Title
Distributed real-time power flow control with renewable integration

Permalink
https://escholarship.org/uc/item/1cf8h3q3

ISBN
9781479915262

Authors
Nakayama, K
Zhao, C
Bic, LF
et al.

Publication Date
2013-12-01

DOI
10.1109/SmartGridComm.2013.6688010

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed
Distributed Real-Time Power Flow Control with Renewable Integration

Kiyoshi Nakayama*, Changhong Zhao†, Lubomir F. Bic*, Michael B. Dillencourt*, Jack Brouwer‡
†Dept. of Computer Science, University of California, Irvine, CA 92697-3435, USA
‡Dept. of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125-0001, USA
Email: kiyoshi.nakayama@ieee.org, czhao@caltech.edu, {bic, dillenco}@ics.uci.edu, jb@apep.uci.edu

Abstract—We formulate an Optimal Real-Time Power Flow (ORPF) problem that integrates renewable energy generation and energy storage. In the ORPF problem, we seek to minimize the costs of energy storage and of power generation from fossil fuel that are required to balance the loads and generation from renewable sources. We present a novel decentralized algorithm for this problem, using tie-set graph theory. Tie-set graph theory significantly reduces the complexity of the ORPF problem by dividing a power network into a set of independent loops referred to as “tie-sets.” Simulation results demonstrate real-time power production responses and flow controls that lead to reliable use of battery systems and reduce the cost of using fossil fuel.

I. INTRODUCTION

Integration of renewables with energy storage systems has been motivated by the increasing availability of renewable energy from solar and wind power and excess generation by customers. According to [1], “It has been estimated that if 20% of the energy that could be harvested from wind farms across the globe were used, all of the world’s electricity demands could be met several times over [2].” Battery systems provide large savings in generation cost without compromising energy availability for customers [3].

Optimal power flow (OPF) problems, first formulated and studied by Carpentier [4], have been studied for more than half a century. Many of the earlier models for OPF focused on static optimizations, i.e., optimizations for isolated periods of time in which power supply and demand must be balanced at every period. Today, it is increasingly common for energy storage devices such as batteries to be installed and used in power grids [5]. By charging and discharging the batteries, mismatches between instantaneous power supply and demand can be balanced. As a result, the requirement of power balance at every instant in the OPF formulation can be relaxed. Multi-period versions of OPF problems can be considered, in which the battery charging/discharging profile becomes one of the control variables over which the optimization is defined.

In this paper, we adopt a model similar to that of [1]. We formulate an Optimal Real-Time Power Flow (ORPF) problem with energy storage devices. Our goal is to minimize the total cost, which consists of (1) the cost of power production by the Controllable Generation Facility (CGF) using fossil fuels and (2) the cost of using batteries across multiple time periods to balance the fluctuation of renewable power generation and loads. To solve the ORPF problem, we present a novel decentralized algorithm to autonomously allocate the CGF generations and battery charges/discharges.

As the topology of today’s traditional grids are modified to integrate and manage distributed energy resources (DERs), mesh topologies will likely be utilized because of the increased flexibility, efficiency, and resiliency they provide [6], [7]. The proposed algorithm effectively divides a meshed power network into a set of independent loops, so-called “tie-sets” in graph theory, that can be seen as a μ-dimensional linear vector space1. Monitoring of loads and renewable production is locally conducted in each tie-set using an autonomous agent. Based on the monitored data, each tie-set independently optimizes the power production response and then distributes power flow at each time step to minimize both CGF and battery cost.

The work in [8] deals with balanced allocation of static distributed energy resources and introduces the simulation results showing that the iterative optimization within the system of fundamental system of tie-sets leads to global solution whose theoretical property can be verified in [9]. On the basis of [8], our previous work achieves the balanced allocation of renewables with the complete automation of a prospective future grid that integrates end-use devices to capture stochastic process of loads and renewable generations automatically [10]. The proposed algorithms achieve sustainable grid operation even though future load is uncertain and renewable generation is variable and unpredictable. Simulation results show that our algorithm reduces the costs of battery usage and power production by CGF at every time step, and demonstrates flow distribution that leads to reliable use of battery systems.

II. PROBLEM FORMULATION

In this section, we formulate an Optimal Real-Time Power Flow (ORPF) problem that integrates renewable energy production and energy storage devices.

We consider a connected graph $\mathcal{G}=(\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V} = \{1, \ldots, n\}$ representing the buses and a set of links $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ representing the power lines. The links are directed with arbitrarily defined directions, and a link from nodes $i$ to $j$ is denoted interchangeably by either $(i,j) \in \mathcal{E}$ or $i \rightarrow j$.

1 $\mu$ is the nullity of an underlying graph of a power grid as defined in Section III-A.
Let $i : i \to j$ and $k : j \to k$ denote the set of predecessors and successors of node $j$ in the directed graph, respectively.

We focus on the notion of a system operating point (SOP) [11], which has a task of integrating growing amounts of renewables as well as loads and in-and-out power flows into a smart grid. A node in this model has the function of SOP and is also responsible for making a decision to balance the supply and load at every moment, with the help of installed storage devices, e.g., batteries. Storage systems are significantly able to save generation cost without wasting consumers’ facilities, by charging or discharging to balance the surplus or shortfall in generation compared to load. Then, we define the following variables in the power network.

- $p_j(t)$: The battery energy level at node $j \in \mathcal{V}$ at time $t \in \mathcal{T} := \{1, 2, ..., \}$.
- $l_j(t)$: The total load of node $j$ at time $t$.
- $r_j(t)$: The amount of real power injection of node $j$ at time $t$ that is provided by renewable energy resources.
- $g_j(t)$: The amount of real power injection of node $j$ at time $t$ provided from Controller Generation Facility (CGF) that utilizes resources except renewable resources, such as fossil fuel (coal, gas powered) or nuclear.
- $f_{ij}(t)$: The real power flow on the link $(i, j) \in \mathcal{E}$.

