Distributed Control Based on Tie-Set Graph Theory for Smart Grid Networks

Kiyoshi Nakayama, *Student Member, IEEE*, Norihiko Shinomiya, *Member, IEEE*
Graduate School of Engineering, Soka University, Tokyo, 192-8577, Japan
Email: {kiyoshi.nakayama, shinomi}@ieee.org

***Abstract*—This paper introduces tie-set graph theory and its application to smart grid networks. Tie-sets stand for loop structure in a network. Tie-set graph theory, which aims to realize distributed control in local units, provides the theoretical nature of loops. In mesh topological networks, all the nodes and links are covered with a fundamental system of tie-sets, which is created by a tree defined by graph theory. If tie-sets are used as local management units in a network, an entire network can be controlled in an orderly fashion due to the graph theoretical basis. Distributed control based on tie-sets has some merit such as easy communication and reduction of electricity loss. In this paper, we propose the unique graph theory on tie-sets, its relationship with smart grid networks, and autonomous configurations based on tie-sets. In addition, we suggest distributed algorithms for an electricity distribution problem as a conceivable issue, and basic simulations and experiments are also conducted.**

***Index Terms*—graph theory, loop management, tie-set, smart grid, electricity distribution**

I. Introduction

Information technology is now being used throughout power grids. This represents the most profound electrical power revolution in a century. Power grid is 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]. The most remarkable change in electric power grids is that demanders such as houses and factories become suppliers installing equipment which generates electricity taking advantage of natural energies. By that change, controlling interactive flowing of both electricity and information on smart grid networks will be required in the future. However, the amount of electricity generated by natural energy fluctuates depending on weather condition. Therefore, a function that can track the status of use of electricity in each facility has been required. There has been also a need to control and adjust the amount of electricity generated by power stations. To manage electronic systems in each demander such as house and factory, smart meters, which are able to monitor the amount of use and generation of electricity by natural energies, have been developed mainly in Europe and the U.S. A smart meter is an advanced electrical meter that identifies consumption in more detail than a conventional meter, and optionally, but generally, communicates that information via some network back to the local utility for monitoring and billing purposes [2]. However, to date there has been no utilization of advances in distributed computing technologies, that offer much more flexibility and adaptability than can be achieved using network technology [3]. Although services utilizing smart meters are somehow contingent on laws, this paper suggests theoretical approaches to issues on smart grid from a perspective of graph theory.

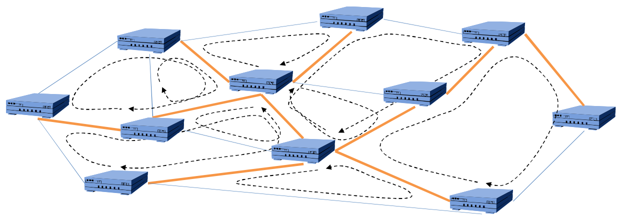


Fig. 1. Autonomous control based on tie-sets

In distributed control, partition units of a network become important. In information networks, there are a lot of techniques that focus on loop or ring structure as partition units. For instance, Unidirectional Path Switched Ring (UPSR) [4] or Bidirectional Line Switched Ring (BLSR) [5] is used in a Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) network. Moreover, Ethernet Automatic Protection Switching or EAPS [6] is utilized in local area networks. If networks on power grids are also managed in local units of tie-sets, more flexible control for distributed energy resources is considered to be possible. Utilization of loop-based management for local units of a power grid is also gathering attention in electricity control. For example, management methods which realize reducing voltage rise and power flow control focusing on loops in a smart grid have been proposed [7]. There are a variety of methods to cover a network with cycles, for example, cycles covering every node in a graph [8], rings covering every link in a network [9], and the smallest total length of cycles covering every edge in a graph [10], etc. Besides those finding methods of cycles, namely ring structure, autonomous distributed configuration of local state information of network nodes still remains as a significant issue of fundamental network management. Although ring or loop management has its great merit in distributed control, its

