Complete Automation of Future Grid for Optimal Real-Time Distribution of Renewables

Kiyoshi Nakayama, *Student Member, IEEE*, Kyle E. Benson, Lubomir F. Bic, Michael B. Dillencourt
Dept. of Computer Science, University of California, Irvine, CA 92697-3435, USA
Email: kiyoshi.nakayama@ieee.org, {kebenson, bic, dillenco}@ics.uci.edu

***Abstract*—In this paper, a novel distributed control technique, which integrates tie-set graph theory with an intelligent agent system, is presented to distribute renewable energy resources to consumers in a future large-scale power grid connecting with huge amounts of real-time end-use devices on its demand side automatically and perfectly. Using the proposed technique based upon tie-set graph theory, the power grid can be effectively divided into a set of loops; each loop can be seen as a tie-set representing a set of all edges comprising a loop in a graph. By integrating the tie-set graph theory with autonomous agents that constantly navigate on a grid, complete automation of a future power grid becomes feasible with dynamic synchronization of state information among tie-sets. The supply and load of electric power at every instant can be balanced even if the future load is uncertain and renewable generation is highly variable and unpredictable. Simulation results on a one hundred-node network demonstrate the optimal real-time distribution of renewables and thus the effectiveness of the proposed method.**

I. INTRODUCTION

Smart grid technologies and services are being developed from a variety of perspectives. While the innovative smart grid infrastructure is expected to increase the efficiency and reliability of grid operations, the increased complexity of the new system has to be addressed. Obtaining huge volumes of data from millions of advanced smart meters itself does not indicate that a grid becomes intelligent. Unlike traditional power grid systems, the future grid will invariably feature rapid integration of alternative forms of energy generation; thus, electricity flows bi-directionally in a complicated manner [1]-[3]. The renewable energy resource applications will offset dependence of fossil fuels, and provide green power options for atmospheric emissions curtailment. However, harvesting renewable energies may be unstable due to weather conditions. This will require new optimization methods for energy resources that are inherently distributed with interconnection standards and operational constraints. To maximize the efficiency of distributed energy resources as well as realize a stable operation for renewables, balanced allocation of dispersed energy resources should constantly take place within autonomous control systems especially due to the following reasons [4]:

- Realizing a reliable and sustainable future grid where generation of renewable energy resources are unstable depending on weather conditions;
- Preventing blackout caused by imbalanced distribution of dispersed energies;
- Cutting the cost of futile power generation and CO_2 from fossil fuels and nuclear powers by distributing renewable energies in a balanced manner.

More recently, attention has focused on mesh topologies partly because of the increased flexibility, efficiency, and resiliency they provide in power distributions, and partly because the natural evolution of power grid topologies leads to a mesh-type connection. Since the birth of graph and circuit theory traced back to Kirchhoff's voltage and current lows (KVL and KCL) in 1845, use of loops in circuits and power systems has demonstrated the effectiveness in controlling power flows and designing fault-tolerant reliable systems even in a complicated non-planar topology of a graph. Those works are listed in [5].

A system operating point (SOP) has the task of integrating growing amounts of renewable power into the power grid. The SOP makes a sequence of decisions to balance the supply and load of electric power at every instant, in the face of several unknowns: future load is uncertain; renewable generation is highly variable and unpredictable. As forecasts of these variables have errors, a real-time approach to get more accurate forecasts is proposed on the basis of observations accumulate when supply and load are balanced [6]. Various sophisticated techniques in residential and power networks to realize optimal demand response are also proposed [7], [8].

Our goal is to make prospective power systems completely autonomous and optimized on the basis of tie-set graph theory and autonomous agents [9] in a future large-scale power grid integrating millions of real-time technologies in its residential area that are connected to such SOPs. In previous works, the principal partition theorem focusing on loops of an information network was proposed for distributed network management [10]. The paper [11] presents an efficient method of enumerating all minimal tie-sets connecting selected nodes in a mesh topological network. Tie-set-based approach to network fault tolerance is also proved to be more effective than traditional tree-based centralized approaches [12]. In [4], [13], a novel technique based on tie-sets is proposed to realize global optimization of balanced distribution of dispersed static energies with local optimizations whose property is theoretically verified in [14].

