit $L(l_i)$, where $V_1 \cup \cdots \cup V_i \cup \cdots \cup V_\mu = V, (\mu = |\bar{T}|)$ for a given graph $G=(V,E)$.

2.2. Tie-set graph

A graph $\underline{G} = (\underline{V}, \underline{E})$ is defined as a tie-set graph, where a set of vertices $\underline{V}$ corresponds to a fundamental system of tie-sets $\{L_1, L_2, \ldots, L_\mu\}$, and a set of edges $\underline{E}$ corresponds to a set of edges $\{\underline{e}(L_i, L_j)\}, (i \neq j)$, which represent the connections among tie-sets. There are two ways to determine $\underline{e}(L_i, L_j)$ by focusing on a relation between two fundamental tie-sets $\Re(L_i, Lj)$.

[1]If a path is a simple path with no repeated vertices or edges other than the starting and ending vertices, it is called a simple circuit, cycle, circle, or polygon.

DOI: 10.1002/cta

2.2.1. E-tie-set graph $\underline{G}_e = (\underline{V}, \underline{E}_e)$. If $\Re(L_i, L_j)$ is determined by the set of common edges of L_i and L_j, $\underline{G}_e = (\underline{V}, \underline{E}_e)$ is denoted as e-tie-set graph [25] as shown in Figure 3(a).

2.2.2. V-tie-set graph $\underline{G}_v = (\underline{V}, \underline{E}_v)$. If $\Re(L_i, L_j)$ is determined by the set of common vertices of L_i and L_j, $\underline{G}_v = (\underline{V}, \underline{E}_v)$ is denoted as v-tie-set graph [25] as shown in Figure 3(b).

Each fundamental tie-set of a given graph G is uniquely mapped to the specific tie-set graph $\underline{G}_e$ and $\underline{G}_v$.

2.3. *Relationship between a tie-set graph and a power grid*

The result of the recent Pacific Crest Mosaic Smart Grid Survey [33] shows that when asked about the relative importance of communication technologies, respondents selected mesh networks as the technology they will most likely use in their service territory (45%) as shown in Figure 4. Point-to-point radio frequency and WiMax were listed as most important by 25% and 10% of respondents, respectively. Tie-set graph theory is discussed on a biconnected graph and applicable to a mesh topological network. Therefore, the distributed control method proposed in this paper has potential to be applied to future smart grid networks. Application of the tie-set concept to distributed control has two major advantages as follows:

1. Simplicity in Synchronization: Now that electricity is able to flow bidirectionally on a power grid, there is a need to communicate the information of demands and supplies in a certain unit. On the basis of loop structures, message passing can be realized within a loop. Therefore, by sending a message around on a circle periodically, each node can recognize the information of other nodes in a loop defined by the tie-set. That leads to the simplicity and consistency in terms of synchronization of state information among nodes in a tie-set.
2. Reduction of Electricity Loss: If the distance between a point of demand and supply becomes larger, the power loss increases. Hence, the distance of power transport should be shortened to minimize the loss of electricity. If electricity is supplied within a loop, the distance of power

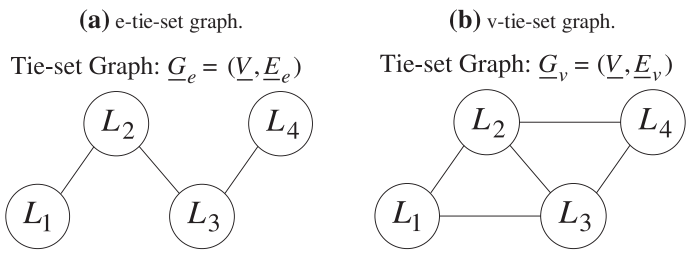


Figure 3. Tie-set graph of Figure 2. (a) e-tie-set graph and (b) v-tie-set graph.

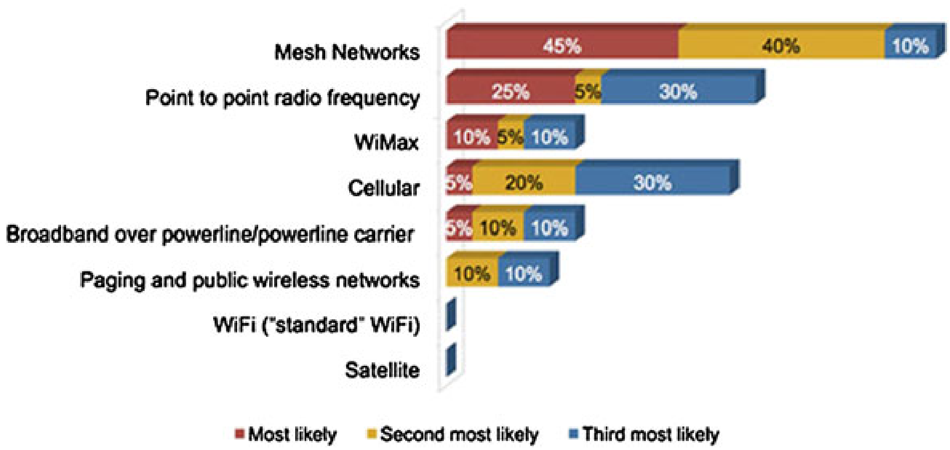


Figure 4. The result of surveys by Pacific Crest Mosaic [33].

DOI: 10.1002/cta

transmission is greatly shortened in comparison with the case that electricity once converses on a substation and is distributed to points of demand [34].

The relation among tie-sets is substituted for $\underset{\sim}{E}$ of a tie-set graph $\underset{\sim}{G} = (\underset{\sim}{V}, \underset{\sim}{E})$. To avoid discrepancy of information among nodes, $\underset{\sim}{G}_V$ is more appropriate than $\underset{\sim}{G}_e$. If procedures are being conducted in a certain tie-set, adjacent tie-sets should not conduct any procedures because discrepancy among tie-sets occurs. After a tie-set L_i finishes its procedures, the processor should notify the updated information to each adjacent tie-set L_j defined by a relation of $\underset{\sim}{e}_V(L_i, L_j)$. If a problem that cannot be solved in one tie-set is emerged, the tie-set contacts adjacent tie-sets connected via $\underset{\sim}{e}_V(L_i, L_j) \in \underset{\sim}{E}_V$ to seek for aid to solve the problem. For instance, the shortage of electricity in certain tie-set can be resolved by obtaining power from adjacent tie-sets.

3. DISTRIBUTED ALGORITHM BASED ON TIE-SET

3.1. State information of a node

Each node v_i mainly has three types of information as state information as follows:

1. *Incident Links*: Information of links connected to v_i.
2. *Adjacent Nodes*: Information of nodes that are connected through incident links of v_i.
3. *Tie-set Information*: Information of fundamental tie-sets to which v_i belongs. When a fundamental tie-set L_i contains e_i that includes v_i in its two vertices, it is defined that v_i belongs to L_i and has information of L_i.

Tie-set Information of L_i is as follows:

- *EdgeTable*: Information of a set of all the edges (links) that constitutes a tie-set L_i.
- *VertexTable*: Information of a set of all the vertices (nodes) included in a circuit formed by a tie-set L_i.

A string of the vertices must be sorted to satisfy the order as follows: $\left(v_0^i, e\left(v_0^i, v_1^i\right), v_1^i, \ldots, v_h^i, e\left(v_h^i, v_0^i\right), v_0^i\right)$, where $V_i = \left\{v_0^i, v_1^i, \ldots, v_h^i\right\}$ represents a set of all the vertices of L_i.

- *State information of a tie-set* L_i: Local information of nodes and links in a tie-set L_i.

Here is an example of state information of a node c in Figure 2. The node c has information of $\{e_2, e_3, e_5, e_6\}$ as incident links, $\{a, b, d, e\}$ as adjacent nodes, and tie-set information of $\{L_1, L_2, L_3\}$.

3.2. Distribute algorithm for configuring tie-set information

As described in Section 3.1, each node has information of fundamental tie-sets to which the node belongs to solve any problems within some loops. To obtain Tie-set Information, each node executes a distributed algorithm to recognize fundamental tie-sets. By automatically configuring state information of tie-sets, a burden for initialization of network managers is greatly lessened especially in large-scale networks.

In the underlying graph of a power network, let a 'tree link' be a link of a tree T and a 'cotree link' be a link of a cotree $\overline{T}$ in $G=(V, E)$, respectively. In this paper, tree links are expressed by thick lines, whereas cotree links are expressed by thin lines as shown in Figure 2. For example, tree links and cotree links are $\{e_2, e_3, e_5, e_7, e_9\}$ and $\{e_1, e_4, e_6, e_8\}$ in Figure 2, respectively.