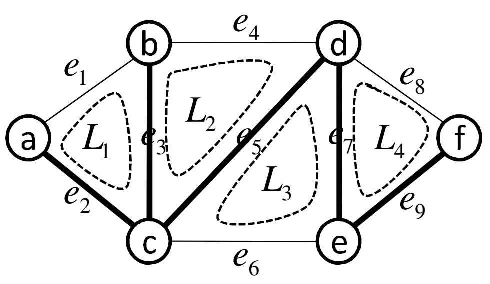


Fig. 2. An example of a fundamental system of tie-sets

application to mesh networks still has its difficulty, especially to large-scale and intricately-intertwined nonplanar networks. Artificially embedding physical rings to mesh networks cannot cope with the natural evolution of network topologies and cannot be applied to logical connected networks such as wireless networks. This paper focuses on the notion of tie-sets defined by graph theory since the theory of tie-sets provides theoretical basis for its application to distributed control. Many authors study on tie-sets 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 [11]. The study introduces tie-set graph theory that provides theoretical nature of loops. The paper [12] 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 work [13] that local optimization by tie-set units leads to global optimization. However, those researches beg the question of how distributed architecture should configure state information based on the tie-sets. In fact, it is quite difficult for a network node to recognize appropriate information of loops with limited peripheral information of adjacent nodes and incident links. Furthermore, segmentation by rings backed up by mathematical basis and orderly distributed control on those units are fraught with complications. This topic becomes a significant issue in this paper. This paper proposes the unique graph theory on tie-sets, its relationship with smart grid networks, and autonomous configurations based on tie-sets. In addition, we suggest distributed algorithms for an electricity distribution problem as a conceivable issue, and basic simulations and experiments are also conducted.

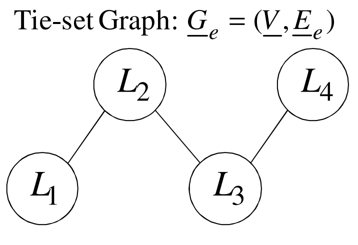
II. Tie-sets and State Information

A. Fundamental System of Circuits and Tie-sets

For a given bi-connected and undirected graph $G = (V, E)$ with a set of vertices $V = \{v_1, v_2, ..., v_n\}$ and a set of edges $E = \{e_1, e_2, ..., e_m\}$, let $L_i = \{e^i_1, e^i_2, ..., e^i_k\}$ be a set of edges which constitutes a loop in G. The set of edges L_i is called a "tie-set" [14]. Let T and $\overline{T}$ be a tree and a cotree of G, respectively, where $\overline{T} = E - T$. $\rho = \rho(G) = |T|$ and $\mu = \mu(G) = |\overline{T}|$ are called the $rank$ and the $nullity$, respectively. A tree T on a graph $G = (V, E)$ is a ultranet set of edges which does not include any tie-set. In other words, for $l \in \overline{T}$, $T \cup \{l\}$ includes one tie-set. Focusing on a subgraph $G_T = (V, T)$ of G and a edge $l = (a, b) \in \overline{T}$, there exists only one elementary path P_T whose origin is b and terminal is a in G_T. Then an elementary circuit which consists of the path P_T and the edge l is uniquely determined as follows:

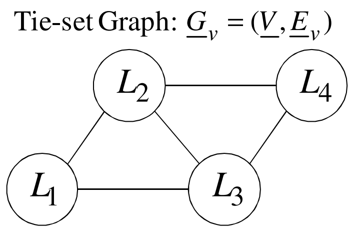
$$\begin{aligned} L(l) &= (a, l = (a, b), P_T(b, a)) \\ &= (a, l, v_0 = b, t_1, v_1, \cdots, t_h, v_h = a) \qquad (1) \end{aligned}$$

In this way, a circuit determined by a edge $l = (a, b) \in \overline{T}$ and a path $P_T(a, b)$ on G_T is denoted as a "fundamental circuit". A simple circuit can be expressed by a set of edges, so called a tie-set. A 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.



Tie-set Graph: $\underline{G}_e = (\underline{V}, \underline{E}_e)$
L_1
L_2
L_3
L_4

Fig. 3. An e-tie-set graph corresponding to tie-sets of Fig.2



Tie-set Graph: $\underline{G}_v = (\underline{V}, \underline{E}_v)$
L_1
L_2
L_3
L_4

Fig. 4. A v-tie-set graph corresponding to tie-sets of Fig.2

B. 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 $\{L_1, L_2, ..., L_\mu\}$, 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) 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 [15] as shown in Fig.3.

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 [15] as shown in Fig.4.

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

C. Relationship between a Tie-set Graph and a Power Grid

The result of the recent Pacific Crest Mosaic Smart Grid Survey [16] shows that when asked about the relative im-

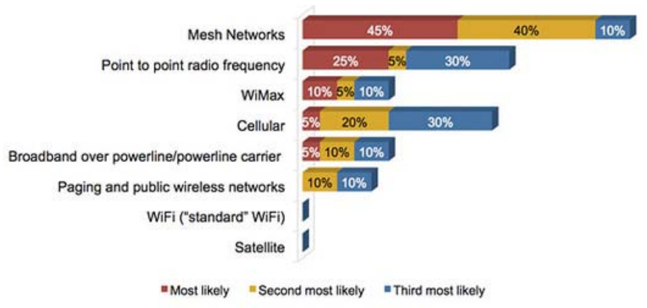


Fig. 5. The result of surveys by Pacific Crest Mosaic

portance of communication technologies, respondents selected mesh networks as the technology they will most likely use in their service territory (45%) as shown in Fig.5. 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 bi-connected 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) Easy Communication: Now that electricity is getting able to flow bi-directionally on a power grid, there is a need to communicate information of such as demands and supplies in a certain unit. Based on loop structures, message passing can be realized within a loop. Therefore, by sending a message in a circle periodically, each node can recognize information of other nodes in a loop.

