Chapter 1

Decentralized Models for Real-Time Renewable Integration in Future Grid

Dr. Kiyoshi NAKAYAMA

1.1 Introduction to Future Smart Grid

Future grid will likely support flexible bi-directional power flows and include energy production from multiple, disparate, and uncontrollable sources due to a high penetration of distributed renewable energy resources [1, 2, 3]. It is substantiated in [4] that battery systems save a large amount of generation cost without hurting customers' utility. Therefore, it is required to maximize the use and efficiency of renewable resources and realize optimal demand and power production responses that can complement renewable intermittency by incorporating storage systems [5, 6, 7].

For this reason, the following important issues need to be addressed:

- Balanced allocation of Distributed Energy Resources (DERs);
- Production of highly reliable and sustainable operation at all times;
- Preventing blackouts caused by sudden load peaks and imbalanced allocation of dispersed renewable energies;
- Reducing the adverse environmental impact (e.g. CO_2 emissions) and the cost of excessive power generation by fossil fuels, nuclear powers, etc.

To compensate the real-time and random nature of renewable generation, decentralized control has become a new direction in creating architectures of future power networks [8, 9]. In this chapter, we describe a decentralized model of future grid, formulate a problem to maximize real-time renewable integration, and introduces approaches for solving the problem in a decentralized manner.

1.2 Hybrid Model of Centralized Resource Management and Decentralized Grid Control

In a future grid, more and more devices are connected to the network(s) and controlled by remote systems. Many vendors are developing smart power devices these days that can be controlled through a network by providing access to the devices.

Therefore, the increasing number of energy systems and devices are connected with each other and exchange their state information and measurement data such as load profile and local generation. With the significant increase in the number of real-time smart devices, it is necessary to set up a centralized server to aggregate multiple types of energy devices as in Figure 1.1. However, it is not realistic for just one centralized server to aggregate and control all the devices exist in an entire grid since a grid may become quite large such as national transmission grid in the U.S. Even within a certain area of a country or state, numerous smart devices may exist so that the amount of real-time data that can constantly be collected from energy devices becomes huge and its management gets complicated.

Therefore, both centralized resource management and decentralized grid control is required to achieve efficient future grid operation. In particular, the use cases of the hybrid model of centralized management and decentralized control are described as follows:

1.2.1 Centralized Resource Management

Centralized management is needed to aggregate the resources that exist in certain geographical area. In particular, one Micro Grid is usually controlled by a centralized energy management system for reliable integration of distributed energy resources [10, 11].

The multiple resource types include residential devices such as Photovoltaics (PVs), Electric Vehicles (EVs), HVAC (heating, ventilating, and air conditioning) devices, and water pumps. The resources in grid-tied virtual power plants and micro grid may also be aggregated by their energy management system. In this case, larger-size stationary batteries, wind turbines will be considered. The forecasting technologies sometimes are applied not only to individual devices but also to aggregated resources. The typical model of aggregated centralized management is shown in Figure 1.1 where each client talks to individual devices to collect measurement data or issue a command to control them. For instance, the battery client collects the data of state of charge (SOC) of batteries and controls charging and discharging processes of them. The measurement data collected by the clients can be stored in the server's database so that the data could be used in optimizing the resources and the grid control.

1.2.1.1 Cloud Computing for Smart Grid

Cloud computing is becoming a way to make the power grid scalable by providing a variety of platforms as well as reduce operational costs and the environmental impact [12, 13]. A model for data management with cloud computing platform is discussed in [14], which includes real-time distributed data management, parallel processing of information retrieval. Amazon Web Services (AWS) [15] as a representative platform offers reliable, scalable, and inexpensive cloud computing resources and services for both Windows and UNIX environments. Therefore, a sever can be set up on top of a node or bus in the grid to aggregate data collected from a number of resources such as smart meters. In this way, one cloud server can support a num-

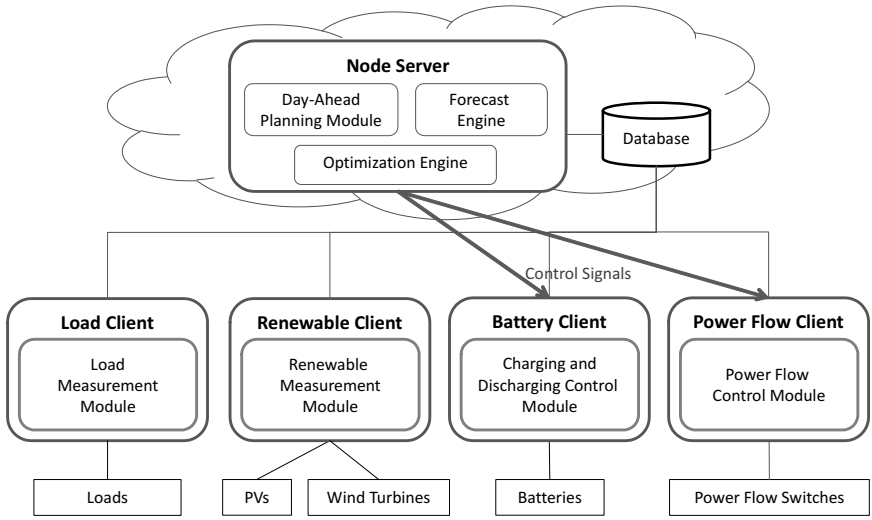


Figure 1.1: Example of cloud platform at a grid node.

ber of real-time devices whose data will be utilized for maximizing the renewable integration as cloud storage also provides efficient data management schemes. Figure 1.1 depicts the case where a server at a node or bus is installed in the cloud platform. The server and clients can communicate over the Internet by using HTTP communications. If the clients are also installed in the cloud platform, a gateway to communicate with local devices should be required to support the energy systems without cloud-based access point. A centralized database (cloud storage) serves as the core part of the platform. A cloud storage also provides multiple access interfaces such as Java Database Connection (JDBC) so that the collected measurement data can be stored in the database in real-time manner.

1.2.1.2 IoT (Internet of Things) Devices

The technology based on Internet of Things (IoT) [16] has been enabling a lot of network devices to be controlled remotely without manually changing operation settings for achieving autonomous network management. Smart grid is one of the domains where the concept and technologies of IoT is intensively applied [17]. Now a lot of power devices especially for residential areas feature remote access mechanisms such as local Application Programming Interface (API) and cloud-based open API. Therefore, more and more devices are becoming programmable and easy to access to obtain the data of load consumption, renewable generation, and energy level of a storage device [18] so that the aggregated computation can be realized by installing a server at each node in a grid and constantly collecting data from the real-time IoT devices.

The examples of the smart IoT devices are NEST thermostats and EVs sold by Greenlots supporting cloud-based API. On the basis of cloud API mechanism, a

remote server can go through the cloud platform provided by the vendor and directly access the device. The security mechanism utilized to support cloud API, among others, is OAuth that is an open standard for authorization.

1.2.2 Decentralized Grid Control

The power grid is adapting decentralized approaches to control bidirectional flow incurred by renewable energy sources more efficiently at consumption sectors. Intelligent grid automation makes the transition possible by integrating decentralized power management as well as communications technologies that enable smart grid modernization. Autonomous distribution networks have been designed in [19] to take scalability into account by dividing the legacy distribution network into a set of subnetworks. Decentralized control based on the utilization of overlapping information on load frequency is discussed in [20] to introduce distributed power generations and minimize the cost function of load frequency control problem. Decentralized control is also conducted in different applications in a smart/micro grid such as managing the change in local grid frequency and the active power generation within a micro grid [21] and controlling charging demands from EV customers [22, 23].

Therefore, control for an entire grid should be decentralized also because setting up a centralized system for a large-scale grid control will be inefficient especially in terms of resiliency. In addition, local autonomy is required due to communication times that are longer than those required to communicate grid perturbations and actuate hardware to prevent widespread outages. Local authority regimes also contain disparate policy and market frameworks in which the optimization and control infrastructure must operate. For this reason, development of an autonomous distributed architecture that can realize optimum allocation of complementary demand and power response together with dispersed renewable energy resources is inevitably required to produce highly efficient and reliable operation of future grid.

This chapter focuses on the decentralized control of an entire future grid to maximize the renewable integration.