A tree T is easily constructed by executing the spanning tree algorithm, which is one of the basic distributed algorithms. A *Find Tie-set* message, which is used to configure tie-set information of a node, includes information as follows:

- *EdgeTable*: A set of links through which a *Find Tie-set* message passed.
- *NodeTable*: A set of nodes through which a *Find Tie-set* message passed.

If *Find Tie-set* messages are processed according to the rules, each node can recognize the information of fundamental tie-sets.

DOI: 10.1002/cta

Initialization: First, each node v_o creates *Find Tie-set* messages and then sends those messages to all the adjacent nodes of v_o. When sending a *Find Tie-set* message to an adjacent node v_a, v_o adds node information of v_o to *NodeTable* and adds information of a link connected to both v_o and v_a to *EdgeTable*. Let v_r be a node that receives a *Find Tie-set* message. After receiving a *Find Tie-set* message, v_r executes a different procedure by the following cases.

Case 1: $v_r \neq v_o$
In this case, if *EdgeTable* of the *Find Tie-set* message includes more than one cotree link, v_r discards the message. If *EdgeTable* contains no or one cotree link, v_r copies the *Find Tie-set* message and sends the copied message to adjacent nodes that are not included in *NodeTable*. In case that the adjacent node is v_o, v_r sends the copied message to v_o even if $v_o \in$ *NodeTable*. When sending a copied message to an adjacent node v_a, v_r adds node information of v_r to *NodeTable* and adds information of a link connected to both v_r and v_a to *EdgeTable*.

Case 2: $v_r = v_o$
In this case, the *Find Tie-set* message has passed through a certain loop in a network. If *EdgeTable* coincides with a fundamental tie-set, the information of *EdgeTable* and *NodeTable* included in the *Find Tie-set* message is stored in v_o.

In this algorithm, the number of messages casted on a network, so called Communication Complexity, can be analyzed by focusing on a format of the *Find Tie-set* message. The communication complexity is determined to be $O(n^4)$ from a perspective of distributed algorithm, where $n = |V|$. However, the number of physical ports of a node is actually limited to certain constant number. In this case, the Communication Complexity is determined to be $O(n^3)$. As for execution time, a message passes through on tree links and one cotree link. Therefore, time complexity is $O(D)$ where D is defined as a diameter of a graph G.

3.3. *Leader election problem in each tie-set*

In each tie-set, it is imperative to select a leader node to solve a dilemma that occurs when using an overlapping resource among other tie-sets. For example, a local control unit defined by a tie-set shares control of overlapping resources with other units. If two control units come up with two different operating points, the leaders of the two units have to negotiate with each other to gain priority of control.

There are some criteria to choose a leader node in a tie-set. Given the state information defined previously, major criteria are as follows:

1. Magnitude relation of node ID (physical addresses).
2. The number of adjacent nodes (incident links).
3. The number of tie-sets to which a node belongs.

The use of criterion 1 enables each tie-set to decide its leader node uniquely, whereas criteria 2 and 3 do not. When focusing on criteria 2 or 3 to determine a leader node, there is a need to combine with criterion 1. In this study, a node with the largest number of tie-sets as well as with the smallest node ID is selected as a leader node. If a leader node is involved in many tie-sets, communication load can be reduced. As each node has Tie-set Information, it is possible for a node to send a message around on a tie-set and recognize state information of other nodes. A leader node can simply be determined by sending a message around on a tie-set and comparing the above value individually. To decide a leader node in each fundamental tie-set L_i, each node sends a *Decide Leader* message to a tie-set. A *Decide Leader* message contains the information as follows:

- id_m: The smallest (or largest) node ID among a set of nodes through which a *Decide Leader* message has passed.
- $value_m$: The optimal value of criteria among a set of nodes through which a *Decide Leader* message has passed. The value of criteria is, for instance, the value of aforementioned criteria 2 or 3.
- *AddressTable*: Addresses of nodes that belong to L_i. A string of addresses must be sorted to satisfy the order as follows: $L_i = \left(v_0^i, e\left(v_0^i, v_1^i\right), v_1^i, \ldots, v_h^i, e\left(v_h^i, v_0^i\right), v_0^i\right)$ where $V_i = \left\{v_1^i, \ldots, v_h^i\right\}$ represents a set of vertices of L_i.

DOI: 10.1002/cta

Initialization: For each fundamental tie-set L_i of Tie-set Information, each node v_o creates a *Decide Leader* message and then sends the message to an adjacent node v_a that belongs to L_i. When sending a *Decide Leader* message to an adjacent node v_a, v_o sets its node ID to id_m and assigns its criteria information $value_{v_o}$ to $value_m$. Let v_r be a node that receives a *Decide Leader* message. After receiving a *Decide Leader* message, v_r executes a different procedure by the following cases:

Case 1: $v_r \neq v_o$
In this case, v_r compares its criteria information $value_{v_r}$ with $value_m$ of the received *Decide Leader* message. If $value_{v_r}$ of v_r is better than $value_m$ of the received message, v_r substitutes $value_{v_r}$ for $value_m$. v_r also substitutes its node ID id_{v_r} for id_m. If $value_{v_r} = value_m$, v_r compares its node ID id_{v_r} with id_m. Given that smaller node ID is better, if $id_{v_r} < id_m$, v_r substitutes $value_{v_r}$ for $value_m$ and substitutes its node ID id_{v_r} for id_m. Otherwise, v_r does not change information of the received *Decide Leader* message. Then, v_r sends the message to adjacent node v_a.

Case 2: $v_r = v_o$
In this case, the *Decide Leader* message has passed through on a tie-set L_i. Then, v_o recognizes a node that holds node ID id_m as a leader of L_i. In this algorithm, communication complexity is $O(\mu \times |V|)$. As for execution time, a message passes through on a tie-set simultaneously. Therefore, time complexity is $O(D)$ where D is defined as a diameter of G.

3.4. Communications among tie-sets

A leader node of a tie-set determines the optimal control for distributed problems and solves them. A leader also decides how to use distributed resources allocated to each node in a tie-set domain. When solving distributed problems, communications among tie-sets are frequently required. To realize communications among tie-sets, each leader of a tie-set must recognize connection information with adjacent tie-sets. In addition to state information described in Section 3.1, a leader node v_l^i of a tie-set L_i has information as follows:

- Adjacent Tie-sets $\mathbf{L}_a = \left\{L_1^a, L_2^a, \ldots\right\}$

On the basis of the concept of $\Re\left(L_i, L_j\right)$ stated in the tie-set graph section, an adjacent tie-set L_j of L_i is determined according to the relation of connection $\underline{e}_V(L_i, L_j) \in \underline{E}_V$ of $\underline{G}_V$.

Let v_l^i and v_l^j be a leader node of L_i and L_j, respectively. As defined in 1, V_j stands for a set of all the nodes included in L_j. When the leader node v_l^i of L_i communicates with the leader node v_l^j of an adjacent tie-set L_j, v_l^i executes a different procedure according to the following cases.

Case 1: $v_l^i = v_l^j$
In this case, there is no need for v_l^i to send any message because v_l^i itself is the leader of the adjacent tie-set L_j. Instead, $v_l^i\left(= v_l^j\right)$ decides a proper procedure considering state information of both L_i and L_j.

Case 2: $v_l^i \in V_j, \left(v_l^i \neq v_l^j\right)$
In this case, v_l^i has state information of L_j. Thereby, v_l^i only sends a *Tie-set Communication* message to v_l^j using the topology information of L_j.

Case 3: $v_l^i \in V_j$
In this case, v_l^i first detects a node v_h that satisfies a condition $v_h \in V_j$ and then sends a *Tie-set Communication* message to v_h. Next, v_h sends the *Tie-set Communication* message to v_l^j by using topology information of L_j.

A routing table for communication among tie-sets should be computed according to the aforementioned rules before conducting certain procedure so that each leader node can quickly communicate with other leaders by using an appropriate path to them.

DOI: 10.1002/cta

4. BALANCED ALLOCATION OF DISTRIBUTED ENERGY RESOURCES

Unlike traditional power grid systems, future grids integrate intermittent renewable energy resources such as solar and wind power. However, energy generation may be unstable because of weather conditions. To maximize efficiency of distributed energy resources as well as realize a stable operation of interactive power flows, balanced allocation of those resources will be required. To discuss the balanced allocation problem with graph theory, we define a NV homogenization problem that minimizes a variance of all the NVs and propose distributed optimization algorithms on the basis of local unit defined by tie-set graph. We do not currently consider characteristics of power systems such as transmission loss, conversion loss to model the balanced allocation problem of dispersed energies on the basis of graph theory in this paper. In addition, data of smart meters (NVs) are derived from a snapshot of a fluctuating power network at a given time.