2) Reduction of Loss of Electricity: The longer the distance between a point of demand and supply becomes, the more the loss of electricity increases. Hence the distance of transfer of electricity should be shortened to minimize the loss. If electricity is supplied within a loop, the distance of power transmission is greatly shortened in comparison with the case that electricity once converses on a substation and is distributed to points of demand.

The relation among the tie-sets is substituted for $\underline{E}$. To avoid discrepancy of information among nodes, G_v is more appropriate than G_e. If procedures are conducted in a certain tie-set, adjacent tie-sets should not conduct any procedures since 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 $e_v(L_i, L_j)$. If a problem which cannot be solved in one tie-set is emerged, the tie-set contacts adjacent tie-sets connected via $\underline{e} \in \underline{E}$ 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.

III. Configuration of Local Information

A. State Information in Each Node

In a network, let us assume that a tree T on $G = (V, E)$ corresponds to communication paths, and a cotree $\overline{T}$ corresponds to non communication paths. In this paper, links representing communication paths are defined as "tree links" which are expressed by thick lines, while links representing non communication paths are defined as "cotree links" which are expressed by thin lines as shown in Fig.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 Fig.2, respectively. Each node n mainly has three types of information as state information as follows:

1) Incident Links: Information of links connected to n.

2) Adjacent Nodes: Information of nodes which are connected through incident links of n.

3) Tie-set Information: Information of fundamental tie-sets to which n belongs. Namely, when a tie-set $L_i \ni n$, it is defined that n belongs to L_i and has its information. Information of a tie-set L_i is as follows:

- Topology structure of a tie-set L_i
- Information of links that constitutes a tie-set L_i
- Information of nodes that is contained in a loop defined by a tie-set L_i

Here is an example of state information of a node c in Fig.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 $\{L_1, L_2, L_3\}$.

B. Algorithm to Configure Tie-set Information

Each node has information of fundamental tie-sets to which the node belongs so that any problems can be resolved within a loop. In order to obtain information of fundamental tie-sets, each node executes an distributed algorithm to recognize fundamental tie-sets. A *Find Tie-set* message, which is used to catch tie-set information, 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 below, each node can hold information of fundamental tie-sets. First, each node n_o creates *Find Tie-set* messages, and then sends those messages to all adjacent nodes of n_o. When sending a *Find Tie-set* message to an adjacent node n_a, n_o adds node information of n_o to *NodeTable*, and adds information of a link connected to both n_o and n_a to *EdgeTable*. Let n_r be a node which received a *Find Tie-set* message. After receiving a *Find Tie-set* message, n_r executes different procedure by the following cases:

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

Case 2: $n_r = n_o$ In this case, the *Find Tie-set* message has passed through 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 n_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. 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.

C. Leader Election Problem in Each Tie-set

In each tie-set, it is imperative to select a leader node that solves dilemma that emerges when using an overlapping resource among other tie-sets. For example, a local control unit defined by tie-sets share 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. Given the state information defined above, major criteria are as follows:

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

The criterion 1) is able to decide a leader node uniquely, while criteria 3) and 2) are 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 sets as a leader node. If a leader node is involved in a lot of tie-sets, communication among the tie-sets is not necessary, thereby communication load is reduced. Since each node has Tie-set Information, it is possible for each 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. In order 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 information as follows:

- Node ID id_m: The largest or smallest node ID among a set of nodes through which a *Decide Leader* message passed.
- Criteria Information $value_m$: The optimal value of some criteria among a set of nodes through which a *Find Tie-set* message passed. The criteria are, for instance, the above criteria 2) and 3).
- *NodeTable*: Addresses of nodes that belong to L_i. A string of addresses must be sorted to satisfy the order as follows: $L_i = (v_0^i, e(v_0^i, v_1^i), v_1^i, \cdots, e(v_h^i, v_0^i), v_0^i)$ where $V_i = \{v_1^i, \ldots, v_h^i\}$ represents vertices of L_i.

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

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

Case 2: $n_r = n_o$ In this case, the *Decide Leader* message has passed through on a tie-set L_i. Then n_o recognizes a node that holds Node ID id_m as a leader of L_i.