1.3 Graph Modeling

Topological studies have been conducted in terms of power line communications [24, 25]. The topology of today's traditional grids must be modified to integrate and manage DERs when creating future grids as current grid topology employs a "tree" structure in the distribution system. However, the tree topology will not hold anymore when DER will be integrated into the grid since the topology is the worst in communication delay and power loss as the number of hops between nodes grows in the order of the size of the network [24]. In addition, recent attention has been focused on mesh topologies for development of new power network infrastructure because of the increased flexibility, efficiency, and resiliency they provide and the familiarity, good understanding and technology associated with the analogous information and communications network of the Internet [26].

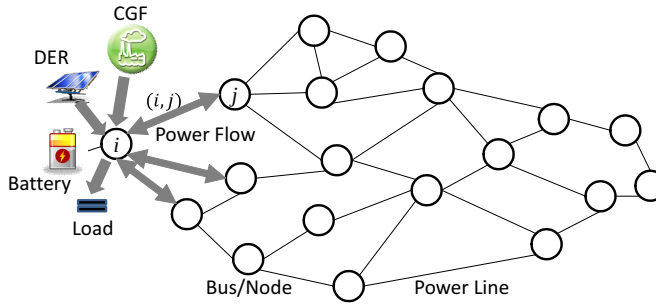


Figure 1.2: Graph modeling of a future grid.

The grid model used in this chapter is assumed to be a connected network that supports DC power flows such as a transmission grid so that th e concept could be applied to a variety of topologies. A connected grid can be expressed using a graph as $G = (\mathscr{V}, \mathscr{E})$ with a set of nodes $\mathscr{V} = \{1, 2, ..., n\}$ and a set of edges $(i, j) \in \mathscr{E}$ representing power lines or power routes among nodes where $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$. Each node can be a System Operating Point (SOP) that has the task of integrating growing amounts of renewable power, load processes, incoming and outgoing power flows, etc. The SOP makes decisions to balance the supply and load of electric power at every instant. As forecasts of these variables may have errors, a real-time approach to improve their accuracy was proposed on the basis of accumulating observations when supply and load are balanced [27]. Therefore, as in Figure 1.2, one node aggregates the process of loads, renewable generation, energy storage(s), and incoming and outgoing power flows by setting up a server. A grid has centralized power plant(s) that generate energy using traditional resources. Assume there is a facility that aggregates power plants using the generation by traditional resources defined as *Centralized Generation Facility (CGF)*. A SOP also controls the generation by CGF.

1.4 Maximizing Real-Time Renewable Integration

Real-time renewable integration becomes the key to achieve future smart grid since there are a lot of economical and social needs to reduce power generation based on the traditional energy resources. The traditional resources include fossil fuels (e.g. coal, gas) and nuclear resources except renewable energies. One of the important factors in integrating renewable energies is minimizing the power supply from CGF.

For any two vertices i, j $(i \neq j)$, let $f_{ij}(t)$ be a real power flow over an edge (i, j) at time t. The real power flow from node i to j in a power network at time t is

$$f_{ij}(t) = V_i V_j Y_{ij}(\theta_i(t) - \theta_j(t)), \quad (i, j) \in \mathscr{E}, \tag{1.1}$$

where V_i is the voltage at node i, $\theta_i(t)$ is the phase angle at node i at time t, and $Y_{ij} = Y_{ji}$ is the admittance on the link (i, j). Assume that $|\theta_i(t) - \theta_j(t)|$ is small enough to approximate $\sin(\theta_i(t) - \theta_j(t))$ by $|\theta_i(t) - \theta_j(t)|$. As voltages must be

constantly maintained close to 1 (usually between 0.95 and 1.05), we consider $V_i = 1$ at each node i. As a stability constraint, the angle difference $\theta_i(t) - \theta_j(t)$ between two neighboring nodes satisfies the following condition:

$$-\frac{\pi}{2} \leq \theta_i(t) - \theta_j(t) \leq \frac{\pi}{2} \tag{1.2}$$

Due to a capacity and/or a thermal safety limit, a power flow $f_{ij}(t)$ is bounded by the capacity $c_{ij}(t) > 0$ as follows:

$$-c_{ij}(t) \leq f_{ij}(t) \leq c_{ij}(t), \ \ (i,j) \in \mathscr{E}. \tag{1.3}$$

Even though future power networks become enough flexible such as information networks today, a capacity of a branch should still be considered.

Here, let $b_i(t)$ be the battery energy level of node $i \in \mathscr{V}$ at time t indexed by $t \in \mathscr{T} := \{1,2,...\}$. $b_i(t)$ is properly scaled according to each time step, so that in this case it is not needed to distinguish between energy and power.

An aggregated load $l_i(t)$ by a server of node i at time t is assumed to be an uncontrollable process.

Each node also has an aggregation function for distributed generation (DG) systems that exploit renewable energy resources. The aggregated amount of power derived from renewable generation devices is defined as $r_i(t)$ at time t, which is also an uncontrollable intermittent process. Then, *net demand* $d_i(t)$ is defined as follows:

$$d_i(t) := l_i(t) - r_i(t). \tag{1.4}$$

If the renewable generation is less than the load, node i demands power at time t; otherwise $d_i(t)$ is charged to a battery.

The generation by CGF $g_i(t)$, which is an controllable process, is bounded by

$$0 \leq g_i(t) \leq \overline{g}_i, \quad i \in \mathscr{V}, \tag{1.5}$$

where $\overline{g}_i$ is a CGF generation capacity at node i.

From Kirchhoff's laws, the *net power export* from node i to adjacent nodes at time t is given by

$$F_i(t) := \sum_{j \in \mathscr{V}} V_i V_j Y_{ij} (\theta_i(t) - \theta_j(t)), \ \ i \in \mathscr{V}. \tag{1.6}$$

If the net power export at node i is positive, i supplies power at time t; otherwise i demands power.

Then, the battery energy level $b_i(t)$ at node i at time t follows the dynamics as

$$b_i(t) = b_i(t-1) - d_i(t) + g_i(t) - F_i(t), \ \ i \in \mathscr{V}, \tag{1.7}$$

with a given initial energy level $b_i(0) \geq 0$. The battery energy level $b_i(t)$ is bounded by a battery capacity where $0 \leq b_i(t) \leq \overline{b}_i$.

Some of the factors are determined in a day-ahead planning phase. For instance, the battery may have to be charged while electricity price is cheap and discharged during peak-price time. In addition, there are load peaks especially in summer season so that batteries should be discharged to smooth the peaks. As this chapter focuses

on real-time control, a day-ahead planning scheme of optimal state of charge for batteries is not discussed. Instead, *Lower Energy Limit (LEL)* $e_i(t)$ is defined in case that a battery needs to satisfy a certain state of charge at time t. Then, the boundary condition of a battery at node i at time t is

$$e_i(t) \leq b_i(t) \leq \overline{b}_i, \quad i \in \mathscr{V}, \tag{1.8}$$

with a given initial LEL $e_i(0)$. $\Delta e_i(t) = \overline{b}_i - e_i(t)$ is preserved for unexpected renewable generation peak when $r_i(t)$ is larger than the load process $l_i(t)$. In this case, the net demand becomes negative as $d_i(t) < 0$, so that $|d_i(t)|$ is stored in the battery at node i.

Given an initial battery energy level $b_i(0)$, battery capacity $\overline{b}_i$, LEL $e_i(t)$, load $l_i(t)$, and renewable generation $r_i(t)$ at each time step t for each $i \in \mathscr{V}$, and a link capacity $c_{ij}(t)$ and admittance Y_{ij} for each $(i, j) \in \mathscr{E}$, the objective function to maximize the renewable integration at each time step $t \in \mathscr{T}$ is

$$\begin{aligned} \text{Minimize} \quad & \sum_{i \in \mathscr{V}} g_i(t), \\ \text{Over} \quad & g, f, F, b, \theta, \\ \text{Subject to} \quad & (1.1) - (1.8). \end{aligned} \tag{1.9}$$

1.5 General Decentralized Approaches

The principle that underlies decentralized algorithms is that each node tries to solve a problem by exchanging messages with other peripheral nodes. All the basics of distributed algorithms are described in [28]. The application of distributed approaches can also be achieved by setting up HTTP severs and clients. For example, once the HTTP methods and functions are implemented in the server of a node, a node can talk to other nodes with HTTP client functions over the Internet. Communications among the nodes can therefore be implemented in multiple ways.

There are two typical scenarios when conducting decentralized control for a power grid modeled by a graph:

- Communicating with adjacent nodes only (nodal approach)
- Clustering a grid into a set of segments (clustering approach)

The nodal approach is a typical way to realize distributed control within a grid and its mechanism often becomes the foundation of a lot of other decentralized approaches. A message that is exchanged among the nodes can be a notification signal or a message that contains state information, measurement data of devices, etc. For example, if a communications scheme exploits HTTP protocols, simple GET and POST, Extensible Markup Language (XML), JavaScript Object Notation (JSON), among others, can be supported to transfer those data.