4.1. Definition of node value homogenization problem

Here, the amount of power (algebraic sum of power production, consumption, inflow, and outflow monitored by smart meters) that each vertex v possesses is defined as NV $C(v)$. For any two vertices v, u $(v \neq u)$, let $f(v, u)$ be a power flow over an edge $e(v, u)$ from vertex v to u and defined as an edge flow, where if $f(v, u)$ flows along the direction of an edge $e(v, u)$, then $f(v, u) > 0$; otherwise, $f(v, u) < 0$. To simplify a problem, both node value $C(v)$ and edge flow $f(v, u)$ are defined as scalar. Let $f_{P(s,t)}$ be a path flow from vertex s to t. Then, a path flow $f_{P(s,t)}$ is defined as follows:

$$f_{P(s,t)} = \{f(s, v_0), f(v_0, v_1), \ldots, f(v_h, t)\} \tag{2}$$

If the value of $f_{P(s,t)}$ is decided, each edge flow of $f_{P(s,t)}$ is decided as follows:

$$f_{P(s,t)} = F' \Longrightarrow \begin{cases} f'(s, v_0) = f(s, v_0) + F' \\ f'(v_0, v_1) = f(v_0, v_1) + F' \\ \vdots \\ f'(v_h, t) = f(v_h, t) + F' \end{cases} \tag{3}$$

If the value of an edge flow $f(v, u)$ over $e(v, u)$ is decided, then the NV of v and u is updated according to the following rule:

$$f(v, u) = F \Longrightarrow \begin{cases} C'(v) = C(v) - F \\ C'(u) = C(u) + F \end{cases} \tag{4}$$

Therefore, if an edge flow $f(v, u)$ is decided, NV $C(v)$ and $C(u)$ are automatically changed according to Equation 4. Under the condition, an NV homogenization problem is defined as follows.

Node Value Homogenization Problem:
For each $e(v, u) \in E$, decide the edge flow $f(v, u)$ that satisfies the following objective function.

$$\text{Minimize} \max_{v \in V} \{C(v)\} \tag{5}$$

In other words, an NV homogenization problem is equivalent to the problem that minimizes the variance of all the NVs by adjusting each edge flow $f(v, u)$. The variance is defined as follows:

$$\sigma^2 = \frac{\sum_{v \in V} (C(v) - \bar{C})^2}{|V|}, \left(\bar{C} = \frac{\sum_{v \in V} C(v)}{|V|}\right) \tag{6}$$

To solve the min–max problem 5, the variance should converge on 0 as follows:

$$\sigma^2 \to 0 \tag{7}$$

A total value of all the nodes does not change because an edge flow is scalar. Thereby, this min–max problem is equivalent to max–min problem and all the NVs that converge on the average value $\bar{C}$.

4.2. Distributed control method for node value homogenization problem

This section suggests a distributed control method for the NV homogenization problem defined previously. To solve the problem, we propose the control model as shown in Figure 5. The model follows the situation where a controller centralizes the information and data of smart meters in a power network. On the basis of the information provided by smart meters, a controller issues an instruction to a tie-set. Here, NVs are the data obtained by smart meters at a given time (snapshot) and do not dynamically change. Completely distributed control without any controller will be discussed in future works.

Figure 6 is the flow chart of overall processes. A controller is a management system of an entire power network. Tie-set processor of L_i is a node that decides any procedure and solves a problem on behalf of a tie-set L_i. Tie-set processor can be a processor assigned to each tie-set or a leader node of a tie-set. Here, a leader node plays a role as a tie-set processor. Node v is a node that sends an edge flow to node u.

First, the controller selects one tie-set L_i from a fundamental system of tie-sets $\mathbf{L}_B = \{L_1, L_2, \ldots, L_\mu\}$, considering tie-set evaluation function $p(L_i)$. The tie-set evaluation function $p(L_i)$ is subject to characteristics of a network, the objective function, and so on. Let V_i be the set of all the nodes included in a fundamental cycle $L(l_i)$ whose fundamental tie-set is L_i. In this research, we define three types of tie-set evaluation functions $p(L_i)$ as follows:

- $p(L_i) = \text{Random}$

  If the value of $p(L_i)$ of a tie-set is randomly assigned, the controller picks up one tie-set at random.

- $p(L_i) = C_{\max} - C_{\min}, \left(C_{\max} = \max\limits_{v \in V_i} \{C(v)\}, C_{\min} = \min\limits_{v \in V_i} \{C(v)\} \right)$

  This evaluation function is the delta between the maximum NV and the minimum NV in a tie-set.

- $p(L_i) = \sigma^2$

  This evaluation function is the variance of NVs in a tie-set. The variance σ^2 is defined in Equation 6.

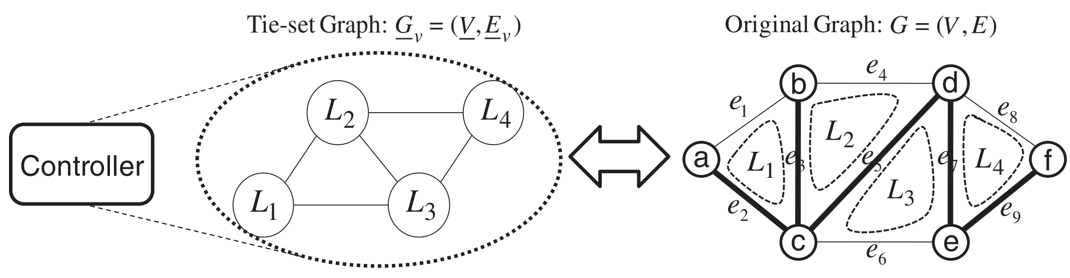


Figure 5. Control model of a power network.

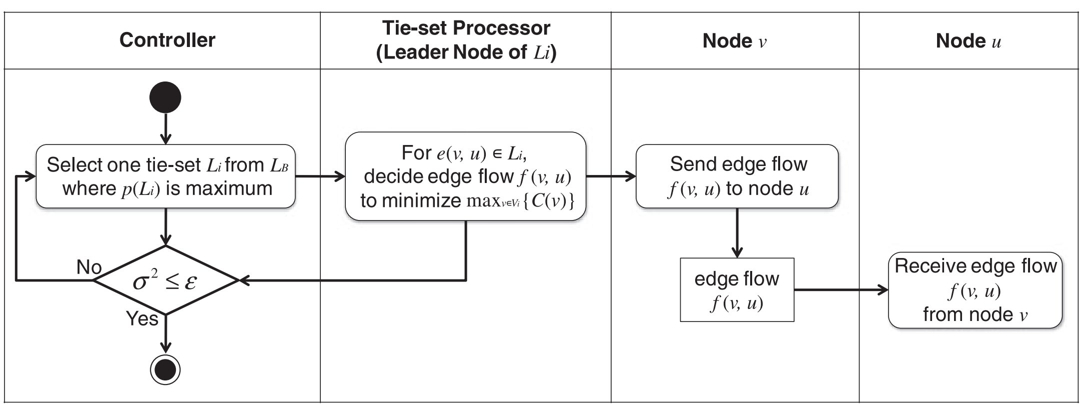


Figure 6. Flow chart of overall processes.

DOI: 10.1002/cta

A controller sends a message to request NV homogenization (Figure 7(1)). Next, a leader node of a selected tie-set L_i decides $f(v, u)$ on each edge $e(v, u) \in L_i$ to minimize $\max_{v \in V_i}\{C(v)\}$ in L_i. Then, a tie-set processor sends a message to node v to produce an edge flow $f(v, u)$ (Figure 7(1.1)). Before node v sends an edge flow to node u, v sends a confirmation message to u (Figure 7(1.1.1)). After homogenizing all the NVs in a tie-set L_i, the controller checks whether $\sigma^2 \leq \varepsilon$ is satisfied where σ^2 is the variance of all the NVs defined in Equation 6. If $\sigma^2 \leq \varepsilon$ is not satisfied, the controller continues to select a tie-set L_i and homogenize its NVs.

To preclude contradiction among tie-sets, if L_i is in the middle of processing, surrounding tie-sets with relation of $\underline{e}_v(L_i, L_j)$ should stand by. This synchronization is realized by using flag concept. If flags of adjacent tie-sets are set to 0, a tie-set must wait for their completion of procedures. After confirming that all the flags of adjacent tie-sets are set to 1, the tie-set on standby begins its procedure. If flags are used in the procedure, parallel selection of more than one tie-set is feasible, although only one tie-set is selected in Figure 6.