D. 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. In order 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 III-A, a leader node n_l^i of a tie-set L_i has information below:

- Adjacent Tie-sets $\boldsymbol{L}_a = \{L_1^a, L_2^a, ...\}$
  Based on the concept of $\Re(L_i, L_j)$ stated in tie-set graph section, an adjacent tie-set L_j of L_i is determined according to the relation of connection $\underline{e}(L_i, L_j) \in \underline{E}_v$ of $\underline{G}_v$.

Let n_l^i and n_l^j be a leader node in L_i and L_j, respectively. When n_l^i in L_i communicates with an adjacent tie-set n_l^j in L_j, n_l^i executes a different procedure according to the following cases.

Case 1: $n_l^i = n_l^j$ In this case, there is no need for n_l^i to send any message since n_l^i itself is the leader of the adjacent tie-set L_j. Instead, n_j^i decides an proper procedure considering state information of both L_i and L_j.

Case 2: $n_l^i \in L_j$ In this case, n_l^i has state information of L_j, thereby only sends a *Tie-set Communication* message to n_l^j using topology information of L_j.

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

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

IV. Approaches to Electricity Distribution Problem

A. Smart Meter and its Function

Advent of smart meters makes it possible to visualize the deficiency and excess of electric power distributed among houses and factories, etc. The functions of a smart meter are as follows [2]:

1) Presence Function: The amount of electricity generated by solar panels and consumed by users is digitalized in real time by using smart meters. 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 wireless communication, ON/OFF operation of equipment and temperature modulation of air conditioning systems can be controlled externally under the direction of an electric power company to adjust electric loads on a power grid.

B. Electricity Distribution Problem in each Tie-set

In this section, let us discuss electricity distribution problems in each tie-set focusing on smart meters and their logical connections. It is conceivable that some houses or facilities are in short of electricity and in need of it. The information of deficiency and excess of electricity at each point is possible to be recognized by using smart meters. If certain points are in need of electricity, other nodes which belong to the same tie-set as well as have excess electricity can distribute power to the points which requires electric power. As a result, futile electric generation in addition to blackout can be prevented if electricity distribution is locally realized. Substations are also able to gather electricity from their domains, and do not need excessive electricity generated by power stations. However, problems such as lack of stored power in substations will arise in case that people make heavy use of electricity. Therefore, there may be a need to control variance of electricity among houses, factories, substations, etc. We solve these problems from a viewpoint of a mathematical optimization problem, and propose distributed algorithms.

Let v_i be a node with a smart meter, and e_k be a logical link that connects between two devices. A logical link means connection by either wired or wireless communication channels. Let $C(v_i)$ be the value of a smart meter at node $v_i \in V$ of G, which stands for electricity stored in storage systems at a node v_i at a certain point in time. Then the objective function for this problem is defined as follows:

$$\text{Minimize } \max_{v_i \in V} \{C(v_i)\} \tag{2}$$

In this paper, we do not consider characteristics of power systems but focus on control for information exchange on power grids. Therefore, a total cost of all the nodes does not change after procedures are conducted. Thereby, this min-max problem is equivalent to max-min problem. This objective function can be solved by a minimization problem on variance among node costs. Variance δ^2 is defined as follows:

$$\delta^2 = \frac{\sum_{i=1}^{n} (C(v_i) - \overline{C})^2}{n}, \left(\overline{C} = \frac{\sum_{i=1}^{n} C(v_i)}{n} \right) \tag{3}$$

To minimize the variance of node costs, the objective function δ^2 should be solved in each tie-set. Since an operating station is difficult to recognize all information of G and to follow up dynamic changes occurring around a gird, a divide-and-conquer method is appropriate to solve this problem by using divided units $L_B = \{L_1, L_2, \ldots, L_\mu\}$. A leader of a tie-set L_i can grasp costs of all nodes belonging to L_i. Then the equation 2 should be solved independently in a tie-set L_i. 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. When the flag of adjacent tie-sets are set to 0, a tie-set waits 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.

```
Initialization:
  1) For all nodes in $V$ of given $G = (V, E)$, set node cost $C(v_i)$ from 0 to 100 at random
  2) Call RandomExecute, or ConditionalExecute
RandomExecute:
  1) Until $\delta^2 \to 0$, select $L_i (\in \boldsymbol{L}_B)$ at random
  2) If $flag$ of $L_i$ is 1, then call DistributeElectricity
ConditionalExecute:
  1) Until $\delta^2 \to 0$, select $L_i (\in \boldsymbol{L}_B)$ where $\delta^2(L_i)$ is max
  2) If $flag$ of $L_i$ is 1, then call DistributeElectricity
DistributeElectricity:
  1) Set 0 to $flag$ of $L_i$ and all adjacent tie-sets in $\boldsymbol{L}_a$
  2) Minimize $\max_{v_k^i \in V_i} \{C(v_k^i)\}$
  3) Set 1 to $flag$ of $L_i$ and all adjacent tie-sets in $\boldsymbol{L}_a$
```