The clustering approach based on graph modeling divides a network into set of segments where the segment can be a tree, cut sets, loops, among others. Sometimes clustering controls demonstrate better performance than nodal approaches in terms of failure detection or flow controls [29].

1.6 Distributed Nodal Approach

Distributed Nodal Approach (DNA) is a representative approach to solve distributed problems.

In this scenario to solve the formulated problem, the message includes the value of the function that evaluates a node with certain predefined criteria. When deciding the process priority for overlapping resources shared by several adjacent nodes, each node i exchanges the value of the *Nodal Evaluation Function (NEF)* denoted as $\Phi_i(t)$ with its adjacent nodes $\mathscr{A}_i$. NEF is subject to objective functions, constraints, characteristics of a grid, etc. Here, for each node $i \in \mathscr{V}$, a *Battery Penalty Cost (BPC)* $H_i(t)$ is defined as

$$H_i(t) := \begin{cases} 0, & \text{if } b_i(t) \geq e_i(t) \\ \alpha_i(e_i(t) - b_i(t)), & \text{if } b_i(t) < e_i(t) \end{cases} \tag{1.10}$$

with some penalty coefficient $\alpha_i > 0$. If BPC is utilized for NEF, $\Phi_i(t) = H_i(t)$. When the process priority cannot be determined by NEF with BPC, node i can use precomputed value or predefined ID such as MAC Address.

Nodal Flag (NF) denoted as ζ_i is used to distinguish a node in process from a stand-by node once the process priority is decided using NEF. The definition of NF is as follows:

- $\zeta_i = 0$: Node i is Stand-By
- $\zeta_i = 1$: Node i is In Process

When $\zeta_i = 1$, all the adjacent nodes $\mathscr{A}_i$ must stand by setting its NF as $\zeta_i = 0$ until node i finishes its procedure. Parallel optimization among nodes is feasible with the concept of NF.

A *Nodal Evaluation Function Message (NEFM)* is utilized when exchanging the value of NEF $\Phi_i(t)$ with adjacent nodes $\mathscr{A}_i$. A *Nodal Flag Signal (NF Signal)* is used for exchanging the binary value of NF of node i. DNA exchanges NEFM asynchronously without any global clock. Flowchart of DNA conducted in each node i is described in Figure 1.3. The procedure of DNA is mainly consists of *Initialize*, *Send*, *Receive*, *Optimize*, *Notify*, *Confirm*, and *StandBy*.

A) Initialize

In *Initialize*, NEF and NF of node i are set as $\Phi_i(t) = 0$ and $\zeta_i = 0$, respectively. Net demand is calculated as $d_i(t) = l_i(t) - r_i(t)$ based on the measurement data of load and renewable generation. In addition, set $g_i(t)$, $\theta_i(t)$, and $F_i(t)$ to 0 so that at this moment of initialization, the battery energy level is $b_i(t) = b_i(t-1) - d_i(t)$.

B) Send

The value of NEF $\Phi_i(t)$ is calculated based upon the measurement data of node i at time t. After calculating $\Phi_i(t)$, node i writes its value of NEF in NEFM. Then, node i sends the NEFMs to all the adjacent nodes $\mathscr{A}_i$.

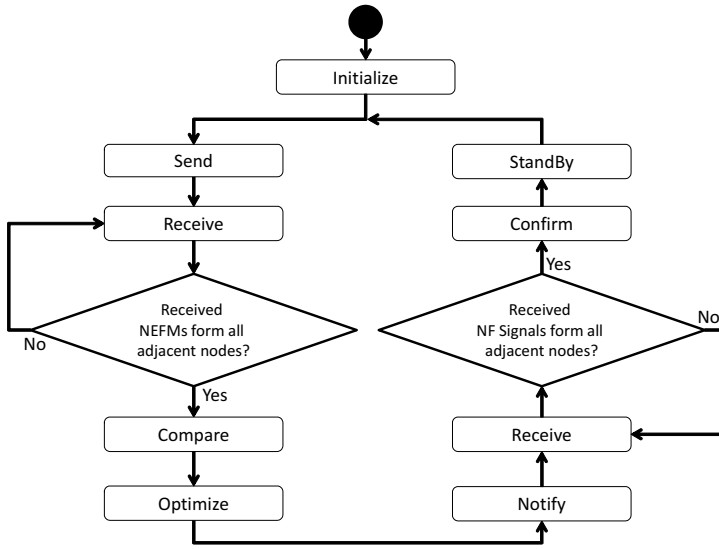


Figure 1.3: Flowchart of Distributed Nodal Approach (DNA).

C) Receive

Receive is called when node i receives a message (e.g. NEFM, NF Signal) from an adjacent node $j \in \mathscr{A}_i$. In case that one of the other procedures of DNA is not completed, the message is temporarily stored in node i. Until node i receives a message from all the adjacent nodes $\mathscr{A}_i$, node i stands by to receive another NEFM. Once i received all the messages from $\mathscr{A}_i$, it moves on to the next procedure.

D) Compare

After receiving all the NEFMs form $\mathscr{A}_i$, node i compares its value of $\Phi_i(t)$ with the value of $\Phi_j(t)$ of each adjacent node $j \in \mathscr{A}_i$. If the value of $\Phi_i(t)$ is the largest among those of all the adjacent nodes, node i sets its NF as $\zeta_i = 1$, otherwise $\zeta_i = 0$. If the value of $\Phi_i(t)$ is the same as the value of $\Phi_j(t)$ of node j, node i uses another NEF to decide the procedure priority such as a precomputed value $\Phi_i^p(t)$ that is unique for each node i.

E) Optimize

The most common way to solve the problem on maximizing renewable integration (1.9) would be exploiting the notion of Linear Programing that has been applied to many linear-optimization problems. As we are solving (1.9) in a decentralized manner, Distributed Linear Programing (DLP) approach is formulated in Algorithm 2 where each node i tries to minimize the amount of power generated at its CGF $g_i(t)$ under the constraints (1.1) - (1.8) by communicating with adjacent nodes $\mathscr{A}_i$.

Algorithm 1 Compare

```
Set NF as ζ_i = 1.
for each j ∈ 𝒜_i do
    if H_i(t) < H_j(t) then
        Set NF as ζ_i = 0.
    else if H_i(t) = H_j(t) then
        if Φ_i^p(t) < Φ_j^p(t) then
            Set NF as ζ_i = 0.
        end if
    end if
end for
```

If $\zeta_i = 1$, node i conducts Algorithm 2. After conducting DLP, node i sets its NF as $\zeta_i = 0$.

F) Notify

In *Notify*, node i sends NF signal to each adjacent node $j \in \mathscr{A}_i$ to notify that $\zeta_i = 0$.

G) Confirm

Confirm is called when node i receives NF Signals from an adjacent node $j \in \mathscr{A}_i$ after *Optimize*. Until node i receives NF signals from all the adjacent nodes $\mathscr{A}_i$, node i waits for another NF signal. After receiving all the NF signals form $\mathscr{A}_i$, node i confirms that each NF of $j \in \mathscr{A}_i$ is $\zeta(L_j) = 0$.

H) StandBy

Let Δt be a communication interval among adjacent nodes. In StandBy, node i waits for Δt minute(s), and then proceed to the next round of DNA as in Figure 1.3.

The procedure of DNA can be terminated if a node receives a termination signal.

1.7 Distributed Clustering Approach

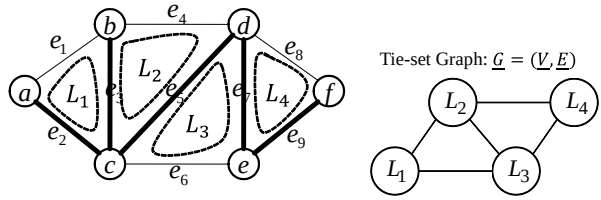
Graph clustering or partitioning approaches have often utilized when dividing a network into a set of segments. For most of the cases, the partitioning is conducted in a centralized manner so that there is a master node that grasp the entire topology of the partition.

Although there are many partitioning approaches have been provided, the concept of "cut-set" and "tie-set" has been used with the Kirchhoff's Laws (e.g. Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL)). Refer to the literature [29, 30, 31] for how to configure the cut-set and tie-set structure in a network.