This paper proposes an autonomous control method for optimal real-time distribution of renewable energy resources (ORDER) that is necessary to realize a future sustainable power grid. We do not currently consider the characteristics of prospective power systems such as generation costs, transmission loss, and conversion loss to model the ORDER problem on the basis of graph theory. Those problems will be dealt with in our future works.

II. Distributed Algorithms based on Tie-Set Graph Theory

The concept of tie-sets has been applied to core fields of circuits and systems in formulating, solving, and characterizing properties of various problems from reliability and optimization perspectives. As all the distributed algorithms in this section are proposed in [4] in detail, we introduce the concept of the algorithms below.

A. Fundamental System of 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 $L_i = \{e^i_1, e^i_2, ...\}$ be a set of all the edges that constitutes a loop in G. The set of edges L_i is called a *tie-set* [5]. Let T and $\overline{T}$ be a spanning tree and a cotree of G, respectively, where $\overline{T} = E - T$. As T on a graph $G = (V, E)$ is a spanning tree, T 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 an 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 a *fundamental circuit* that consists of the path P_T and the edge l is uniquely determined as follows:

$$C(l) \quad = \quad (a, l = (a, b), P_T(b, a)) \qquad (1)$$

The tie-set $L(l)$ corresponding to $C(l)$ is denoted as a *fundamental tie-set*. It is known that $\mu = \mu(G) = |\overline{T}|$ fundamental tie-sets exist in G, and they are called a *fundamental system of tie-sets*. A fundamental system of tie-sets $\mathbb{L}_B = \{L_1, L_2, ...L_\mu\}$ covers all the vertices and edges in G as shown in Fig. 1. Let V_i be a set of all the vertices contained in a fundamental circuit $C(l_i)$. Although a tie-set L_i is defined as a set of all the "edges" of a loop, we sometimes refer to a tie-set as a loop that is decided by (V_i, L_i).

Fundamental tie-sets are independent of each other; any fundamental tie-set cannot be obtained by calculus $\oplus$[1] among other tie-sets. As $l \in \overline{T}$ is only included in a fundamental tie-set $L(l)$ of a fundamental system of tie-sets $\mathbb{L}_B$, a tie-set that includes l cannot be created if the calculus $\oplus$ is applied to other fundamental tie-sets than $L(l)$.

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.

In this paper, $\underline{e}(L_i, L_j)$ is determined by the set of common vertices of L_i and L_j. Each fundamental tie-set of a given graph G is uniquely mapped to the specific tie-set graph $\underline{G}$ as shown in Fig. 2.

[1]The definition of $\oplus$ for a set A and a set B is defined as follows: $A \oplus B = (A - B) \cup (B - A) = (A \cup B) - (A \cap B)$.

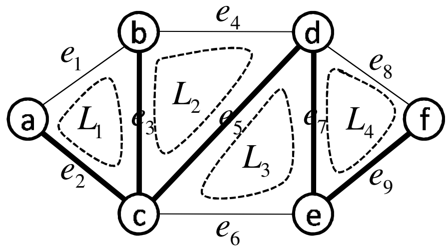

Fig. 1. An example of a fundamental system of tie-sets.

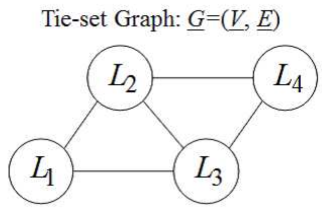

Fig. 2. Tie-set graph of Fig. 1.

C. State Information of a Node

Each node v_i mainly has three types of information as state information [4]:

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 its information.

The node c in Fig. 1, for example, has state information of $\{e_2, e_3, e_5, e_6\}$ as incident links, $\{a, b, d, e\}$ as adjacent nodes, and $\{L_1, L_2, L_3\}$ as Tie-set Information.

D. Distributed Algorithm for Configuring Tie-Set Information

As described in II-C, each node has state information of fundamental tie-sets to which the node belongs to solve problems within some related loops. Each node executes a distribute algorithm for configuring Tie-set Information to recognize fundamental tie-sets. Although the mechanism is proposed in [4], this distributed algorithm becomes more efficient if only a node a or b of each $l = (a, b) \in \overline{T}$ sends a *Find Tie-set* Message (FTM) on l and T, and stores the information of FTM that comes back to the node. In that case, the procedure completes its procedures with Communication Complexity of $O(\mu|V|)$ and Time Complexity of $O(D)$ where D is a diameter of a graph.