Fig. 6. Distributed algorithm for minimizing max $\{C(v_k^i)\}$

C. Distributed Control among Tie-sets

The distributed algorithm is described in Fig.6. Let V_i be a set of vertices involved in a loop denoted by L_i. In initialization, all nodes have a random node cost ranging from 0 to 100. A leader node of each tie-set recognizes all the node costs in its unit. Next, **RandomExecute** or **ConditionalExecute** program selects one tie-set L_i. If a tie-set L_i and adjacent tie-sets $\boldsymbol{L}_a$ are not in process, L_i conducts electricity distribution in its local units. **RandomExecute** program selects a tie-set at random, while **ConditionalExecute** program selects a tie-set based on certain condition. The condition is $\delta^2(L_i)$ that stands for a calculated value of the equation 3 in a tie-set L_i. A tie-set that has the largest $\delta^2(L_i)$ is selected as a tie-set in which the distributed algorithm is conducted. In a selected tie-set L_i, the equation 2 is resolved. An example of electricity distribution is shown in Fig.7

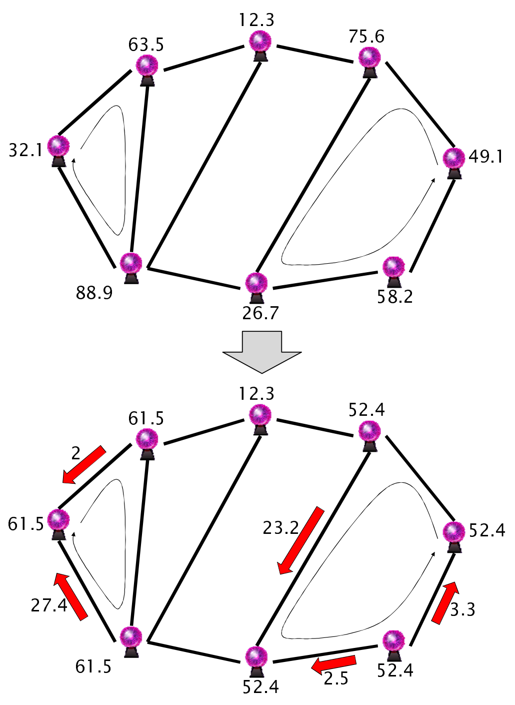


Fig. 7. Electricity distribution based on tie-sets

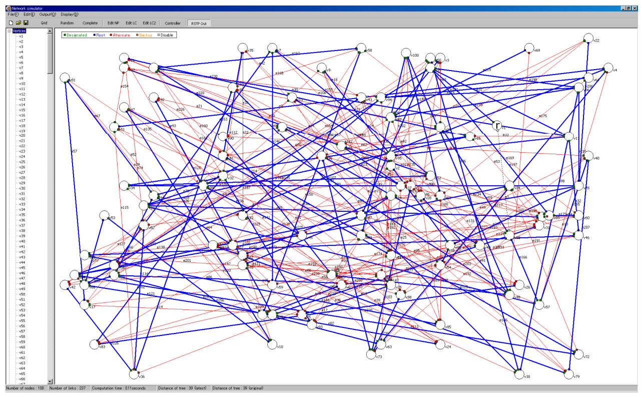


Fig. 8. Network configuration consisting of 100 nodes

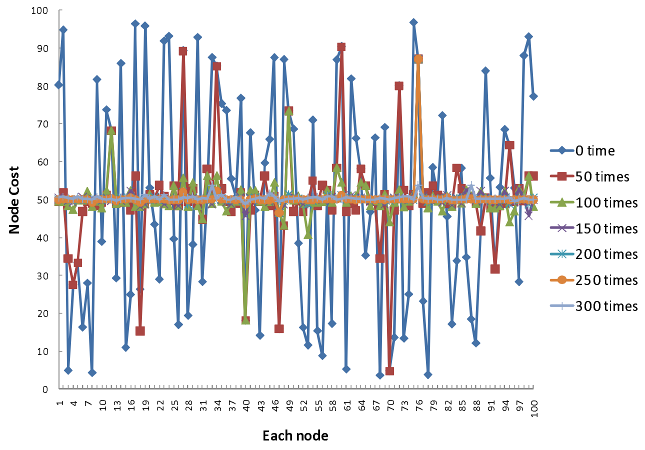


Fig. 9. The change of node costs after 300 times of computations (100 nodes)