As one of the clustering approaches on maximizing renewable integration, we introduce an approach based on Tie-Set Graph Theory.

Algorithm 2 Distributed Linear Programing (DLP) Algorithm

if NF $\zeta_i = 1$ **then**
  Set $flag_{opt} = 1$.
  for each $j \in \mathscr{A}_i$ **do**
    if $b_j(t) < e_j(t)$ **then**
      Set $flag_{opt} = 0$.
    end if
  end for
  if $flag_{opt} = 1$ **then**
    $\theta_{min} \leftarrow +\infty$.
    for each $j \in \mathscr{A}_i$ **do**
      $\theta_{tmp} := -\frac{b_j(t)-e_j(t)}{V_i V_j Y_{ij}}$.
      if $|\theta_{tmp}| < |\theta_{min}|$ **then**
        $\theta_{min} \leftarrow \theta_{tmp}$.
      end if
    end for
    if $|F_i(t)| > e_i(t) - b_i(t)$ **then**
      $\theta_i(t) \leftarrow -\frac{e_i(t)-b_i(t)}{\sum_{j\in\mathscr{A}_i} V_i V_j Y_{ij}}$.
      $g_i(t) = 0$.
      $F_i(t) = \sum_{j\in\mathscr{A}_i} V_i V_j Y_{ij} \theta_i(t)$.
    else
      $\theta_i(t) \leftarrow \theta_{min}$.
      $F_i(t) = \sum_{j\in\mathscr{A}_i} V_i V_j Y_{ij} \theta_i(t)$.
      $g_i(t) \leftarrow e_i(t) - b_i(t)$.
    end if
  else
    $g_i(t) \leftarrow e_i(t) - b_i(t)$.
    $\theta_i(t) = 0$.
    $F_i(t) = 0$.
  end if
  Set NF as $\zeta_i = 0$.
else
  Skip *Optimize*.
end if



(a) A fundamental system of tie-sets $\{L_1, L_2, L_3, L_4\}$.

(b) The tie-set graph of Figure 1.4(a).

Figure 1.4: An example of a fundamental system of tie-sets and its tie-set graph.

1.7.1 Tie-set Graph Theory and its Application to Distributed Systems

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, resiliency, and optimization perspectives. The tie-set graph theory has been described in [31, 32, 33] and applied to many areas of smart grid applications [34, 35]. Here, we introduce the basis for the unfamiliar reader.

1.7.1.1 Fundamental System of Tie-sets

For a given bi-connected graph $G = (\mathscr{V}, \mathscr{E})$ with a set of vertices $\mathscr{V} = \{1, 2, ..., n\}$ and a set of edges $\mathscr{E} = \{e_1, e_2, ..., e_m\}$, let $L_i = \{e_1^i, e_2^i, ...\}$ be a set of all edges that constitutes a loop in G. The set of edges L_i is called a *tie-set* [30]. Let T and $\overline{T}$ respectively be a spanning tree and a cotree of G, where $\overline{T} = \mathscr{E} - T$. For $l \in \overline{T}$, $T \cup \{l\}$ includes one tie-set. Focusing on a subgraph $G_T = (\mathscr{V}, T)$ 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 tie-set* that consists of the path P_T and the edge l is uniquely determined as $L(l) = \{l\} \cup P_T(b, a)$. It is known that $\mu = |\overline{T}|$ fundamental tie-sets exist in G, and they are called a *fundamental system of tie-sets* $\mathbb{L}_B = \{L_1, L_2, ...L_\mu\}$. A fundamental system of tie-sets guarantees that it covers all the vertices and edges (Figure 1.4(a)) even in a bi-connected and non-planar graph G (Figure 1.5).

Fundamental tie-sets are independent of each other; any fundamental tie-set cannot be obtained by calculus $\oplus$[1] among other tie-sets.

1.7.1.2 Tie-set Graph

A graph $\underline{G} = (\underline{\mathscr{V}}, \underline{\mathscr{E}})$ is defined as a *tie-set graph*, where a set of vertices $\underline{\mathscr{V}}$ corresponds to a fundamental system of tie-sets $\{L_1, L_2, ..., L_\mu\}$, and a set of edges $\underline{\mathscr{E}}$ corresponds to a set of edges $\{\underline{e}(L_i, L_j)\}, (i \neq j)$, which represent the connections among tie-sets. $\underline{e}(L_i, L_j)$ is determined by the set of common vertices of L_i and L_j.

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

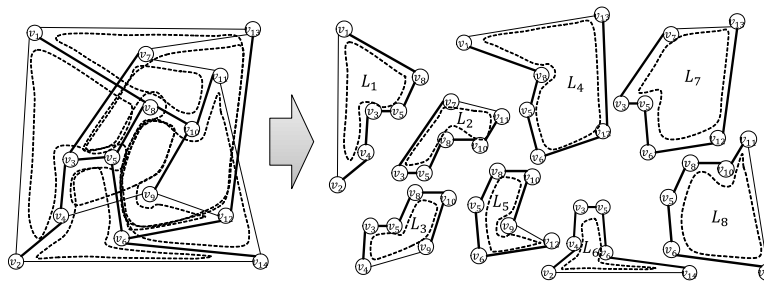


Figure 1.5: A fundamental system of tie-sets in a non-planar graph.

Let $\mathscr{V}(L_i)$ be a set of all the vertices included in a tie-set L_i. If $\mathscr{V}(L_i) \cap \mathscr{V}(L_j) \neq \emptyset$, L_i and L_j have an edge $\underline{e}(L_i, L_j)$.

Each fundamental tie-set of a given graph G is uniquely mapped to the specific tie-set graph $\underline{G}$ as shown in Figure 1.4(b).

1.7.1.3 Tie-set Information $\mathbb{L}_i$

Tie-set Information is the State Information of tie-set(s) to which the node i belongs. When node i belongs to L_i where $i \in \mathscr{V}(L_i)$, i has Tie-set Information of L_i. The node c in Figure 1.4(a), for example, has information of fundamental tie-sets $\{L_1, L_2, L_3\}$ as Tie-set Information.

1.7.1.4 Communications among Tie-Sets

After constructing Tie-set Information at each node, a leader node v_l^i is selected in each tie-set L_i by exchanging state information of the nodes within L_i. 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. The criteria to decide a leader of a tie-set are 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 [36]. A leader node v_l^i of a tie-set L_i has information of *Adjacent Tie-sets* $\mathbb{L}_i^a = \{L_1^i, L_2^i, ...\}$ so that a leader v_l^i of L_i communicates with leaders of adjacent tie-sets $\mathbb{L}_i^a$. An adjacent tie-set L_j of L_i is determined according to the relation of connection $\underline{e}(L_i, L_j) \in \underline{\mathscr{E}}$ of $\underline{G}$. For example, adjacent tie-sets of L_1 in Figure 1.4(a) are $\mathbb{L}_1^a = \{L_2, L_3\}$ so that the leader node of L_1 constantly communicates with leaders of $\mathbb{L}_1^a$ using its precomputed routing table as explained in [36].

1.7.1.5 Tie-set Agent

Tie-set Agent (TA) is an autonomous agent that constantly navigates a tie-set L_i around and brings state information of nodes in L_i to its leader node v_l^i. The concept of a distributed intelligent agent system with autonomous agents are explained in [37]. TA contains a measurement vector $y_i(t)$ that provides the value of current loads $l_i(t)$ and renewable power generations $r_i(t)$, where $i \in \mathscr{V}(L_i)$. 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 $y_i(t)$ and decides the procedure in L_i based upon $y_i(t)$. TA constantly brings the measurement data $y_i(t)$ of L_i to its leader node with certain time interval.

1.7.1.6 Tie-set Evaluation Function

Tie-set Evaluation Function (TEF) is the function that evaluates a tie-set with certain predefined criteria. When deciding the process priority for overlapping resources shared by several adjacent tie-sets, each tie-set exchanges the value of TEF denoted as $\Phi(L_i,t)$ with its adjacent tie-sets $\mathbb{L}_i^a$. TEF $\Phi(L_i,t)$ is calculated based upon the current measurement vector $y_i(t)$.