4.3. Distributed optimization algorithm in a tie-set

Let V_i be the set of all the nodes included in a fundamental cycle $L(l_i)$ whose fundamental tie-set is L_i. The distributed optimization algorithm to minimize $\max_{v \in V_i}\{C(v)\}$ in L_i is shown in Figure 8. This optimization is conducted when a tie-set L_i is selected by a controller and requested to execute NV homogenization in L_i. In Step0, the average value $\bar{C}\left(= \bar{C}(L_i) = \frac{\sum_{v \in V_i} C(v)}{|V_i|}\right)$ is calculated. Let X be a set of nodes where $C(v) > \bar{C}$ and Y be a set of nodes where $C(v) < \bar{C}$. In Step1, each node of V_i is classified into X or Y. In Step2, $f_{P(x,y)}$ is decided, and thus, edge flows over edges on the path $P(x, y)$ are updated according to Equation 3. Although there are two paths from x to y in a tie-set L_i, an arbitrary path is selected from those two paths as transduction loss and transmission loss are not dealt with in this paper. All the NVs of V_i are also homogenized according to the flows decided before following Equation 4. For each execution of the while sentence in Figure 8, the value of either node in X or node in Y becomes $\bar{C}$. Therefore, procedure in Step2 is $O(|V_i|) = O(|L_i|) = O(D)$ where D is the diameter of a graph. In Step3, each edge flow is calculated and actually distributed to L_i.

Let us explain the distributed optimization algorithm in a tie-set described in Figure 8 by using L_3 of Figure 9. In Step0, the average of NVs $\bar{C}$ is calculated in L_3, thereby $\bar{C} = 51.5$. In Step1, e and f are classified into X, and c and d are classified into Y.

In Step2, as $X \cup Y = \{c, d, e, f\} \neq \emptyset$, let e be a node from X and c be a node from Y. If $C(e) - \bar{C} = 92 - 51.5 = 40.5 > \bar{C} - C(c) = 51.5 - 24 = 27.5$, $f_{P(e,c)} = 51.5 - 24 = 27.5$. Then, $C(c)$ and $C(e)$

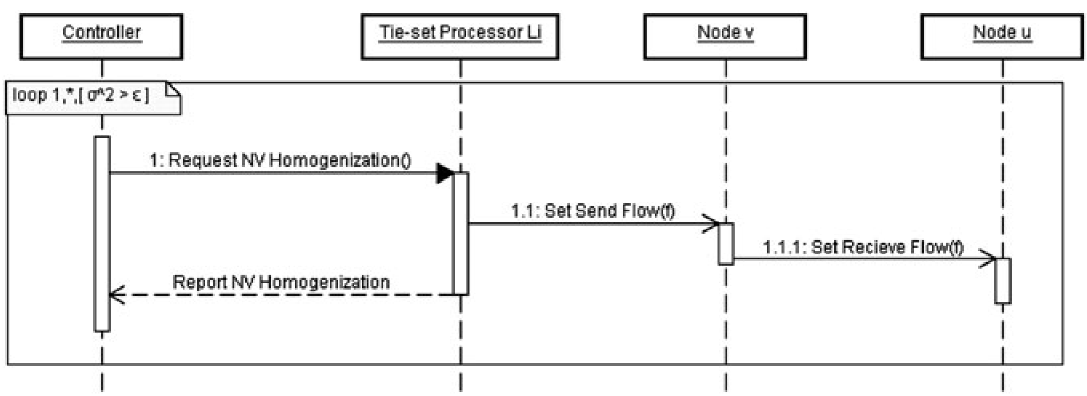


Figure 7. Message flow from a controller to a tie-set processor and nodes.

DOI: 10.1002/cta

Setp0

Calculate the average value $\overline{C}$ of $C(v)$, where $C(v)$ is the node value of $v \in V_i$.

Setp1

If $C(v) > \overline{C}$, classify $v \in V_i$ into a set of nodes X.

Else if $C(v) < \overline{C}$, classify $v \in V_i$ into a set of nodes Y.

Setp2

While $X \cup Y \neq \emptyset$, repeat the following procedure:

Select a node x from X, and a node y from Y arbitrarily.

If $C(x) - \overline{C} > \overline{C} - C(y)$, then

$f_{P(x,y)} = \overline{C} - C(y)$.

Update $C(y)$ to $\overline{C}$, and $C(x)$ to $C(x) + C(y) - \overline{C}$.

Remove y from Y.

Else if $C(x) - \overline{C} < \overline{C} - C(y)$, then

$f_{P(x,y)} = C(x) - \overline{C}$.

Update $C(x)$ to $\overline{C}$, and $C(y)$ to $C(y) + C(x) - \overline{C}$.

Remove x from X.

Else (if $C(x) - \overline{C} = \overline{C} - C(y)$)

$f_{P(x,y)} = C(x) - \overline{C}$.

Update $C(x)$ to $\overline{C}$, and $C(y)$ to $C(y) + C(x) - \overline{C}$.

Remove x from X, y from Y.

Step3

Calculate the value of each edge flow $f(v,u)$ over $e(v,u) \in L_i$ considering all flows decided above, and distribute those flows to the tie-set L_i.

Figure 8. Distributed optimization algorithm in a tie-set L_i.

are updated as $C(c) = 51.5$, $C(e) = 92 + 24 - 51.5 = 64.5$. After that, c is removed from Y. Because $X \cup Y = \{d, e, f\} \neq \varnothing$, let e be a node from X and d be a node from Y. If $C(e) - \bar{C} = 64.5 - 51.5 = 13 < \bar{C} - C(d) = 51.5 - 32 = 19.5$, $f_{P(e,d)} = 64.5 - 51.5 = 13$. Then, $C(d)$ and $C(e)$ are updated as $C(d) = 32 + 64.5 - 51.5 = 45$, $C(e) = 51.5$. After that, e is removed from X. Because $X \cup Y = \{d, f\} \neq \varnothing$, let f be a node from X and d be a node from Y. If $C(f) - \bar{C} = 58 - 51.5 = 6.5 = \bar{C} - C(d) = 51.5 - 45 = 6.5$, $f_{P(f,d)} = 58 - 51.5 = 6.5$. Then, $C(d)$ and $C(f)$ are updated as $C(d) = 45 + 58 - 51.5 = 51.5$, $C(f) = 51.5$. After that, f is removed from X and d is removed from Y. Because $X \cup Y = \varnothing$, Step2 is finished.

In Step3, the value of each edge flow is calculated on the basis of all the flows decided in Step2. As $f_{P(e,c)} = 27.5$, $f(e,f) = 0 + 27.5 = 27.5$, $f(f,c) = 0 + 27.5 = 27.5$. As $f_{P(e,d)} = 13$, $f(e,d) = 0 + 13 = 13$. As $f_{P(f,d)} = 58 - 51.5 = 6.5$, $f(e,f) = 27.5 + (-6.5) = 21$, $f(e,d) = 13 + 6.5 = 19.5$. As a result, $f(d,c) = 0$, $f(e,d) = 19.5$, $f(e,f) = 21$, $f(f,c) = 27.5$. Then, those flows are distributed as shown in Figure 9.

Until $\sigma^2 \leq \varepsilon$, a controller continues to select a tie-set and repeats the optimization in a tie-set.

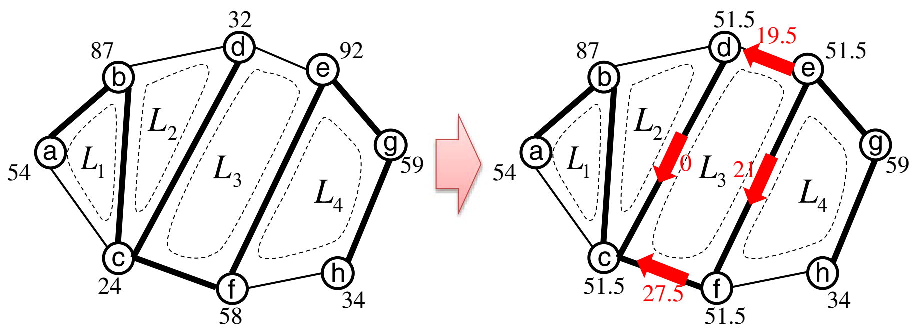


Figure 9. Optimization in a tie-set L_3.

DOI: 10.1002/cta

4.4. Theoretical analysis

It is important to prove that all the NVs converge on the average value by executing the proposed algorithm and to analyze the speed of the convergence, although it has not been solved yet. Let us suggest one result that seems to be useful to the solution of the study.