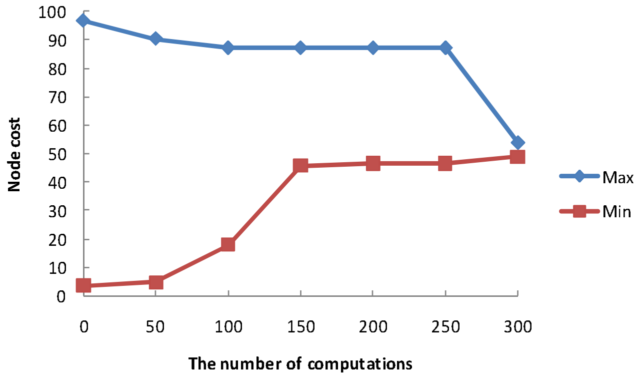


Fig. 10. The change of the maximum node cost and the minimum node cost

D. Simulation and Experiments

A simulator is made to realize distributed control for electricity distribution suggested in this paper, and to conduct experiments. In configuring a network, links are set to be undirected through which data can flow bi-directionally. In addition, network is designed to be redundant, in other words, bi-connected to be able to cope with failure as shown in Fig.8. As node configuration, each node has input ports and output ports, a message buffer, and a processor. Common buffering method is taken 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.

1) Experiment on 100 nodes: $G = (V, E)$ is a bi-connected graph and created at random with 100 nodes. We first conducted experiments by **RandomExecute** in the distributed algorithm in Fig.6 300 times. **RandomExecute** picks up one tie-set at random and conduct distributed control in the tie-set to solve equation 3.

Fig.9 shows the process of convergence to the average of all the node costs by conducting the distributed algorithm shown in Fig. 6. The node costs are plotted when the number of computations reaches 0, 50, 100, 150, 200, 250, 300 times. All the node costs gradually converge at mean value.

Fig.10 shows the process of the change of maximum node cost and minimum node cost. Around 300 times of total computing, the maximum value and minimum value become almost the same. This result shows that local optimization based on tie-set units leads to global optimization. This is the result by employing **RandomExecute** that picks up one tie-set at random. It is conceivable that picking up one tie-set at random makes the total number of computations unnecessarily increased. Then we conducted **ConditionalExecute** and compared with the results of **RandomExecute**.

2) **RandomExecute** *and* **ConditionalExecute***:* Next, we conducted experiments to count the number of computations by **RandomExecute** and **ConditionalExecute**. **ConditionalExecute** chooses one tie-set L_i in which $\delta^2(L_i)$ is max.

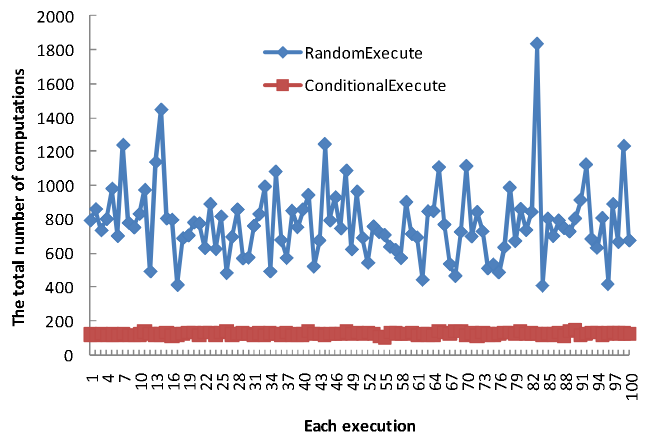


Fig. 11. The total number of computations by **RandomExecute** and **ConditionalExecute** per each execution

TABLE I
COMPARISON BETWEEN **RandomExecute** AND **ConditionalExecute**

RandomExecute

	Total Computations	Max Tie-set	Mini Tie-set
Average	778.19	12.98	0.76
Worst	1834	24	3
Best	409	7	0

ConditionalExecute

	Total Computations	Max Tie-set	Mini Tie-set
Average	125.4	4.29	0.02
Worst	147	6	2
Best	103	3	0

In this experiment, the network configuration is the same as the above experiment. From here, δ^2 of the equation 3 is set to 0.1. Then we executed **RandomExecute** and **ConditionalExecute**, respectively. For each execution, the total number of computations that satisfies $\delta^2 \to 0.1$ is counted. The above experiment is conducted 100 times.

Fig.11 shows the total number of computations per each execution. As shown Fig.11, **RandomExecute** requires unnecessary computations while **ConditionalExecute** solves the objective function much more efficiently.

Table I shows numerical results. Max Tie-set stands for the total number of computations in the tie-set which conducts most computations. Min Tie-set stands for the total number of computations in the tie-set which conducts least computations For example, as shown in Table I, **RandomExecute** conducts 1834 times of calculations in the worst case, 409 times at best, whereas **ConditionalExecute** conducts 147 times of calculations at worst. Generally, **ConditionalExecute** is much better than **RandomExecute**, especially in terms of worst values.