Here, TEF is defined using BPC as follows:

$$\Phi(L_i,t) = \sum_{i \in \mathscr{V}(L_i)} H_i(t). \tag{1.11}$$

For instance, as BPC is used for TEF, the TEF values of tie-sets in the upper network in Figure 1.6 are

$$\begin{aligned}
\Phi(L_1,t) &= H_a(t)+H_b(t)+H_c(t) = 25.0,\\
\Phi(L_2,t) &= H_b(t)+H_c(t)+H_d(t) = 12.0,\\
\Phi(L_3,t) &= H_c(t)+H_d(t)+H_e(t)+H_f(t) = 24.0,\\
\Phi(L_4,t) &= H_e(t)+H_f(t)+H_g(t)+H_h(t) = 28.0,
\end{aligned}$$

as LELs of all nodes in the example network is set as $e_i(t) = 40.0$. Therefore, the NFs of $\{\zeta(L_1),\zeta(L_2),\zeta(L_3),\zeta(L_4)\}$ are set as $\{1,0,0,1\}$ so that L_1 and L_4 are in process in Figure 1.6.

1.7.2 Tie-set Based Optimization Algorithm

Tie-set Based Optimization (TBO) Algorithm is described in Algorithm 3 when an tie-set L_i optimizes its resources. In a tie-set graph $\underline{G}$, each node represents a tie-set itself. Therefore, if we consider a tie-set as a node, the distributed nodal approach can be directly applied to the parallel computation among tie-sets where each tie-set communicates with adjacent tie-set(s) in order to decide which tie-set conducts optimization. The tie-set that conducts optimization calculates the power flows $f_{ij}(t)$ that realize the balanced renewable energy sharing as well as decide the amount of CGF generation $g_i(t)$.

STEP 0: Step 0 is an initialization procedure. For each node i in a tie-set L_i, its angle $\theta_i(t)$ and CGF generation $g_i(t)$ are set as 0. Battery energy levels in L_i from a previous time step $b_i(t-1)$ have been preserved in a leader node v_l^i of L_i at time t. As information of loads $l_i(t)$ and renewables $r_i(t)$ in L_i are provided by TA, the net demand is decided as $d_i(t) := l_i(t) - r_i(t)$. Then, the net battery energy level is calculated as $b_i^d(t) := b_i(t-1) - d_i(t)$.

STEP 1: In Step 1, the angles of nodes in L_i are calculated, and thus power flows on the edges are decided. We define a $flag = \{0,1\}$ to decide whether or not L_i

Algorithm 3 Tie-set Based Optimization (TBO) Algorithm

STEP 0: Initialzation

for each $i \in \mathscr{V}(L_i)$ **do**
  $\theta_i(t) = 0$, $g_i(t) = 0$.
  $d_i(t) := l_i(t) - r_i(t)$.
  $b_i^d(t) := b_i(t-1) - d_i(t)$.
end for

STEP 1: Deciding Angles for Power Flows

Set $falg = 1$.
for each $i \in \mathscr{V}(L_i)$ **do**
  if $b_i^d(t) > e_i(t)$ **then**
    $flag = 0$.
  end if
end for
while $flag \neq 1$ **do**
  Select $i \in \mathscr{V}(L_i)$, where $b_i^d(t)$ is maximum.
  Calculate $\theta_i'(t) = \frac{b_i^d(t) - e_i(t)}{\sum_{j \in \mathscr{A}_i} V_i V_j Y_{ij}}$.
  Update $\theta_i(t) \leftarrow \theta_i(t) + \theta_i'(t)$.
  for each $j \in \mathscr{A}_i$ **do**
    Update $b_i^d(t) \leftarrow b_i^d(t) - V_i V_j Y_{ij} \theta_i'(t)$.
    Update $b_j^d(t) \leftarrow b_j^d(t) + V_i V_j Y_{ij} \theta_i'(t)$.
  end for
  Set $flag = 1$.
  for each $i \in \mathscr{V}(L_i)$ **do**
    if $b_i^d(t) < e_i(t)$ **then**
      Set $flag = 0$.
    end if
  end for
  $b_{max} := \max_{i \in \mathscr{V}(L_i)} \{b_i^d(t)\}$.
  if $b_{max} - e_i(t) \leq \varepsilon$ **then**
    $flag = 1$.
  end if
end while
Set angles $\theta_i(t)$ for all nodes $i \in \mathscr{V}(L_i)$ with (1.2).
Distribute power flows in L_i according to (1.1) and (1.3).

STEP 2: Deciding Power Supply from CGF

for each $i \in \mathscr{V}(L_i)$ **do**
  if $b_i(t) < e_i(t)$ **then**
    $g_i(t) = e_i(t) - b_i(t)$.
    CGF supplies power to i with the amount of $g_i(t)$.
  end if
end for

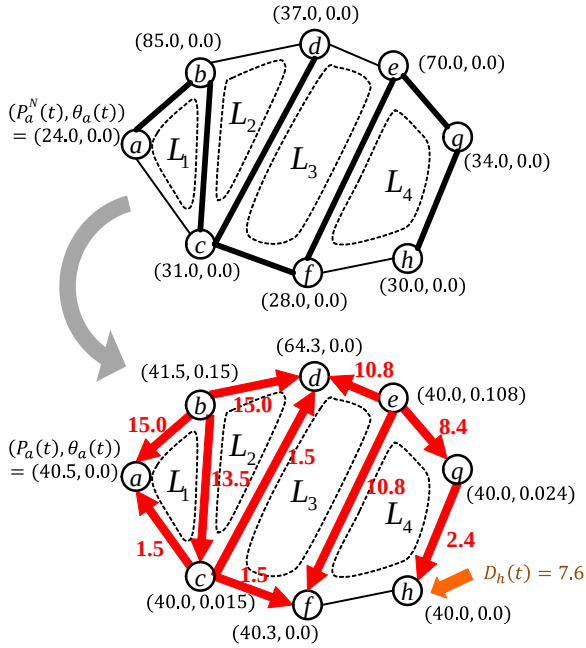


Figure 1.6: Parallel optimization in L_1 and L_4.

conducts the **while** loop of Algorithm 3. If there is a node in L_i that has energy that can be distributed to other nodes, that is, $b_i^d(t) > e_i(t)$, then $flag = 0$. Otherwise, L_i goes to Step 2 without distributing energy of nodes. In the **while** loop, phase angles are calculated in order to balance the battery energy levels. The node with maximum energy that meets $b_i^d(t) > e_i(t)$ distributes its power to other nodes until it reaches LEL $e_i(t)$. The **while** procedure stops if all the nodes in L_i store energy greater than or equal to LEL where $b_i^d(t) \geq e_i(t)$ is satisfied. The **while** procedure also stops if energy of the nodes that meet $b_i^d(t) > e_i(t)$ converges on its LEL $e_i(t)$, where a certain small value ε is used to check the convergence.

After calculating the angles, the leader v_l^i notifies the nodes in L_i of the magnitude of angles to distribute power flows.

STEP 2: Step 2 decides the amount of power generated at CGF $g_i(t)$ at nodes in L_i. With the LEL $e_i(t)$ at each node in a tie-set, if node i does not have minimum level of power as in $b_i(t) < e_i(t)$, then CGF supplies power to the node i with the amount of $g_i(t) = e_i(t) - b_i(t)$.

1.8 Case Study of Decentralized Grid Control

In order to validate the behavior of the decentralized model facing with different levels of uncertainty that are seen in load processes and renewable generation, case studies are discussed on the performance of node i and j of the network in Figure

1.2, which are shown in Figure 1.7(a) and 1.7(b). The distributed nodal approach is exploited to validate the behavior of the processes.

At time $t = 0$, the battery energy level and LEL of nodes i and j are given with $b_i(0) = b_j(0) = 60$ and $e_i(0) = e_j(0) = 20$, respectively. As in the case studies, nodes i and j increase their LEL $e_i(t)$ and $e_j(t)$ to cope with sudden load peaks. For instance, at $t = 30$ and 135, $e_i(30)$ and $e_i(135)$ at node i are updated as there are sudden load peaks expected in control planning phase. This procedure demonstrates that those nodes can deal with the uncertainty of load processes by changing their LEL in order to prevent the nodes from running out of power.

There also are sudden renewable peaks as indicated in the Figures 1.7(a) and 1.7(b). For example, when $t = 45$, 75, 105, and 180 at node i, the renewable generation is higher than the load process. In this case, some amount of the sudden renewable generation is stored in the battery of node i, and then shared to the adjacent nodes at the next time step. This also demonstrates the effectiveness of having preserved space $\Delta e_i(t) = \overline{b}_i - e_i(t)$ for the renewable peak as the excess amount of renewable generation can be temporarily stored at the node, and then delivered to the adjacent nodes that require power to satisfy their net demands.