Let X and Y be a tie-set with u nodes and v nodes, respectively. X and Y are connected with w nodes as shown in Figure 10. Let V_X and V_Y be a set of all nodes in a fundamental cycle that corresponds to X and Y, respectively. Then, the delta δ_X and δ_Y are respectively defined as follows:

$$\delta_X = C^X_{\max} - C^X_{\min}, \left(C^X_{\max} = \max_{v \in V_X} \{C(v)\}, C^X_{\min} = \min_{v \in V_X} \{C(v)\} \right) \tag{8}$$

$$\delta_Y = C^Y_{\max} - C^Y_{\min}, \left(C^Y_{\max} = \max_{v \in V_Y} \{C(v)\}, C^Y_{\min} = \min_{v \in V_Y} \{C(v)\} \right) \tag{9}$$

Let δ'_Y be the delta of Y after X is averaged. The relationship of those deltas is shown in Table I. Then, the following inequality can be derived.

$$\delta'_Y \leq \left(1 - \frac{w}{u}\right)\delta_X + \delta_Y \tag{10}$$

If a tie-set X is selected and its NVs are averaged, there is a possibility that the delta of tie-set Y adjacent to X increases. However, the increase is at most $(1 - w/u)\delta_X$. Therefore, the increase of δ_Y by averaging NVs of V_X propagates a tie-set to tie-set with exponential decrease.

As the propagation becomes widely branched and spreads to an entire network, decrease of the total delta of an entire network cannot be proved by the inequality mentioned previously. Deeper analysis is required to prove the convergence of all the NVs.

5. SIMULATION AND EXPERIMENTS

A simulator is made to realize a distributed algorithm to solve the NV homogenization described in Section 4 and to conduct experiments. As to an experimental environment, we used Microsoft Windows XP as an operating system and Java as a development language. In configuring a network, links are set to be undirected through which electricity can flow bidirectionally. In addition, network is designed to be redundant or biconnected from a graph theoretical standpoint. All NVs are real numbers and are randomly assigned between 0 and 100. As for node configuration, each node has

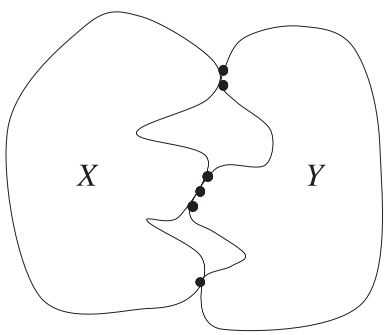


Figure 10. Tie-set X and Y connected with w nodes.

Table I. The relationship of deltas before and after X is averaged.

	Delta of X	Delta of Y
Before	δ_X	δ_Y
After	0	δ'_Y

DOI: 10.1002/cta

input ports and output ports, a message buffer, and a processor. Common buffering method is employed in a simulation node, where all messages received through input ports go to the message buffer. The processor takes each message from the message buffer by polling method. After each message is processed in the processor, the message is sent to other nodes through appropriate output ports unless it is received or discarded.

Let $N(L_i)$ be the number of times that a tie-set $L_i(\in \mathbf{L}_B)$ executes the distributed optimization algorithm described in Figure 8. Then, N_{Total}, $N_{\max}$, and $N_{\min}$ are respectively defined as follows:

$$N_{\mathrm{Total}} = \sum_{L_i \in \mathbf{L}_B} N(L_i) \tag{11}$$

$$N_{\max} = \max_{L_i \in \mathbf{L}_B} \{N(L_i)\} \tag{12}$$

$$N_{\min} = \min_{L_i \in \mathbf{L}_B} \{N(L_i)\} \tag{13}$$

Let t be the computational time that completes the NV homogenization defined previously. In detail, computational time is the time from start point to finish point of the processes described in Figure 6. In the experiments, $\varepsilon = 1 \times 10^{-6}$ and thus, the procedure is iterated until $\sigma^2 \leq 1 \times 10^{-6}$.

5.1. *Simulation with a 4-node network*

Figure 11 is the simulation result in a 4-node network as shown in Figure 12. In this experiment, the tie-set evaluation function is $p(L_i) = $ Random. In this 4-node network, two fundamental tie-sets, $L_0 = \{e(n1, n0), e(n2, n1), e(n0, n2)\}$ and $L_1 = \{e(n0, n3), e(n3, n2), e(n0, n2)\}$, exist. In tie-set L_0 and L_1, five times of optimization described in Figure 8 is respectively executed. After those procedures, all the NVs converged on the average value and the values of all the edge flows are decided. In Figure 11, F of each edge stands for the total amount of edge flows after the procedure shown in Figure 6. On the basis of the result of Figure 11, edge flows after the procedure are described in Figure 12.

5.2. *Behavior of convergence*

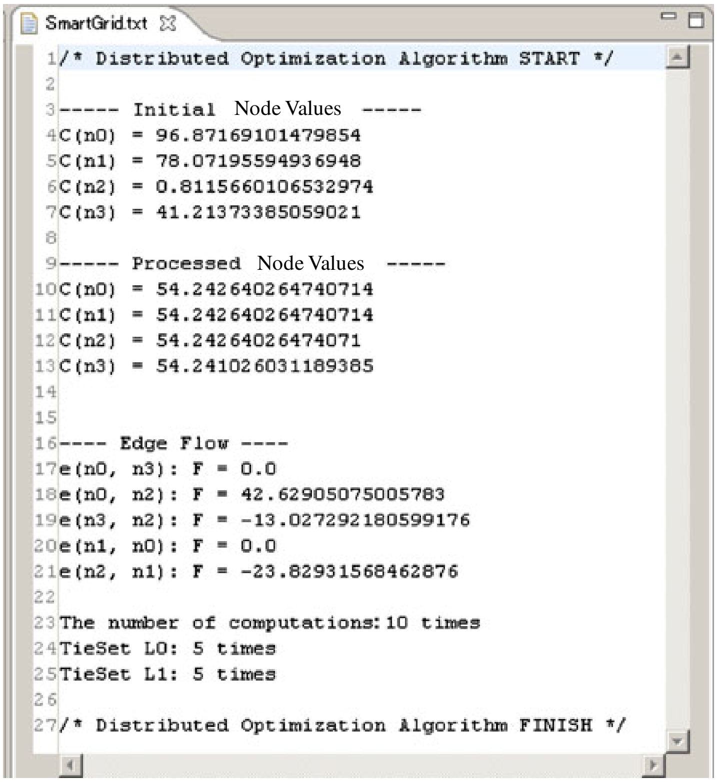
As mentioned, the NV homogenization problem can be solved in local optimization in μ-dimensional liner vector space. By conducting the distributed control method explained in Section 4, all of the NVs in a graph gradually converge on the average value $\bar{C}$. We demonstrated the convergence of all the NVs by simulation experiments.

5.2.1. Convergence in 100 nodes. Here, $G = (V, E)$ is created at random with 100 nodes and 193 links. We first conducted experiments in the condition where $p(L_i) = $ Random. Figure 13 shows the process of convergence to the average value $\bar{C}$ by executing the distributed algorithm. The NVs are plotted when N_{Total} becomes 0, 50, 100, 150, 200, 250, and 300, respectively. As N_{Total} increases, all the NVs gradually converge on $\bar{C}$. Figure 14 shows the process of change of the maximum NV $C_{\max}\left(= \max_{v \in V}\{C(v)\}\right)$ and the minimum NV $C_{\min}\left(= \min_{v \in V}\{C(v)\}\right)$ of Figure 13. Around 300 times of total computing, $C_{\max}$ and $C_{\min}$ become almost the same and thus solved min–max problem defined in Equation 5. In this experiment with 100 nodes, $N_{\mathrm{Total}} = 1322$, $N_{\max} = 22$, and $N_{\min} = 6$ after $\sigma^2 \leq 1 \times 10^{-6}$ is satisfied. Computational time t is 78 ms.

This is the result in the condition where $p(L_i) = $ Random. From this result, it is demonstrated that the solution of the NV homogenization problem does not depend on the order of tie-set selection. This result is strongly related to the realization of global optimization by parallel processing in independent tie-sets where several tie-sets are randomly processed in parallel.

5.2.2. Convergence in 1000 nodes. Next, we conducted simulation in $G = (V, E)$ created at random with 1000 nodes and 1991 links as shown in Figure 15. In this experiment, processes are iterated until $\sigma^2 \leq 1 \times 10^{-6}$. $p(L_i)$ is also set at random. Figure 16 shows the convergence of all the NVs in the 1000-node simulation. In this simulation with 1000 nodes, $N_{\mathrm{Total}} = 13098$, $N_{\max} = 25$, and

DOI: 10.1002/cta



SmartGrid.txt

```
/* Distributed Optimization Algorithm START */

----- Initial Node Values -----
C(n0) = 96.87169101479854
C(n1) = 78.07195594936948
C(n2) = 0.8115660106532974
C(n3) = 41.21373385059021

----- Processed Node Values -----
C(n0) = 54.242640264740714
C(n1) = 54.242640264740714
C(n2) = 54.2426402647407l
C(n3) = 54.241026031189385


---- Edge Flow ----
e(n0, n3): F = 0.0
e(n0, n2): F = 42.62905075005783
e(n3, n2): F = -13.027292180599176
e(n1, n0): F = 0.0
e(n2, n1): F = -23.82931568462876

The number of computations:10 times
TieSet L0: 5 times
TieSet L1: 5 times

/* Distributed Optimization Algorithm FINISH */
```

Figure 11. Simulation result in a 4-node network.