3) Experiments from 20 to 100 nodes: To examine the behavior for increase in the number of nodes, we conducted experiments from 20 to 100 nodes. Each G is created at random and δ^2 of the equation 3 is set to 0.1. For each network configuration, the total number of computations that satisfies $\delta^2 \to 0.1$ is counted. The experiment above is conducted 100 times for each network configuration. We employed **ConditionalExecute** and executed the algorithm for each network configuration.

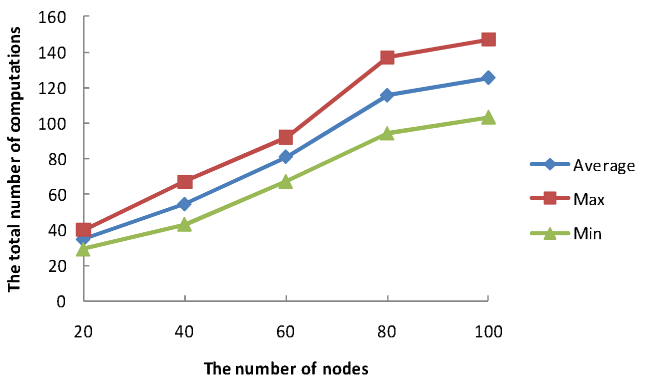


Fig. 12. The total number of computations in a network consisting of 20 to 100 nodes

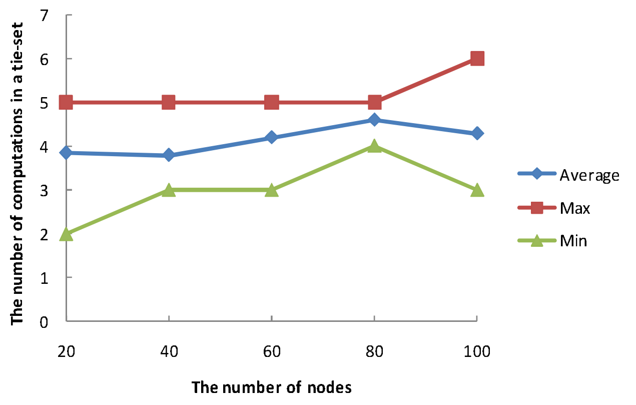


Fig. 13. The number of computations of a tie-set conducting the largest number of computations (20 to 100 nodes)

Fig.12 shows the total number of computations in a network consisting of 20 to 100 nodes. Max stands for the worst value, and Mini stands for the best value in 100 times of experiments for each network configuration. As shown Fig.12, the total number of computations and the number of nodes are in proportional relationship.

Fig.13 shows the number of computations in a tie-set that conducts the largest number of computations among all tie-sets. Unlike the result of Fig.12, Fig.13 shows the number of computations conducted in one tie-set, not the total number of computations of all tie-sets. Max stands for the worst value, and Mini stands for the best value in 100 times of experiments for each network configuration. While the total number of computations is proportionate to the number of nodes, the number of computations in individual tie-set increases very little. This indicates that distributed control based on tie-sets is suitable to large-scale networks.

4) Experiments under lax conditions: Experiments above are the resolution of optimization problems. However, there

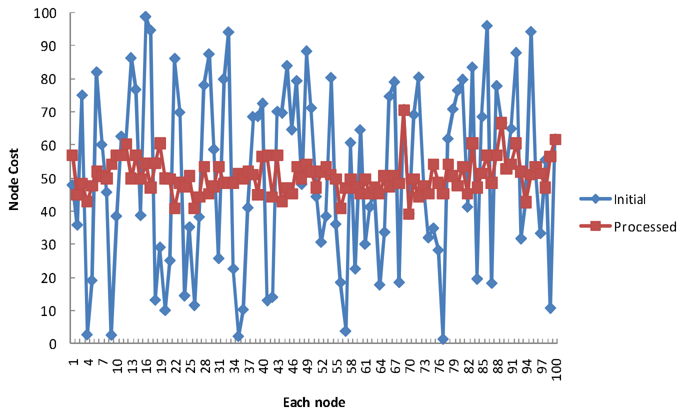


Fig. 14. The node costs before and after the procedure that makes all node costs above 30