E. Leader Election Problem in a Tie-Set

It is imperative to develop a Leader Election Algorithm in a Tie-set (LEAT) that select a leader node to solve a dilemma that occurs when using an overlapping resource among other tie-sets. If two control units come up with two different operating points, the leaders of the two units negotiate with each other to gain priority of control. In [4], a leader of a tie-set is determined by exchanging *Decide Leader* Messages (DLMs) with magnitude relation of node ID (physical addresses), the number of adjacent nodes (incident links), the number of tie-sets to which a node belongs, among others within a tie-set.

The leader node constantly sends a token message around its tie-set, and collects state information of the other nodes within L_i. If the leader fails, every node within a tie-set can be a new leader node by just conducting LEAT again.

F. 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 (CAT) [4] are frequently required. In order to realize CAT, each leader of a tie-set must recognize connection information with adjacent tie-sets. In addition to state information described in II-C, a leader node v_l^i of a tie-set L_i has the additional information below:

- Adjacent Tie-sets $\mathbb{L}_i^a = \{L_1^i, L_2^i, ...\}$
  An adjacent tie-set L_j of L_i is determined according to the relation of connection $\underline{e}(L_i, L_j) \in \underline{E}$ of $\underline{G}$.

For example, adjacent tie-sets of L_1 in Fig. 1 are $\mathbb{L}_1^a = \{L_2, L_3\}$ so that L_1 constantly communicates with $\mathbb{L}_1^a$.

A routing table for communications among tie-sets should be computed beforehand so that each leader node can quickly communicate with other leaders using an appropriate path to them.

III. Optimal Real-Time Distribution of Renewable Energy Resources

In this section, we define an Optimal Real-time Distribution problem of renewable Energy Resources (ORDER problem) and provide a completely autonomous distributed control method to solve the ORDER problem.

A. Problem Formulation

Here, let $P_v(t)$ be the amount of power (algebraic sum of power production, consumption, inflow, and outflow monitored by smart meters) that each node $v \in V$ possesses at time t indexed by $t \in \mathcal{T} := \{1, 2, ...\}$. Each node has an initial power $P_v(0)$. Let $L_v(t)$ be a load of a node v at t. Each node also has distributed generation (DG) functions that exploit renewable energy resources. The amount of power derived from renewables is defined as $R_v(t)$ at t. $L_v(t) - R_v(t)$ is often referred to as *net demand*. The load L and renewable power R are both random processes. In each tie-set, let us assume that there is a large Centralized Generation Facility (CGF) that utilizes resources except renewables, such as fossil fuel (coal, gas powered), nuclear, etc. Therefore, at time t, the amount of power produced by CGF assigned to each tie-set is defined as $D_v(t)$. Then, the amount of power $P_v(t)$ at a node v at time t is defined as follows:

$$P_v(t) = P_v(t-1) - L_v(t) + R_v(t) + D_v(t) \tag{2}$$

For any two vertices v, u ($v \neq u$), let $f(v, u, t)$ be a power flow over an edge $e(v, u)$ from vertex v to u at time t and defined as an edge flow, where if $f(v, u, t)$ flows along the direction of an edge $e(v, u)$ then $f(v, u, t) > 0$; otherwise $f(v, u, t) < 0$. If the value of an edge flow $f(v, u, t)$ over $e(v, u)$ at time t is decided, then $P_v(t)$ and $P_u(t)$ are respectively updated as $P'_v(t)$ and $P'_u(t)$ according to the following rule:

$$f(v, u, t) = F \Longrightarrow \begin{cases} P'_v(t) = P_v(t) - F \\ P'_u(t) = P_u(t) + F \end{cases} \tag{3}$$

Therefore, if an edge flow $f(v, u, t)$ is decided, $P_v(t)$ and $P_u(t)$ are automatically changed according to the equation 3, which physically stands for power outflow from a node v and inflow to a node u. Under those conditions, an optimal real-time distribution problem of renewable energy resources (ORDER problem) is defined as follows:

ORDER Problem: For a given time sequence $\mathcal{T} = \{t\}$, each $v \in V$, and each $e(v, u) \in E$, decide the amount of power by CGF $D_v(t)$ and the edge flow $f(v, u, t)$ that satisfies the following conditions with certain threshold P_{min}.

$$\text{Minimize } \max_{v \in V} \{P_v(t)\} \tag{4}$$

$$\text{Minimize } \sum_{t \in \mathcal{T}} \sum_{v \in V} D_v(t) \tag{5}$$

$$P_v(t) \geq P_{min} \tag{6}$$

Equation 4 is equivalent to minimization of the variance of all the powers of nodes by adjusting each edge flow $f(v, u, t)$ and $D_v(t)$, which intrinsically indicates balanced distribution of renewables. The variance at t is defined as follows:

$$\sigma^2(t) = \frac{\sum_{v \in V} (P_v(t) - \overline{P})^2}{|V|}, \left(\overline{P} = \frac{\sum_{v \in V} P_v(t)}{|V|}\right) \tag{7}$$

To solve the min-max problem 4, the variance should be 0 as follows [4]:

$$\sigma^2(t) \to 0 \tag{8}$$

Equation 5 indicates that $D_v(t)$ must be generated as little as possible because the generations by fossil fuels and nuclear plants have an impact on the environment and society. At the optimal $D_v(t)$ and $f(v, u, t)$, each value of $P_v(t)$ must be equal to or more than the threshold P_{min} (Equation 6).

Ultimately, each $P_v(t)$ converges on P_{min} with equations 4 - 6. Thus, $D_v(t)$ is a net demand as $D_v(t)$ becomes $L_v(t) - R_v(t)$ at the end.

In real power systems, $D_v(t)$ cannot be provided instantly as it takes a while to produce electric power from CGF. Therefore, we need to consider such a time lag in real operations.

B. Autonomous Distributed Control for ORDER Problem

In order to solve the ORDER problem, we propose an autonomous distributed control method by a set of μ autonomous tie-set agents.

1) Tie-set Agent (TA): The leader node of a tie-set L_i, which is considered to be a System Operating Point (SOP) in a tie-set, receives a measurement vector $y(t)$ at certain time t that provides the value of $P_v(t)$ where $v \in V_i$ in L_i. The measurements may also include current load $L_v(t)$, renewable power generation $P_v(t)$, weather forecasts, pricing info, among others. The measurement data of $y(t)$ is obtained using a Tie-set Agent (TA), which is an autonomous agent that constantly navigates around on a tie-set and brings the data to a leader node with certain time interval (TI).

2) Tie-set Evaluation Function (TEF): Obviously, many tie-sets share some overlapping resources with adjacent tie-sets. To avoid contradictions of state information within or among tie-sets, each tie-set communicates with adjacent tie-sets to decide which tie-set processes their overlapping resources first. When deciding the process priority for resources shared by several tie-sets, each tie-set exchanges the value of Tie-set Evaluation Function (TEF) denoted as $\Phi(L_i)$ with its adjacent tie-sets. TEF $\Phi(L_i)$ is calculated based upon $y(t)$ and subject to objective functions, constraints, characteristics of a network, etc. In this research, we consider three types of TEF $\Phi(L_i)$ as follows:

- $\Phi_1(L_i) = \text{Random}$
  The value of $\Phi(L_i)$ is randomly assigned.
- $\Phi_2(L_i) = P_v(t)^i_{max} - P_v(t)^i_{min}$
  This TEF is the delta between $P_v(t)^i_{max}$ and $P_v(t)^i_{min}$ within a tie-set L_i, where $P_v(t)^i_{max} = \max_{v \in V_i}\{P_v(t)\}$, $P_v(t)^i_{min} = \min_{v \in V_i}\{P_v(t)\}$.
- $\Phi_3(L_i) = \sigma^2(t)$
  This TEF is the variance of $P_v(t)$ as defined in the equation 7, where $v \in V_i$.

$\Phi_1(L_i)$ (= Random) is only used when the process priority cannot be determined by other TEFs.