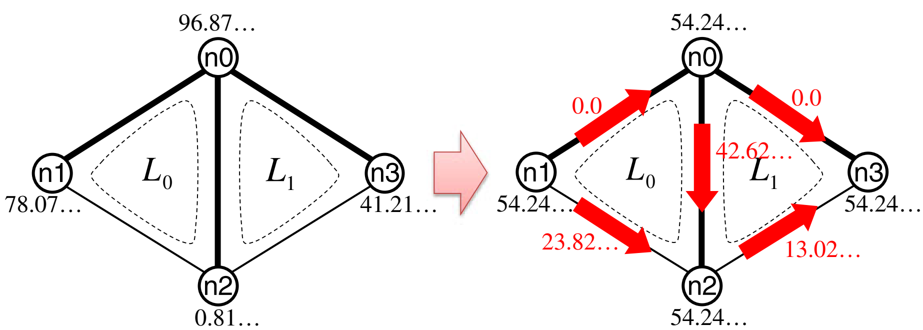


Figure 12. Edge flows of the 4-node simulation.

$N_{\min} = 4$. Computational time that completes the processes described in Figure 6 is 6797 ms. Those results show that local optimization based on tie-set units leads to global optimization. These experiments are the results of selecting a tie-set at random. Therefore, it is conceivable that picking up one tie-set at random makes N_{Total} unnecessarily increased. Then, we compare experimental results, changing evaluation function $p(L_i)$.

5.3. *Comparison experiments focusing on tie-set evaluation function $p(L_i)$*

Next, we conducted experiments to count the number of times of computations N_{Total}, $N_{\max}$, $N_{\min}$, and computational time t. Computational time t to complete the processes is described in Figure 6. Given $G = (V, E)$ is the same as the graph used in the previous experiment in Section 5.2.2, where $|V| = 1000$, $|E| = 1991$. In this experiment, ε is also set to 1×10^{-6}. Then, we conducted simulation by changing

DOI: 10.1002/cta

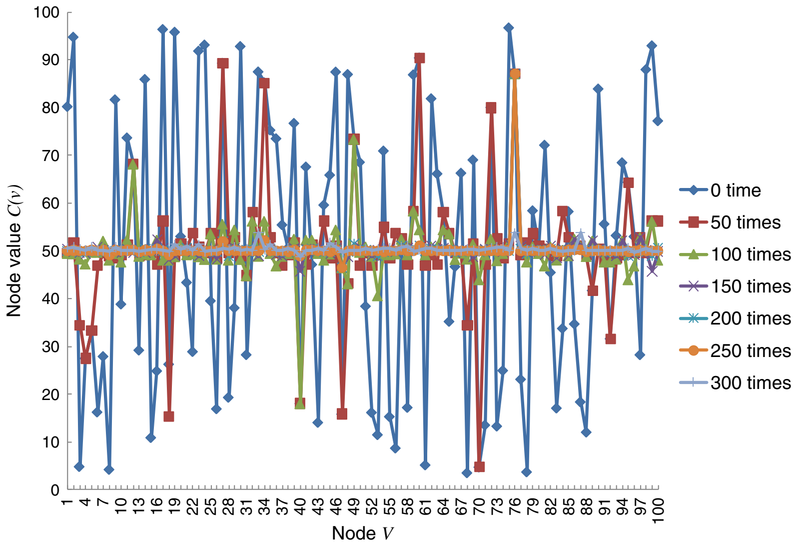


Figure 13. Convergence of all the node values (100 nodes).

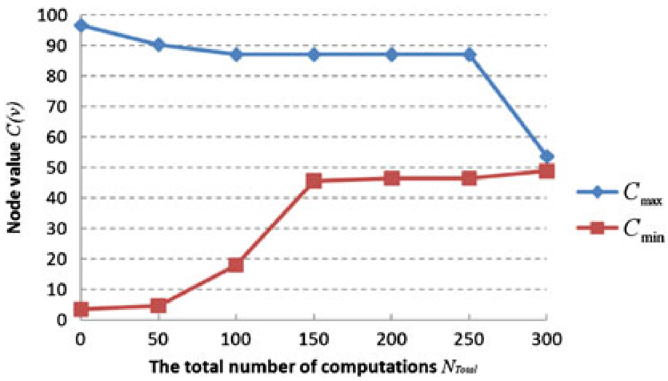


Figure 14. Convergence of C_{max} and C_{min} (100 nodes).

tie-set evaluation function $p(L_i)$. $p(L_i)$ is defined in Section 4.2. For each execution, N_{Total}, N_{max}, N_{min}, and t are measured when $\sigma^2 \leq 1 \times 10^{-6}$ is satisfied. Experiments are conducted 10 times for each $p(L_i)$, and the average value, the worst value, and the best value are calculated.

As shown in Table II, when $p(L_i) = \text{Random}$, the average value of N_{Total}, N_{max}, and N_{min} is generally larger than the other two evaluation functions. In addition, the difference between the worst value (18,541) and the best value (12,735) of N_{Total} is large, and thus, convergence can be unstable. When $p(L_i) = C_{max} - C_{min}$ and $p(L_i) = \sigma^2$, the difference between the worst and best value of N_{Total} is small. Therefore, convergence is stable compared with random selection. There exist some tie-sets that do nothing ($N_{min} = 0$) when $p(L_i) = C_{max} - C_{min}$ and $p(L_i) = \sigma^2$. The average value of N_{Total} by $p(L_i) = C_{max} - C_{min}$ is a little worse than $p(L_i) = \sigma^2$. However, when $p(L_i) = C_{max} - C_{min}$, computation time t is the best among three functions.

5.4. *Experiments from 1000 to 4000 nodes*

To examine the behavior for increase in the number of nodes, we conducted experiments with 1000, 2000, 3000, and 4000 nodes whose number of links is about 2000, 4000, 6000, and 8000, respectively. Each G is created at random. For each network configuration, N_{Total}, N_{max}, N_{min}, and computational time t are measured when $\sigma^2 \leq 1 \times 10^{-6}$ is satisfied. Experiments are conducted 10 times for each network configuration, and the average is calculated. We executed the algorithm for

DOI: 10.1002/cta

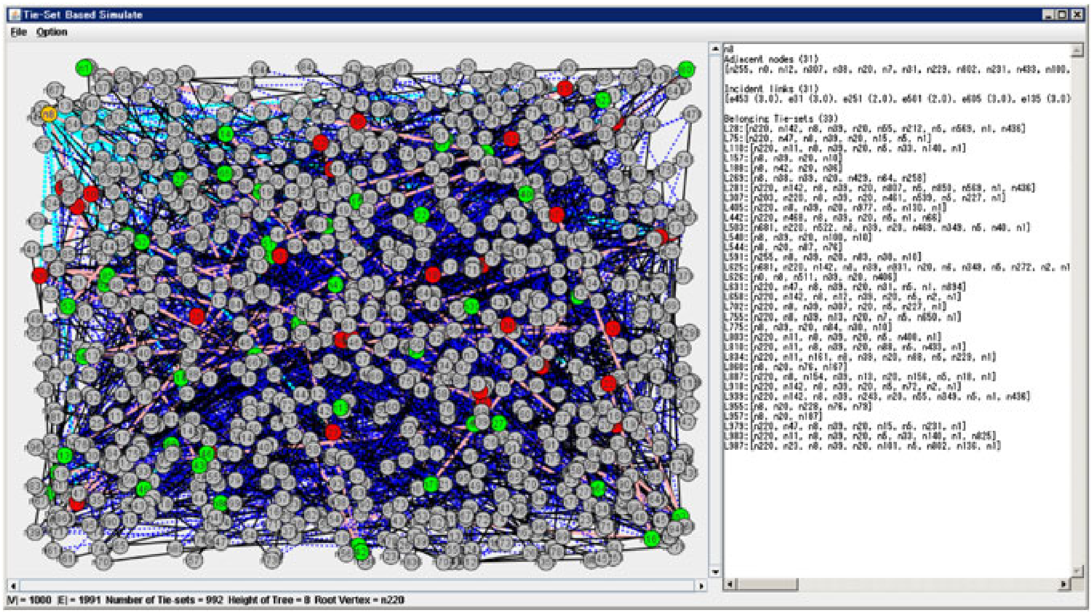


Figure 15. 1000-node simulation network.