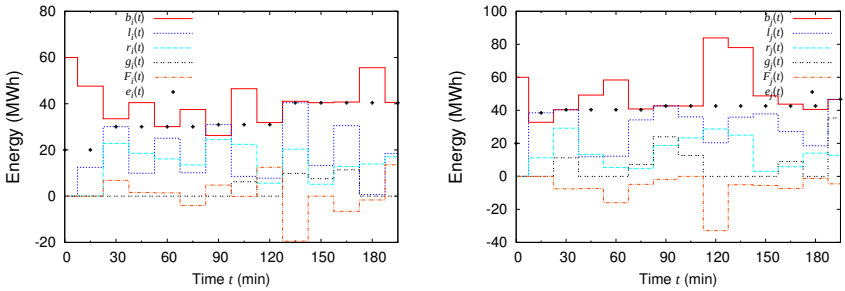
Power flows between the nodes i and j on the case studies also demonstrate the framework that each node distributes power to the other peripheral nodes when a node has enough power, that is, $b_i(t) > e_i(t)$. At the same time, the node in short of energy ($b_i(t) < e_i(t)$) receives incoming power flows with negative value of $F_i(t)(< 0)$ from the nodes that have the excess amount of energy. For example, as node i had enough energy from the previous time step $t = 15$, i shares its energy to the adjacent nodes at $t = 30$. The portion of the energy from i is sent to node j as seen in the incoming power flow $F_j(30)$, which satisfies the net demand $d_j(30) = l_j(30) - r_j(30)$ together with supplementary CGF generation $g_j(30)$. Thereby, the node j makes up for the first load peak that happened from $t = 15$ to 30.

The process of CGF constantly compensates the net demand that has been partially satisfied by incoming power flows. As in the case studies, although node i obtains incoming power flows at $t = 135$, i still needs more power to meet $b_i(135) \geq e_i(135)$. In order to satisfy the demand, CGF supplies power to i with the amount of $g_i(135)$.

1.9 Simulation and Experiments

Let us introduce the simulation results based on the decentralized control models described in previous sections. In a bi-connected power network, power flows can pass bi-directionally. Each node employs common buffering method and polling mechanism.

A simulation network $G = (\mathscr{V}, \mathscr{E})$ is created with 100 nodes and 190 links with random connections. Aggregated data of loads $l_i(t)$ and renewable energies $r_i(t)$ at each node are available every 15 minutes from smart meters. The profiles of loads and renewables are based on the data provided in [38, 39, 40] so that each node has different profiles with random nature. For a given time sequence ($\mathscr{T} = \{0, 15, 30, ...\}$) at each node i, $l_i(t)$ and $r_i(t)$ are respectively obtained. The com-



(a) Performance at node i in Figure 1.2.

(b) Performance at node j in Figure 1.2.

Figure 1.7: Case studies for performance over time facing with different levels of uncertainty.

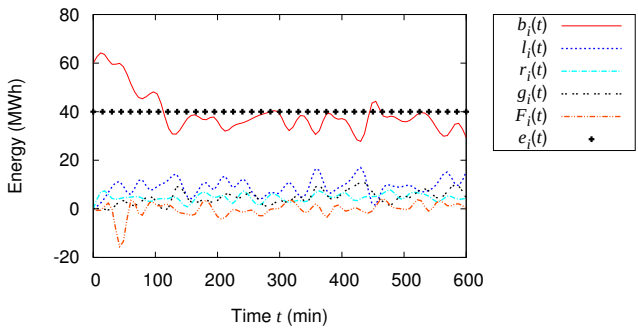


Figure 1.8: Energy stimulus response of $b_i(t)$, $l_i(t)$, $r_i(t)$, $g_i(t)$, and $F_i(t)$ from $t = 0$ to 600 where load and renewable generation are given with $0 \leq l_i(t) \leq 20$ MWh and $0 \leq r_i(t) \leq 10$ MWh, respectively.

munication interval Δt in StandBy of Figure 1.3 is defined as 1 minute. In this experiment, the voltages are set to be $V_i = 1$ MV, and each admittance is set to be $Y_{ij} = 100$. The capacity of each battery is $\overline{b}_i = 100$ MWh. The initial energy $b_i(0)$ of each node is assigned as $b_i(0) = 60$ MWh. The maximum flow (capacity) on each edge is $c_{ij}(t) \equiv 50$ MW. α_i of BPC defined in (1.10) is set as 1. ε in Algorithm 3 is set to be 0.1.

Following sections describe the simulation results by the distributed clustering approach based on tie-sets. In this network, the number of tie-sets is uniquely decided to be 91. Every measurement vector $y_i(t)$ is sent to the leader node of each tie-set constantly using a Tie-set Agent.

1.9.1 Energy Stimulus Response

In this experiment, we look at the simulated behavior of energy stimulus response by picking up several nodes with different load and renewable profiles. Initial LELs are set as $e_i(0) = 40$ MWh.

Figure 1.8 shows the changing process of energy stimulus response of $b_i(t)$, $l_i(t)$, $r_i(t)$, $g_i(t)$, and $F_i(t)$ from $t = 0, 15, 30, ..., 600$ (min) at node i. At this node, the load input and renewable generation are randomly given between 0 to 20 MWh and 0 to 10 MWh, respectively. As the decentralized approach based on tie-sets is applied to controlling $g_i(t)$ and $F_i(t)$, the value of $b_i(t)$ keeps to be around 40 MWh from $t \approx 100$ min. The value of $e_i(t)$ is always 40 MWh as the load is bounded by 20 MWh. It is also indicated that i distributes power flows to adjacent nodes when it has an excess amount of battery energy. When the size of the battery is enough large compared with the load and renewable generation, the stimulus response for battery energy level $b_i(t)$ becomes stable. Therefore, it realizes sustainable control against the randomness of loads and renewables in case of installing large-size energy storages.

Next, we look at the node where the magnitude of load and renewable generation is large against the size of energy storage. Figure 1.9 is the energy stimulus response at another node i where the load input is assigned with the range of 0 to 80 MWh, and the renewable generation is given between 0 to 60 MWh. There are many sudden peak load processes that make convergence of battery energy level at i unstable. However, the iteration of decentralized optimization constantly tries to cope with the uncertainty of load processes so that the energy at node i never runs out. The behavior of LEL at this node shows dynamic change according to the load peaks, which is also the key control point to prevent a node from running out of energy. The experiments shown in Figure 1.8 and 1.9 utilize larger load bound than maximum renewable generation. Therefore, we analyze the node that exploit larger renewable generation than the process of load.

At a node at which the maximum load and renewable generation are respectively bounded by 40 MWh and 80 MWh as shown in Figure 1.10, the energy level is generally greater than the LEL at almost all times unless net demand is positive and large. As $\Delta e_i(t) = \overline{b}_i - e_i(t)$ is preserved for unexpected renewable peaks, the preserved space is effectively used to temporally store excess renewable generation. It is also indicated that when the node cannot store the excess amount of renewable, it distributes the energy to the other nodes that require it. The process of this node contributes to minimizing total amount of energy generated at CGFs as this node constantly distributes the DER stored in its battery.

From the results shown in Figures 1.8 to 1.10, it is demonstrated that the decentralized model copes with unexpected load peaks and renewable generation by dynamically controlling power flows and CGF generation.

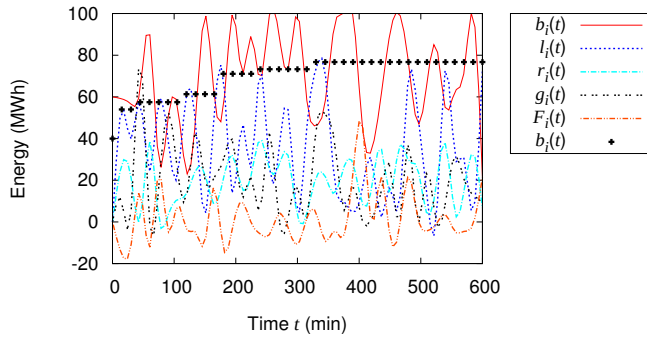


Figure 1.9: Energy stimulus response where load and renewable generation are given with $0 \leq l_i(t) \leq 80$ *MWh and* $0 \leq r_i(t) \leq 60$ *MWh, respectively.*