3) Tie-set Flag (TF): Tie-set Flags (TFs) are used to distinguish tie-sets that are in process from tie-sets that are stand-by once the process priority is decided using TEFs. The definition of Tie-set Flag (TF) denoted as $\zeta(L_i)$ is as follows:

- $\zeta(L_i) = 0$: L_i is Stand-By
- $\zeta(L_i) = 1$: L_i is In Process

If $\zeta(L_i) = 1$, each adjacent tie-set $L_j \in \mathbb{L}^a_i$ defined by a relation of $\underline{e}(L_i, L_j)$ should stand by making its TF as $\zeta(L_i) = 0$. After L_i finishes its procedure, L_i sends a signal to each adjacent tie-set in order to notify the completion of its procedure as well as making its TF as $\zeta(L_i) = 0$. Then, each tie-set L_i calculates its value of TEF $\Phi(L_i)$ based on $y(t)$ that is periodically sent with its Tie-set Agent (TA), and compares its value of $\Phi(L_i)$ with those of adjacent tie-sets. If TFs are used in the procedure, parallel optimizations among tie-sets are feasible.

4) Autonomous Control with TEF Message (TEFM): Let a *Tie-set Evaluation Function* Message (TEFM) be a message that is used when exchanging the value of TEF $\Phi(L_i)$. Then, complete autonomous control is realized by constantly repeating the following procedures with TEFM in each tie-set.

If $\zeta(L_i) = 0$, each tie-set L_i sends TEFM to each adjacent tie-set L_j. The tie-set L_j that receives the TEFM writes its value of TEF $\Phi(L_j)$ in the message, and then sends it back to L_i. After L_i receives the TEFMs from all the adjacent tie-sets $\mathbb{L}^a_i = \{L^i_1, L^i_2, ...\}$, L_i compares its value of $\Phi(L_i)$ with those of adjacent tie-sets.

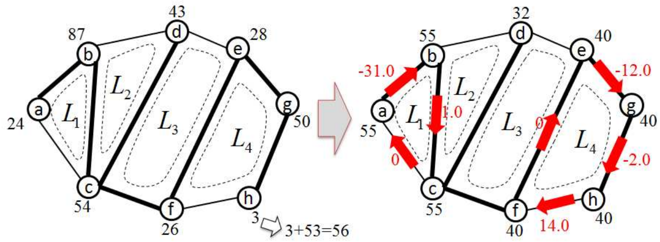

Fig. 3. Parallel optimizations in tie-sets L_1 and L_4.

Only if the value of $\Phi(L_i)$ is the largest among those of all the adjacent tie-sets, L_i conducts a distributed optimization algorithm described in Algorithm 1. If the value of $\Phi(L_i)$ is the same as $\Phi(L_j)$, L_i uses another TEF $\Phi(L_i)$ to decide the procedure priority.

Otherwise, L_i waits until it receives a signal from the tie-set that occupied the shared resources and finished its procedures. Then, L_i checks the value of $\Phi(L_i)$ on the basis of a measurement vector $y(t)$ of L_i, and sends TEFMs again to all the adjacent tie-sets to compare its TEF value with those of adjacent tie-sets.

For instance, if $\Phi_2(L_i)$ $(= P_v(t)^i_{max} - P_v(t)^i_{min})$ is used as TEF, the TEF value of each tie-set in Fig. 3 is $\Phi_2(L_1) = 63, \Phi_2(L_2) = 44, \Phi_2(L_3) = 28, \Phi_2(L_4) = 47$, respectively. Therefore, the TFs of $\{\zeta(L_1), \zeta(L_2), \zeta(L_3), \zeta(L_4)\}$ are $\{1, 0, 0, 1\}$ so that L_1 and L_4 are in process in Fig. 3.

5) Distributed Optimization Algorithm in Tie-set (DOAT): Distributed Optimization Algorithm in a Tie-set (DOAT) is described in Algorithm 1. On the basis of the DOAT, the optimal flows at time t in L_1 in Fig. 3 are $f(a, b, t) = -31.0$, $f(b, c, t) = 1.0$, and $f(c, a, t) = 0$ considering directions of those flows. Similarly, the optimal flows at time t in L_4 in Fig. 3 are $f(e, g, t) = -12.0$, $f(g, h, t) = -2.0$, $f(h, f, t) = 14.0$, and $f(f, e, t) = 0$. DOAT finishes with Time Complexity of $O(D)$ where D is a diameter of a graph.