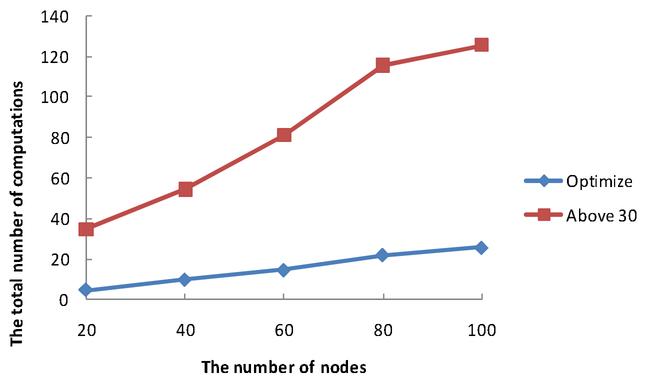


Fig. 15. The comparison of the total number of computations between optimization and semi-optimization (Above 30)

are many cases where electricity over certain point is enough. In other words, a node should only have electricity more than a standard point, and does not need excessive power. For example, in the range from 0 to 100, each node must have at least 30 and does not require more power. We solve this kind of problems as semi-optimization problems. Here, we set 30 to the least point of a node cost, and apply our distributed algorithm seen in Fig.6 to this problem. In this experiment, a network is created at random with 100 nodes.

The result is shown in Fig.14. The distributed control by tie-set units stops their procedures when all node costs become more than 30. All the node costs generally converge on the average point, albeit they are not optimal.

Fig.15 shows the total number of computations by optimization and semi-optimization from 20 to 100 nodes. As shown in Fig.15, the total number of computations of the programs under the lax conditions is one-sixth of that of the programs of optimization. Therefore, computations under loose conditions decrease procedure load. According to the result, determining a least point to store electricity is important to manage the respective resources allocated to each node.

V. Conclusion and Future Work

In this paper, tie-set graph theory is first introduced as a theoretical approach. Then distributed control for an electricity distribution problem focusing on smart meters is discussed. The electricity distribution problem is theoretically modeled as a minimization problem on variance. The result of simulation shows 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 due to 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. That is because the structure of tie-sets is uniquely determined by the graph theoretical concept of a tie-set basis implicitly underlies a network.

As a future work, we will conduct more extensive simulations focusing on scalability, network topology, and properties of power grids. This paper does not consider the properties of power systems such as transmission loss, conversion loss, etc. Therefore, we need to add those factors to simulation conditions and conduct more pragmatic experiments.

References

[1] Patrick Mazza *A Northwest Initiative for Job Creation, Energy Security and Clean, Affordable Electricity*, Powering Up the Smart Grid, April 27, 2005

[2] Rob van Gerwen, Saskia Jaarsma and Rob Wilhite, KEMA, The Netherlands *Smart Metering*, www.leonardo-energy.org, Distributed Generation, July 2006

[3] David E. Bakken, Carl H. Hauser, Harald Gjermundrod, Anjan Bose *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

[4] *Understanding SONET UPSRs*, http://www.sonet.com/EDU/upsr.htm, Available online

[5] *Understanding SONET BLSRs*, http://www.sonet.com/EDU/blsr.htm, Available online

[6] S.Shah, M.Yip, *Extreme Networks' Ethernet Automatic Protection Switching (EAPS) Version 1*, Network Working Group, Request for Comments: 3619, October 2003.

[7] N.Okada, etc., *A method to determine the distributed control setting of looping devices for active distribution systems*, IEEE PowerTech 2009, No.331, June 2009.

[8] Wasem OJ., *An algorithm for designing rings for survivable fiber networks*, IEEE Transactions on Reliability 1991; 40:428-439.

[9] L. M. Gardner, M. Heydari, J. Shah, I. H. Sudborough, I. G. Tolis, and C. Xia, *Techniques for finding ring covers in survivable networks*, in Proc. IEEE GLOBECOM, San Francisco, CA, Nov. 1994, pp.1862-1866.

[10] Thomassen C., *On the complexity of finding a minimum cycle cover of a graph*, SIAM Journal on Computing 1997; 26(3):675-677.

[11] N.Shinomiya, etc., *A theory of tie-set graph and its application to information network management*, International Journal of Circuit Theory and Applications 2001; 29:367-379

[12] Jacek Malinowski, *A new efficient algorithm for generating all minimal tie-sets connecting selected nodes in a mesh-structured network*, IEEE Transactions on reliability, Vol.59, No.1, March 2010

[13] T.Koide, H.Kubo, and H.Watanabe, *A study on the tie-set graph theory and network flow optimization problems*, International Journal of Circuit Theory and Applications 2004; 32:447-470

[14] Iri M, Shirakawa I, Kajitani Y, Shinoda S etc., *Graph Theory with Exercises*, CORONA Pub: Japan, 1983.

[15] N.Shinomiya, T.Koide, M.Kuroha, H.Watanabe, *Tie-Set Graph, Meta-Tie-Set Graph and Information Network*, IEEK/IEICE ITC-CSCC '99

[16] Pacific Crest Mosaic, *Mesh Networks Is Communications Winner in Utility Survey*, Pacific Crest Mosaic Smart Grid Survey - August 2009