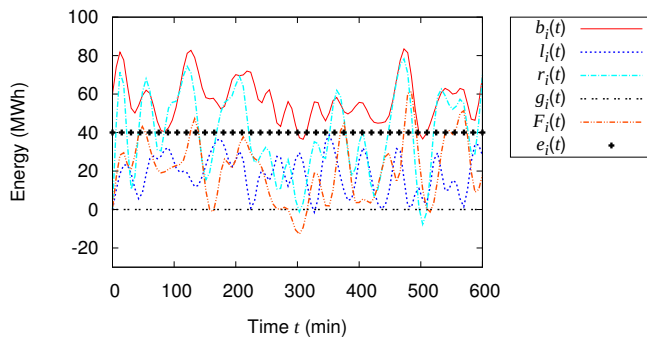


Figure 1.10: Energy stimulus response where load and renewable generation are given with $0 \leq l_i(t) \leq 40$ *MWh and* $0 \leq r_i(t) \leq 80$ *MWh, respectively.*

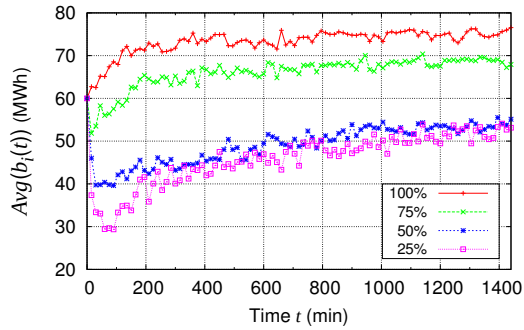


Figure 1.11: Convergence behavior of the average battery energy level $Avg(b_i(t)) = \overline{B}(t)$ *with different Renewable Penetration Rates (RPRs) (100%, 75%, 50%, 25%) in a 100-node power network.*

1.9.2 Convergence with Different Renewable Penetration Rates

We conducted simulation with different Renewable Penetration Rates (RPRs) as it is an important issue to be considered when integrating renewable energies. Here, the RPR is adjusted by changing the boundary of renewable generation capacity. In this simulation, load input is random bounded by 60 MWh at each node i at each time t. When expecting RPRs to be 100%, 75%, 50%, and 25% as in Figure 1.11, the random renewable input is bounded by 60 MWh, 45 MWh, 30 MWh, and 15 MWh at each node i at each time t, respectively. Initial LELs are set as $e_i(0) = 20$ MWh.

Simulation data of $b_i(t)$ of all the nodes taken at $t = 0, 15, 30, ..., 1440$ min (24 hours) are averaged. Hence $Avg(b_i(t)) = \overline{B}(t)$, which stands for the average battery energy level at time t defined as follows, is used to see the convergence behavior of entire grid energy level.

$$\overline{B}(t) = \frac{\sum_{i \in \mathcal{V}} b_i(t)}{|\mathcal{V}|} \tag{1.12}$$

Experiments are conducted 20 times for each case of RPR to average the results.

Figure 1.11 shows the simulated behavior of the average battery energy level $\overline{B}(t)$ in a 100-node power network with different RPRs. When RPR is 100%, the average battery energy level $\overline{B}(t)$ converges around 75 MWh since each node stores the renewable energy in case the amount of renewable generation is larger than the load process. After the average battery energy level draws to 75 MWh, power flows among nodes become stable by distributing stored energy to the nodes in short of energy with minimum energy supply from CGF. The similar behavior can be seen when RPR is 75% where $\overline{B}(t)$ converges around 70 MWh. In case of 50% and 25% RPRs, the average powers $\overline{B}(t)$ first drops since the initial LELs are $e_i(0) = 20$ MWh at all the nodes. The values of $e_i(t)$ are dynamically updated according to the peak load profile at each node over the time range. Thus, the average energy gradually converge to 60 MWh with the updated LELs $e_i(t)$.

It has been verified that the convergence behavior of the average battery energy level in 200-node to 500-node networks with the different RPRs from 25% to 100% is similar to the behavior in the 100-node network. In addition, it is confirmed that the convergence process does not depend on the initial energy at each node.

1.9.3 Comparison of Tie-set Based Optimization and Distributed Linear Programing

In this experiment, the optimality of the Tie-set Based Optimization (TBO) and Distributed Linear Programing (DLP) is assessed. The mechanism of both DLP and TBO are described in Algorithms 2 and 3, respectively. Then, we compare the following factors:

- Changing process of the average battery energy level: $Avg(b_i(t)) = \overline{B}(t)$.
- Amount of total energy generated at CGF: $Total(g_i(t)) = \sum_{i \in \mathscr{V}} g_i(t)$.

Although any optimization technique cannot excel in the solution of the problem, it is important to analyze how the result of those approaches are close to the theoretical limitation in terms of convergence of battery energy levels and the energy supply from CGF. The simulation network is the same as the 100-node network in the previous section. The RPR is intended to be 50% by giving load and renewable generation with the range of 0 to 60 MWh and 0 to 30 MWh, respectively. Initial LELs are $e_i(0) = 20$ MWh. Simulation data of $\sum_{i \in \mathscr{V}} g_i(t)$ is taken at $t = 0, 15, 30, ..., 600$ (min). Experiments are conducted 20 times and the average is calculated.

Figure 1.12 demonstrates the average battery energy level $\overline{B}(t) = Agv(b_i(t))$ within the time from $t = 0$ to 600 min that compares DLP and TBO (Tie-st) with the optimal $\overline{B}^*(t)$. The average battery energy level by TBO is closer to the theoretical solution than DLP since TBO can calculate power flows that balance the battery energy levels within individual tie-sets. DLP requires more energy than TBO almost at each time step since the optimization is based on individual nodes and energy is exchanged only with adjacent nodes, which restricts the scale of optimization, even though the process of DLP is completely decentralized by communicating with other peripheral nodes with limited state information. Therefore, CGF generation at each node is not minimized as it must satisfy $b_i(t) \geq e_i(t)$ to compensate the demand that has not been met by its net power export $F_i(t)$.

Figure 1.13 shows the changing process of $\sum_{i \in \mathscr{V}} g_i(t)$ by DLP and TBO as well as the theoretical solution (Optimal). As indicated in Figure 1.13, the optimal line shows minimum allocation of energy generated at CGFs at each time. The the result by TBO is similar to the theoretical solution as both TBO and optimal line converge between 1400 MWh and 1600 MWh after $t = 60$ min. DLP makes up for the demand not sufficiently satisfied by power flows at each node, which leads to excessive energy supply from CGF. That is why the simulated behavior of the average battery energy level by DLP in Figure 1.12 is constantly higher than that of TBO and theoretical solution.

Table 1.1 shows comparison of the total CGF power provision from $t = 0$ to 600 at all the nodes in $\mathscr{V}$ calculated as $\sum_{t=0}^{600} \sum_{i \in \mathscr{V}} g_i(t)$ with different RPRs. The result

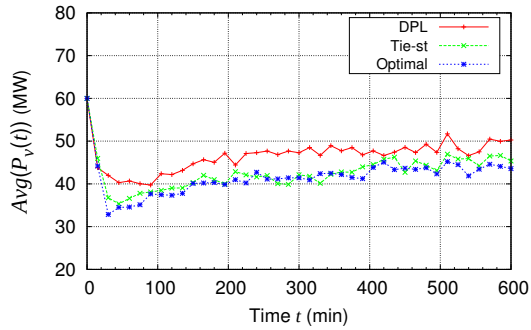


Figure 1.12: Comparison on the average battery energy level $\overline{B}(t) = Agv(b_i(t))$ of Distributed Linear Programing (DLP), Tie-set Based Optimization (TBO), and the optimal $\overline{B}^*(t)$ when RPR is 50%.

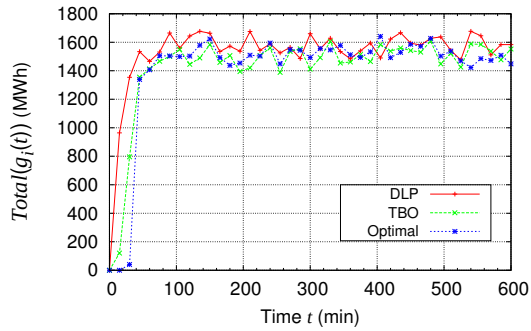


Figure 1.13: Comparison on total CGF energy supply $Total(g_i(t)) = \sum_{i \in \mathcal{V}} g_i(t)$ of Distributed Linear Programing (DLP), Tie-set Based Optimization (TBO), and the optimal supply from CGFs when RPR is 50%.

indicates that the total CGF supply by TBO is close to the theoretical solution with modest increase, whereas the amount of CGF supply by DLP is larger than TBO and the optimal amount. Therefore, we could reduce the cost of using fossil fuels and nuclear plants by fully taking advantage of renewable generation on the basis of the nature of iterative TBO where the optimization in a system of μ independent tie-sets quickly leads to global optimization [32, 41].