IV. Simulation and Experiments

A simulator is made to realize distributed algorithms to solve the ORDER problem described in section III-A and to conduct experiments. As to an experimental environment, we used Microsoft Windows 7 as an operating system and Java as a development language. In configuring a power network, links are set to be undirected through which power flows can pass bi-directionally. In addition, the power network is designed to be redundant, or bi-connected from a graph theoretical standpoint. As for node configuration, each node has 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

Algorithm 1 Distributed optimization algorithm in a tie-set

Step0:
  Calculate the average value $\overline{P}$ of $P_v(t)$, where $v \in V_i$.
  if $\overline{P} < P_{min}$ **then**
    For each $v \in V_i$, $D_v(t) = P_{min} - \overline{P}$.
    Select one node $v_d \in V_i$, and then $P'_{v_d}(t) = P_{v_d}(t) + \sum_{v \in V_i} D_v(t)$.
    Recalculate the average value $\overline{P}$.
  end if
Step1:
  for each node $v \in V_i$ **do**
    if $P_v(t) \geq \overline{P}$ **then**
      Classify $v \in V_i$ into a set of nodes X.
    else if $P_v(t) < \overline{P}$ **then**
      Classify $v \in V_i$ into a set of nodes Y.
    end if
  end for
Step2:
  while $X \cup Y \neq \emptyset$ **do**
    Select a node x from X, and a node y from Y arbitrarily.
    if $P_x(t) - \overline{P} > \overline{P} - P_y(t)$ **then**
      $f_{P(x,y)}(t) = \overline{P} - P_y(t)$.
      Update $P_y(t)$ to $\overline{P}$, and $P_x(t)$ to $P_x(t) + P_y(t) - \overline{P}$.
      Remove y from Y.
    else if $P_x(t) - \overline{P} < \overline{P} - P_y(t)$ **then**
      $f_{P(x,y)}(t) = P_x(t) - \overline{P}$.
      Update $P_x(t)$ to $\overline{P}$, and $P_y(t)$ to $P_y(t) + P_x(t) - \overline{P}$.
      Remove x from X.
    else $\{P_x(t) - \overline{P} = \overline{P} - P_y(t)\}$
      $f_{P(x,y)}(t) = P_x(t) - \overline{P}$.
      Update $P_x(t)$ to $\overline{P}$, and $P_y(t)$ to $P_y(t) + P_x(t) - \overline{P}$.
      Remove x from X, and y from Y.
    end if
  end while
Step3:
  Calculate the value of each edge flow $f(v, u, t)$ over $e(v, u) \in L_i$ considering all flows decided above, and distribute those flows to the tie-set L_i.

in the processor, the message is sent to other nodes through appropriate output ports unless it is received or discarded.

A simulation network $G = (V, E)$ is created with 100 nodes and 190 links with random connection. The number of tie-sets is 91 and the height of tree is 6. Initial power $P_v(0)$ of each node is randomly assigned between 50 and 100 as a real number. P_{min} is set as 40. Here, $L_v(t)$ and $R_v(t)$ have the following conditions;

$$0 \leq L_v(t) \leq l, 0 \leq R_v(t) \leq r \tag{9}$$

In this experiment, $l = 20, r = 10$, where the generation of renewables is intended to be 50% for a demand at each node. Therefore, for a given time sequence with the time interval of 5 ms ($\mathcal{T} = \{0, 5, 10, ...\}$), $L_v(t)$ and $R_v(t)$ are given between 0 to 20 and 0 to 10 at random as real numbers, respectively.

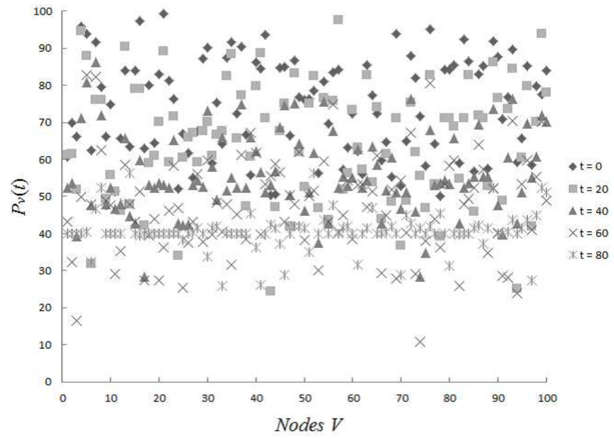

Fig. 4. Behavior of overall convergence in a 100-node grid.