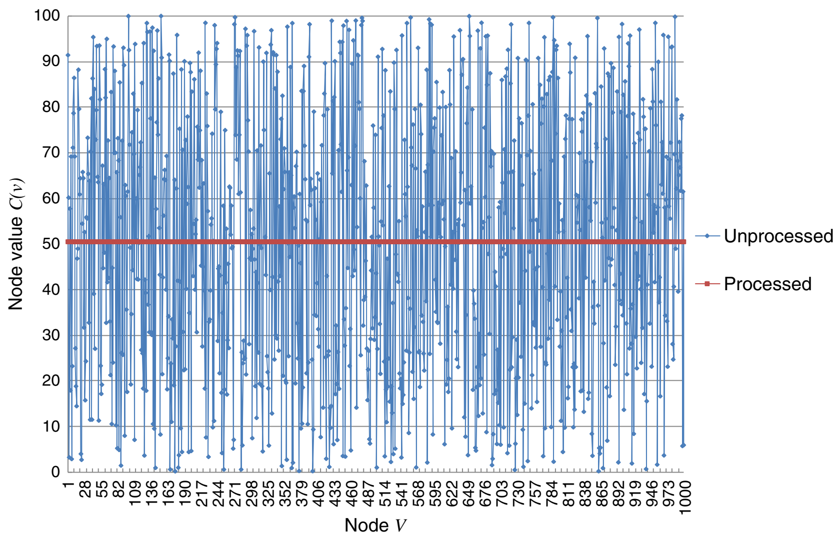


Figure 16. Convergence of all the node values (1000 nodes).

all $p(L_i)$ defined in Section 4.2. Through Figures 17–19, 'Random' stands for $p(L_i) = \text{Random}$, 'Delta' stands for $p(L_i) = C_{\max} - C_{\min}$, and 'Variance' stands for $p(L_i) = \sigma^2$.

Figure 17 shows the average of N_{Total} with networks consisting of 1000–4000 nodes. As shown in Figure 17, the total number of computation times and the number of nodes are in proportional relationship.

However, $N_{\max}$ does not change even though a network becomes large as shown in Figure 18. This result indicates that distributed control method based on tie-set graph is suitable to manage a large-scale network because processing load of each tie-set does not increase.

Figure 19 is the average of $N_{\min}$ by tie-set evaluation function $p(L_i)$ in 1000–4000 nodes. As shown in Figure 19, there exist some tie-sets that execute nothing even in a 4000-node network when $p(L_i) = C_{\max} - C_{\min}$ and $p(L_i) = \sigma^2$.

Figure 20 shows computation time t by tie-set evaluation function $p(L_i)$ in 1000–4000 nodes. Computation time is almost proportionate to the number of nodes.

DOI: 10.1002/cta

Table II. Comparison experiment of $p(L_i)$ by N_{Total}, $N_{\max}$, $N_{\min}$, and t with a 1000-node network.

N_{Total}	$p(L_i) = \mathrm{Random}$	$p(L_i) = C_{\max} - C_{\min}$	$p(L_i) = \sigma^2$
Average	14217.2	2651.6	2515.1
Worst	18,541	2826	2573
Best	12,735	2520	2196
$N_{\max}$	$p(L_i) = \mathrm{Random}$	$p(L_i) = C_{\max} - C_{\min}$	$p(L_i) = \sigma^2$
Average	28.7	10.2	16.2
Worst	36	12	21
Best	25	8	13
$N_{\min}$	$p(L_i) = \mathrm{Random}$	$p(L_i) = C_{\max} - C_{\min}$	$p(L_i) = \sigma^2$
Average	4	0	0
Worst	6	0	0
Best	2	0	0
t (ms)	$p(L_i) = \mathrm{Random}$	$p(L_i) = C_{\max} - C_{\min}$	$p(L_i) = \sigma^2$
Average	3992.2	2878.1	6293.19
Worst	5195	3026	6531
Best	3541	2777	6047

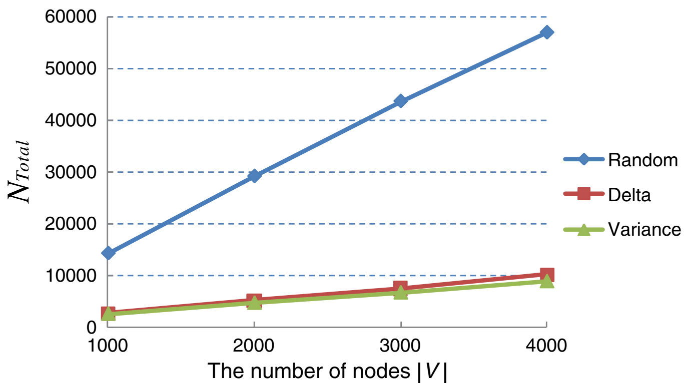


Figure 17. N_{Total} by tie-set evaluation function $p(L_i)$ (1000–4000 nodes).

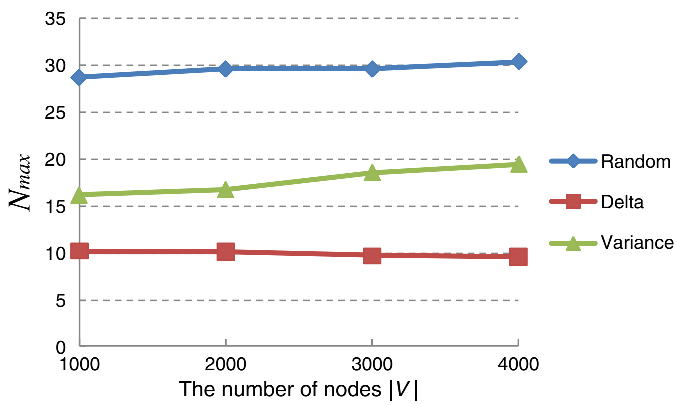


Figure 18. $N_{\max}$ by tie-set evaluation function $p(L_i)$ (1000–4000 nodes).

DOI: 10.1002/cta

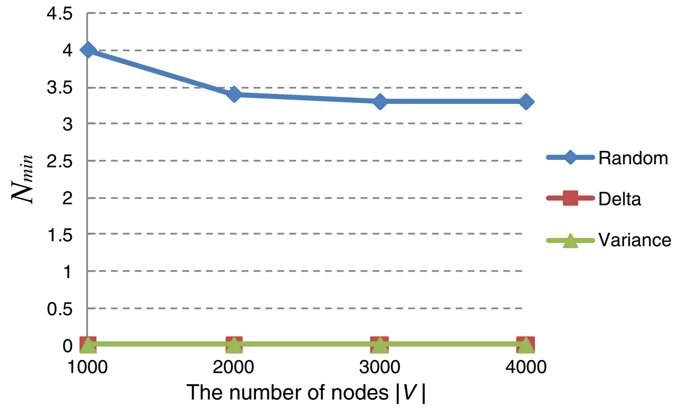


Figure 19. $N_{\min}$ by tie-set evaluation function $p(L_i)$ (1000–4000 nodes).

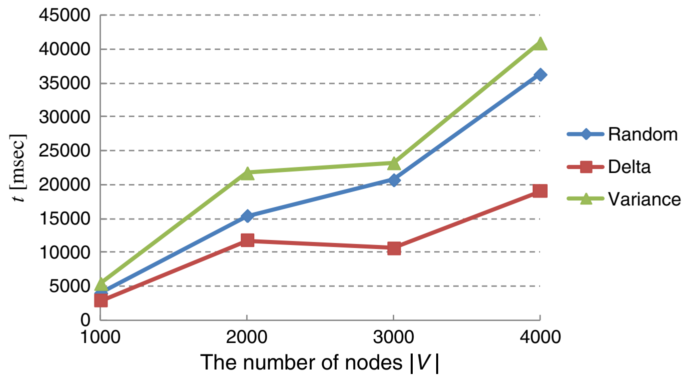


Figure 20. Computational time t by tie-set evaluation function $p(L_i)$ (1000–4000 nodes).

6. CONCLUSION AND FUTURE WORKS TO BE SOLVED

In this paper, tie-set graph theory is first introduced as a theoretical approach. Then, we formulate balanced allocation problem of distributed energy resources as an NV homogenization problem and propose a distributed control method for the problem focusing on smart meters. The results of simulations show that local optimization by tie-sets leads to global optimization. Although each tie-set has its limited local information, an entire network is controlled in an orderly fashion because of the theoretical basis of a tie-set graph. Furthermore, a series of local information of each node is consistent with the condition of an entire network. This is because the structure of tie-sets is uniquely determined by the graph theoretical concept of a tie-set basis implicitly underlies a network. Future works to be solved are as follows.