1.10 Summary

This chapter has discussed a decentralized model for maximizing renewable integration in future grid and described how to realize real-time energy sharing and power production response by incorporating batteries and many end-use smart devices connected to a local network or through the Internet. The uncontrollability of loads and

Table 1.1: Comparison of Total CGF Energy Supply $\sum_{t=0}^{600}\sum_{i\in\mathcal{V}} g_i(t)$ *(MWh) with Different Renewable Penetration Rates*

RPR	25%	50%	75%	100%
DLP	86708.70	62315.36	38134.25	28874.94
Tie-set	85637.09	57945.28	36460.37	25686.16
Optimal	85007.26	57535.15	34474.77	25302.25

renewables is dealt with by the real-time decentralized algorithms such as distributed nodal approach with distributed linear programing and clustering approach with tie-set based optimization. The simulation results show that balanced energy sharing of renewable energies constantly conducted with decentralized control models realize a sustainable power network complimented by dynamic power supplies from centralized generation facilities even if the load process and renewable generation is highly variable and unpredictable. The discussed models extract essential aspects of a prospective power grid so that various optimization theories and algorithms being studied in the boundary domain between power systems and distributed network systems can be integrated into the future grid. A variety of latest technologies and forecasting techniques on load and renewable generation can also be utilized for both planning and real-time control phases in order to create a reliable, resilient, and sustainable future grid.

Bibliography

[1] C. Batlle, A. Dòria-Cerezo, and E. Fossas, "Bidirectional power flow control of a power converter using passive hamiltonian techniques," *Int. J. Circuit Theory Appl.*, vol. 36, pp. 769–788, Oct. 2008.

[2] T. Takuno, M. Koyama, and T. Hikihara, "In-home power distribution systems by circuit switching and power packet dispatching," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 427–430, 2010.

[3] K. Tashiro, R. Takahashi, and T. Hikihara, "Feasibility of power packet dispatching at in-home dc distribution network," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 401–405, 2012.

[4] N. Li, L. Chen, and S. Low, "Optimal demand response based on utility maximization in power networks," in *2011 IEEE Power and Energy Society General Meeting*, pp. 1–8, 2011.

[5] T. Yau, L. N. Walker, H. Graham, A. Gupta, and R. Raithel, "Effects of battery storage devices on power system dispatch," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 1, pp. 375–383, 1981.

[6] E. Sortomme and M. El-Sharkawi, "Optimal power flow for a system of microgrids with controllable loads and battery storage," in *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pp. 1–5, 2009.

[7] C. Hill, M. Such, D. Chen, J. Gonzalez, and W. Grady, "Battery energy storage for enabling integration of distributed solar power generation," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 850–857, 2012.

[8] P. Yang, P. Chavali, and A. Nehorai, "Parallel autonomous optimization of demand response with renewable distributed generators," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 55–60, 2012.

[9] Y. Zhang, N. Gatsis, and G. Giannakis, "Robust distributed energy management for microgrids with renewables," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 510–515, 2012.

[10] D. Olivares, C. Canizares, and M. Kazerani, "A centralized optimal energy management system for microgrids," in *IEEE Power and Energy Society General Meeting*, pp. 1–6, July 2011.

[11] A. Tsikalakis and N. Hatziargyriou, "Centralized control for optimizing microgrids operation," in *IEEE Power and Energy Society General Meeting*, pp. 1–8, July 2011.

[12] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755 – 768, 2012. Special Section: Energy efficiency in large-scale distributed systems.

[13] J. POPEANGĂ, "Cloud computing and smart grids," *ERP and E-Business Application Deployment in Open Source Distributed Cloud Systems*, p. 57, 2012.

[14] S. Rusitschka, K. Eger, and C. Gerdes, "Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 483–488, Oct 2010.

[15] F. P. Miller, A. F. Vandome, and J. McBrewster, *Amazon Web Services*. Alpha Press, 2010.

[16] R. H. Weber and R. Weber, *Internet of Things*. Springer, 2010.

[17] M. Yun and B. Yuxin, "Research on the architecture and key technology of internet of things (iot) applied on smart grid," in *International Conference on Advances in Energy Engineering (ICAEE)*, pp. 69–72, June 2010.

[18] S. Karnouskos, "The cooperative internet of things enabled smart grid," in *IEEE international symposium on consumer electronics (ISCE2010), June*, pp. 07–10, 2010.

[19] C.-H. Lo and N. Ansari, "Decentralized controls and communications for autonomous distribution networks in smart grid," *IEEE Transactions on Smart Grid*, vol. 4, pp. 66–77, March 2013.

[20] T. Suehiro and T. Namerikawa, "Decentralized control of smart grid by using overlapping information," in *Proceedings of SICE Annual Conference (SICE)*, pp. 125–130, Aug 2012.

[21] J. Shah, B. Wollenberg, and N. Mohan, "Decentralized power flow control for a smart micro-grid," in *IEEE Power and Energy Society General Meeting*, pp. 1–6, July 2011.

[22] H. Liu, Z. Hu, Y. Song, and J. Lin, "Decentralized vehicle-to-grid control for primary frequency regulation considering charging demands," *IEEE Transactions on Power Systems*, vol. 28, pp. 3480–3489, Aug 2013.

[23] C. Ahn, C.-T. Li, and H. Peng, "Decentralized charging algorithm for electrified vehicles connected to smart grid," in *American Control Conference (ACC), 2011*, pp. 3924–3929, June 2011.

[24] S. Galli, A. Scaglione, and Z. Wang, "For the grid and through the grid: The role of power line communications in the smart grid," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 998–1027, 2011.

[25] Z. Wang, A. Scaglione, and R. Thomas, "Generating statistically correct random topologies for testing smart grid communication and control networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 28–39, 2010.

[26] P. C. Mosaic, "Mesh networks is communications winner in utility survey," 2009.

[27] R. Rajagopal, E. Bitar, P. Varaiya, and F. Wu, "Risk-limiting dispatch for integrating renewable power," *International Journal of Electrical Power and Energy Systems*, vol. 44, no. 1, pp. 615 – 628, 2013.

[28] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[29] K. Nakayama, N. Shinomiya, and H. Watanabe, "An autonomous distributed control method for link failure based on tie-set graph theory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 11, pp. 2727–2737, 2012.

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

[31] N. Shinomiya, T. Koide, and H. Watanabe, "A theory of tie-set graph and its application to information network management," *International Journal of Circuit Theory and Applications*, vol. 29, no. 4, pp. 367–379, 2001.

[32] 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*, vol. 32, no. 6, pp. 447–470, 2004.

[33] J. 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, pp. 203–211, 2010.

[34] K. Nakayama, K. Benson, L. Bic, and M. Dillencourt, "Complete automation of future grid for optimal real-time distribution of renewables," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 418–423, 2012.

[35] K. Nakayama, K. E. Benson, V. Avagyan, M. B. Dillencourt, L. Bic, and N. Venkatasubramanian, "Tie-set based fault tolerance for autonomous recovery of double-link failures," in *IEEE International Symposium on Computers and Communications (ISCC)*, 2013.

[36] K. Nakayama, N. Shinomiya, and H. Watanabe, "An autonomous distributed control method based on tie-set graph theory in future grid," *International Journal of Circuit Theory and Applications*, vol. 41, no. 11, pp. 1154 – 1174, 2013.

[37] L. Bic, "Distributed computing using autonomous objects," in *Proceedings of the Fifth IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems*, pp. 160–168, 1995.

[38] B. Tarroja, F. Mueller, J. D. Eichman, J. Brouwer, and S. Samuelsen, "Spatial and temporal analysis of electric wind generation intermittency and dynamics," *Renewable Energy*, vol. 36, no. 12, pp. 3424 – 3432, 2011.

[39] "Irvine smart grid demonstrate project," tech. rep., EDISON and EPRI, Nov. 2010, Available Online at http://www.sustainability.uci.edu/Resources1/ISGDOverview.pdf.

[40] J.-P. Maton, L. Zhao, and J. Brouwer, "Dynamic modeling of compressed gas energy storage tocomplement renewable wind power intermittency," *International Journal of Hydrogen Energy*, vol. 38, no. 19, pp. 7867 – 7880, 2013.

[41] Y. Sakai, K. Nakayama, and N. Shinomiya, "A node-weight equalization problem with circuit-based computations," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2525–2528, 2013.