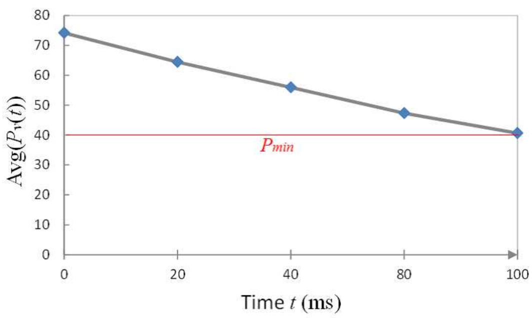

Fig. 5. Convergence of the average node power on P_{min} based on time.

Each tie-set exchanges $\Phi_2(L_i)$ $(= P_v(t)^i_{max} - P_v(t)^i_{min})$ as TEF, and every measurement vector $y(t)$ is sent to a leader node of each tie-set with the time interval of 1 ms ($\mathcal{T} = \{1, 2, 3, ...\}$) in this simulation.

A. Behavior of Overall Convergence

As mentioned, the ORDER problem can be solved with local optimizations in a system of μ independent tie-sets. By conducting the autonomous distributed control explained in section III-B, each $P_v(t)$ in a graph gradually converges around the threshold P_{min}.

Fig. 4 shows the behavior of overall convergence of all the node powers in a 100-node grid by executing the proposed algorithms where simulation data of $P_v(t)$ at each node $v \in V$ is taken at $t = 0, 20, 40, 60, 80$ (ms).

Fig. 5 is the another way of showing the convergence process of Fig. 4 toward the threshold P_{min}. $Avg(P_v(t))$ stands for the average value of all the node powers $P_v(t)$ as follows:

$$Avg(P_v(t)) = \frac{\sum_{v \in V} P_v(t)}{|V|} \tag{10}$$

As indicated in Fig. 5, each $P_v(t)$ gradually converges on P_{min} by iterative optimizations among tie-sets. When $t = 100$, the average value is $\overline{P} = 40.6645$; thereby all of the powers $P_v(100)$ almost converge on 40. However, the variance is $\sigma^2(100) = 25.94677$, which is not a small number. When

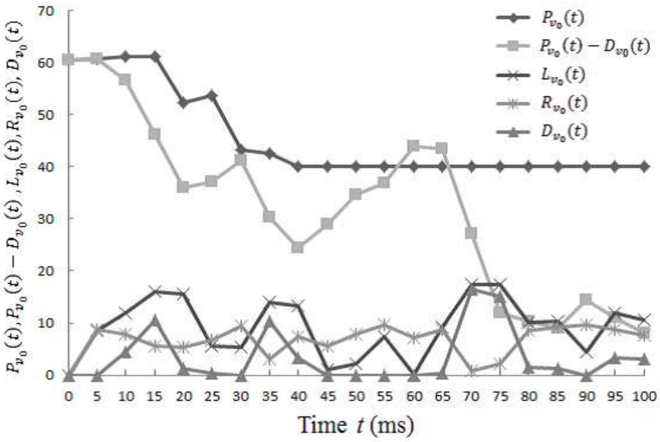

Fig. 6. Behavior of $P_{v_0}(t)$, $P_{v_0}(t)-D_{v_0}(t)$, $L_{v_0}(t)$, $R_{v_0}(t)$, and $D_{v_0}(t)$ from $t=0$ to 100.

$t = 1285$ ms, $\sigma^2(1285) = 0.00154$. This result demonstrates that most of the powers first quickly converge on a near-optimal solution within a short time, and then all of them become an almost optimal value as t increases.

B. Analysis of Behavior at Node

Next, we look at the behavior at a node focusing on v_0. The simulation conditions are the same as the ones stated above.

Fig. 6 shows the behavior of $P_{v_0}(t)$, $P_{v_0}(t)-D_{v_0}(t)$, $L_{v_0}(t)$, $R_{v_0}(t)$, and $D_{v_0}(t)$ from $t=0$ to 100. As the autonomous optimizations based on tie-sets are applied to $P_{v_0}(t)$ with modest generations of $D_{v_0}(t)$, the value of $P_{v_0}(t)$ keeps to be around 40 $(= P_{min})$ from $t \approx 40$. The total generation amount by CGF is $\sum_{0 \le t \le 100} D_{v_0}(t) = 71.2747$. This result substantiates the effectiveness of the proposed autonomous control method as local optimizations by tie-sets make the demand response at each node quite stable.