6.1. *Proof of convergence of all the node values*

This paper proves that the delta of NVs of two tie-sets after homogenization is narrowed down to a certain small rate. Therefore, there is a possibility to exponentially decrease the delta of NVs if homogenization is repeated. However, it has not yet been proved strictly, and thus, the proof becomes one of the most important issues. It is also important to analyze the behavior of convergence with a variety of network topologies and to extract prerequisites of proof.

6.2. *Realization of completely autonomous parallel distributed control*

In this paper, a controller centralizes data of smart meters and issues an order to a selected tie-set. In addition, NVs are derived from a snapshot at a given time. However, renewable distributed energies

DOI: 10.1002/cta

are intermittent and dynamically fluctuate depending on weather conditions. Therefore, it is indispensable to develop completely autonomous distributed control architecture by allocating a processor to each tie-set where each tie-set processor constantly communicates with adjacent tie-sets and conducts local optimization in parallel.

The important work is to virtually realize a controller on the basis of distributed μ processors on vertices of a tie-set graph. Then, this work will lead to the solution of the global optimization problem by completely autonomous local control as indicated in the result of random tie-set selection. As mentioned, ON/OFF operation of each tie-set controller focusing on flag concept will enable the completely autonomous parallel distributed architecture.

We will conduct more extensive simulations focusing on scalability, variations of network topologies, and properties of power grids. This paper does not consider the properties of power systems such as transmission loss, conversion loss, and so on. Therefore, we need to add those factors to problem formulation as well as simulation conditions to conduct more pragmatic experiments.

ACKNOWLEDGMENTS

The authors would like to thank Professors Lubomir Bic, Michael Dillencourt, Nalini Venkatasubramanian, and Elaheh Bozorgzadeh at University of California, Irvine, for their valuable comments and suggestions on this research. The authors also acknowledge Professor Emeritus Kojiro Kobayashi at Soka University with all our heart for his valuable advice and suggestions. This work benefited from fruitful discussions with researchers at Stella Environment Science Corporation who are creating a Smart Grid Network in Niue in the South Pacific Ocean.

REFERENCES

1. Mazza P. Powering up the smart grid, a northwest initiative for job creation, energy security and clean, affordable electricity. *A Special Report from Climate Solutions*, July 2005.
2. Bakken DE, Hauser CH, Gjermundrod H, Bose A. Towards more flexible and robust data delivery for monitoring and control of the electric power grid. www.gridstat.net/TR-GS-009.pdf, May 30, 2007.
3. Locke G, Gallagher PD. NIST framework and roadmap for smart grid interoperability standards, release 1.0. *NIST Special Publication 1108*, January 2010.
4. He MM, Reutzel EM, Jiang X, Katz RH, Sanders SR, Culler DE, Lutz K. An architecture for local energy generation, distribution, and sharing. *IEEE Energy 2030 Conference* 2008; 1–6. DOI: 10.1109/ENERGY.2008.4781028
5. Jiang X, Dawson-Haggerty S, Dutta P, Culler D. Design and implementation of a high-fidelity AC metering network. *International Conference on Information Processing in Sensor Networks (IPSN)* 2009; 253–264.
6. Momoh JA. Smart grid design for efficient and flexible power networks operation and control. *IEEE/PES Power Systems Conference and Exposition (PSCE)* 2009; 1–8. DOI: 10.1109/PSCE.2009.4840074
7. Hsu J, Zahedi S, Kansal A, Srivastava M, Raghunathan V. Adaptive duty cycling for energy harvesting systems. *Proceedings of the 2006 International Symposium on Low Power Electronics and Design (ISLPED'06)* 2006; 180–185. DOI: 10.1109/LPE.2006.4271832.
8. Batlle C, Doria-Cerezo A, Fossas E. Bidirectional power flow control of a power converter using passive Hamiltonian techniques. *International journal of circuit theory and applications* 2008; **36**:769–788. DOI: 10.1002/cta.459.
9. Hui S, Chung H, Yip S. A bidirectional ac-dc power converter with power factor correction. *IEEE Transactions on Power Electronics* 2000; **15**(5):942.949.
10. Heimbuch J. Smart grids can cut CO2 emissions by 15 percent. *SMART 2020 Report*, October 9, 2008.
11. Moslehi K, Kumar R. A reliability perspective of the smart grid. *IEEE Transactions on Smart Grid* 2010; 57–64. DIO: 10.1109/TSG.2010.2046346.
12. Gerwen RV, Jaarsma S, Wilhite R. Smart metering. *Distributed Generation*, www.leonardo-energy.org, July 2006.
13. Singh T, Vara PK. Smart metering the clouds. *18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'09)* 2009; 66–71. DOI: 10.1109/WETICE.2009.49
14. Shokooh S, Khandelwal T, Shokooh F, Tastet J, Dai JJ. Intelligent load shedding, need for a fast and optimal solution. *IEEE Petroleum and Chemical Industry Committee (PCIC) Europe* 2005.
15. Kim HM, Kinoshita T, Shin MC. A multiagent system for autonomous operation of islanded microgrids based on a power market environment. *Energies* 2010; **3**(12):1972–1990. DOI: 10.3390/en3121972
16. Shinomiya N, Koide T, Watanabe H. A theory of tie-set graph and its application to information network management. *International Journal of Circuit Theory and Applications* 2001; **29**:367–379. DOI: 10.1002/cta.155.
17. Understanding SONET UPSRs; http://www.sonet.com/EDU/upsr.htm, Available online.
18. Understanding SONET BLSRs; http://www.sonet.com/EDU/blsr.htm, Available online.
19. Shah S, Yip M. Extreme networksf ethernet automatic protection switching (EAPS) version 1. *Network Working Group, Request for Comments*: 3619, October 2003.
20. Wasem OJ. An algorithm for designing rings for survivable fiber networks. *IEEE Transactions on Reliability* 1991; **40**:428–439. DOI:10.1109/24.93762

21. Gardner LM, Heydari M, Shah J, Sudborough IH, Tolis IG, Xia C. Techniques for finding ring covers in survivable networks. Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'94). *Communications: The Global Bridge* 1994; **3**:1862–1866. DOI: 10.1109/GLOCOM.1994.513193
22. Thomassen C. On the complexity of finding a minimum cycle cover of a graph. *SIAM Journal on Computing* 1997; **26**(3):675–677. DOI: 10.1137/S0097539794267255
23. Erol-Kantarci M, Kantarci B, Mouftah HT. Reliable overlay topology design for the smart microgrid network. *IEEE Network* 2011; **25**(5): 38–43, DOI: 10.1109/MNET.2011.6033034
24. Okada N. A method to determine the distributed control setting of looping devices for active distribution systems. *IEEE Bucharest PowerTech* 2009; **2009**; 1–6. DOI: 10.1109/PTC.2009.5281982
25. Shinomiya N, Koide T, Kuroha M, Watanabe H. Tie-set graph, meta-tie-set graph and information network. *IEICE/IEEK Proceedings of ITC-CSCC'99* 1999; **2**:872–875.
26. Malinowski J. A new efficient algorithm for generating all minimal tie-sets connecting selected nodes in a mesh-structured network. *IEEE Transactions on Reliability* 2010; **59**(1): 203–211. DOI: 10.1109/TR.2009.2036712
27. Koide T, Kubo T, Watanabe H. A study on the tie-set graph theory and network flow optimization problems. *International Journal of Circuit Theory and Applications* 2004; **32**:447–470. DOI: 10.1002/cta.275
28. Nakayama K, Shinomiya N. Distributed control based on Tie-set Graph Theory for smart grid networks. *Proceedings of 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)* 2010; 957–964. DOI: 10.1109/ICUMT.2010.5676490
29. Nakayama K, Shinomiya N. Autonomous recovery for link failure based on tie-sets in information networks. *Proceedings of 2011 IEEE Symposium on Computers and Communications (ISCC)* 2011; 671–676. DOI: 10.1109/ISCC.2011.5983916
30. Kadena K, Nakayama K, Shinomiya N. Network failure recovery with tie-sets. *Proceedings of 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)* 2011; 467–472, DOI: 10.1109/WAINA.2011.125
31. Nakayama K, Shinomiya N, Watanabe H. Distributed control for link failure based on tie-sets in information networks. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)* 2010; 3913–3916, DOI: 10.1109/ISCAS.2010.5537682.
32. Iri M, Shirakawa I, Kajitani Y, Shinoda S, et al. Graph theory with exercises. *CORONA Pub: Japan* 1983.
33. Pacific Crest Mosaic. Mesh networks is communications winner in utility survey. *Pacific Crest Mosaic Smart Grid Survey-August* 2009.
34. Fan JY, Zhang L, McDonald, JD. Distribution network reconfiguration: single loop optimization. *IEEE transactions on Power Systems* 1996; **11**(3):1643–1647. DOI: 10.1109/59.535709