$P_{v_0}(t) - D_{v_0}(t)$ changes depending only on loads $L_{v_0}(t)$ and renewable powers $R_{v_0}(t)$ without any correction amount from the generation by CGF $D_{v_0}(t)$. As indicated in Fig. 6, $P_{v_0}(t) - D_{v_0}(t)$ decreases and becomes unstable with random assignments of $L_v(t)$ and $R_v(t)$ as time progresses. Therefore, if none of the optimization algorithms described in the previous section are applied, each node will deplete its power at some point during operations.

Although only the simulation result at a node v_0 is introduced here, the results of other nodes are similar with a minor difference of initial power $P_v(0)$ as it is assigned at random from 50 to 100.

V. Conclusion

In this paper, tie-set graph theory and distributed algorithms based on *tie-sets* are first introduced as a theoretical approach to an optimal real-time distribution problem of renewable energy resources (ORDER problem). Then, we formulate the ORDER problem and provide how to realize completely autonomous distributed controls with autonomous agents navigating around tie-sets. The simulation results show that local optimizations constantly iterated within tie-sets lead to global optimization. Although each tie-set has its limited local information, an entire power network is controlled in an orderly fashion due to the theoretical basis of a tie-set graph. Furthermore a series of local information about each node is consistent with the condition of an entire network because of the structure of tie-sets that is uniquely determined by the graph theoretical nature of a tie-set basis which implicitly underlies a network.

This work deals with abstract aspects of prospective power systems. Given the characteristics of power systems such as generation costs, transmission loss, and conversion loss, balancing distribution of renewables does not necessarily required. In that case, we consider the upper bound of a storage capacity as well as a minimum threshold to reformulate the optimization problem on real-time distribution of renewables.

Acknowledgment

The authors would like to thank Prof. J. Brouwer, Associate Director of the National Fuel Cell Research Center (NFCRC) and Advanced Power and Energy Program (APEP) at UC Irvine, for his invaluable comments and suggestions.

References

[1] J.A. Momoh, *Smart grid design for efficient and flexible power networks operation and control*, IEEE/PES Power Systems Conference and Exposition (PSCE) 2009; pp.1-8.

[2] G. Locke, P. D. Gallagher, *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0*, NIST Special Publication 1108, January 2010.

[3] S. Hui, H. Chung, S. Yip, *A bidirectional ac-dc power converter with power factor correction*, IEEE Transactions on Power Electronics 2000; 15(5): pp. 942 - 949.

[4] K. Nakayama, N. Shinomiya, H.Watanabe, *An autonomous distributed control method based on tie-set graph theory in future grid*, International Journal of Circuit Theory and Applications, March, 2012. DOI: 10.1002/cta.1818.

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

[6] R. Rajagopal, E. Bitar, P. Varaiya, F. Wu, *Risk-Limiting Dispatch for Integrating Renewable Power*, International Journal of Electrical Power and Energy Systems, 2011.

[7] D. ONeill, M. Levorato, A.J. Goldsmith and U. Mitra, *Residential Demand Response Using Reinforcement Learning*, IEEE SmartGridComm, Oct. 2010, pp. 409-414.

[8] N. Li, L. Chen, S. H. Low, *Optimal Demand Response Based on Utility Maximization in Power Networks*, IEEE Power Engineering Society General Meeting, 2011, pp. 1-8.

[9] L. F. Bic, M. Fukuda, M. B. Dillencourt, *Distributed computing using autonomous objects*, IEEE Computer, Vol.29(8), pp. 55-61.

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

[11] J. Malinowski, *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.

[12] K. Nakayama, N. Shinomiya, H.Watanabe, *An autonomous distributed control method for link failure based on tie-set graph theory*, IEEE Transactions on Circuits and Systems-1, Regular Paper, 2012. DOI: 10.1109/TCSI.2012.2196109.

[13] K. Nakayama, N. Shinomiya, *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, pp. 957 - 964.

[14] T. Koide, T. Kubo, 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.