We use variables without subscripts to denote vectors with the proper components. For example, $p = (p_j(t), j \in \mathcal{V}$ and $t \in \mathcal{T})$ denotes the vector of node battery energy levels.

The battery energy levels $p_j$ follow the dynamics as

$$p_j(t) = p_j(t - 1) - d_j(t) + g_j(t) - F_j(t), \quad \text{for } j \in \mathcal{V}, \quad (1)$$

where $d_j(t) := l_j(t) - r_j(t)$ is called the net demand, and

$$F_j(t) := \sum_{k : j \to k} f_{jk}(t) - \sum_{i : i \to j} f_{ij}(t) \quad (2)$$

is the net power injection from node $j$ to other nodes at time $t$. Note that the battery energy levels are bounded by finite capacities as

$$0 \leq p_j(t) \leq \bar{p}_j, \quad \text{for } j \in \mathcal{V}. \quad (3)$$

Also, the CGFs are subject to generation capacity constraints

$$0 \leq g_j(t) \leq \bar{g}_j, \quad \text{for } j \in \mathcal{V}, \quad (4)$$

and, due to thermal and safety limits, the power flows on the lines are subject to capacity constraints

$$-\bar{f}_{ij} \leq f_{ij}(t) \leq \bar{f}_{ij}, \quad \text{for } (i, j) \in \mathcal{E}. \quad (5)$$

The generated power injections $g_j(t)$ by CGF incur some cost. We denote the cost of generation $g_j(t)$ at time $t$ by $c_j(g_j(t), t)$. We also consider a cost $b_j(p_j(t))$ for the battery with energy level $p_j(t)$ at time $t$. Then, we formulate the ORPF problem to minimize the cost of CGF generations and efficiently distribute DERs for reliable use of batteries.

**ORPF:** Given initial battery levels $p(0) \geq 0$, net demand $d$, battery capacities $\bar{p}$, CGF capacities $\bar{g}$, link capacities $\bar{f}$, and a given time sequence $\mathcal{T} = \{t\}$, the ORPF is

$$\min \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{V}} (c_i(g_i(t), t) + b_i(p_i(t))) \quad (6)$$

over $j, F, g, p$

s.t. $(1), (2), (3), (4), (5)$. In this paper, we consider quadratic cost functions for CGFs:

$$c_i(g_i(t), t) := \frac{1}{2} \gamma_i(t) g_i^2(t) \quad (7)$$

where $\gamma_i(t)$ is a time-varying coefficient. The battery cost is assumed to be dependent on the battery level $p_i$. As we want to maintain the useful life of a battery, we consider the battery cost function that imposes a penalty proportional to the deviation from its capacity as

$$b_i(p_i(t)) := \alpha_i(\bar{p}_i - p_i(t)) \quad (8)$$

for some $\alpha_i > 0$.

### III. Optimal Real-Time Power Flow Based on Tie-Sets

We introduce tie-set graph theory and an autonomous distributed control model to solve the ORPF problem.

**A. Tie-set Graph Theory**

Tie-set graph theory is described in detail in [12], [13]. Here we provide the basics.

For a given bi-connected graph $G = (V, \mathcal{E})$, let $L_\lambda = \{e_1^\lambda, e_2^\lambda, ...\}$ be a set of all the edges that constitutes a loop in $G$, which is called a tie-set [14]. The set of all vertices included in a tie-set $L_\lambda$ is denoted as $V_\lambda$. Let $T$ and $\bar{T}$ respectively be a spanning tree and a cotree of $G$, where $T = \mathcal{E} - T$. $\mu = \mu(G) = |\bar{T}|$ is called the nullity of a graph. Focusing on a subgraph $G_T = (V, T)$ of $G$ and an edge $l = (a, b) \in T$, there exists only one elementary path $P_T(b, a) \subseteq T$ 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 $L(l) = \{l\} \cup P_T(b, a)$. There are $\mu = |T|$ fundamental tie-sets in $G$, which are called a fundamental system of tie-sets denoted as $L_B = \{L_1, L_2, ..., L_\mu\}$. A fundamental system of tie-sets covers all the vertices and edges even in a non-planar graph $G$ as shown in Fig. 1 where thick lines represent edges of a spanning tree $T$ and thin lines correspond to edges of a cotree $\bar{T}$.

**B. Tie-set based Autonomous Distributed Control**

We describe the model of Tie-set based Autonomous Distributed Control (TADiC) conducted in each $L_\lambda$, which is described in Algorithm 1, to accomplish complete automation of a future smart grid. Algorithm 2 is called by TADiC if $L_\lambda$ gains process priority to control resources within the tie-set. In order to explain TADiC, we use the notations and definitions in Table I.
In a non-planar graph

Tie-set Agent. An autonomous agent that constantly navigates each tie-set based upon the current Measurement Vector (MV) \( y_i(t) \) to the leader node \( v^A_i \). Based on the MV \( y_i(t) \), \( v^A_i \) calculates the value of Tie-set Evaluation Function (TEF) \( \Phi(L_i, t) \) and exchanges it with Adjacent Tie-sets Evaluation Function Message (TEFM) to decide the process priority among tie-sets. If \( L_i \) gains the process priority, \( v^A_i \) sets its Tie-set Flag (TF) as \( \zeta(L_i) = 0 \), a tie-set based upon the current MV \( y_i(t) \) with certain predefined criteria.

Based on the MV \( y_i(t) \), \( v^A_i \) calculates the value of Tie-set Evaluation Function (TEF) \( \Phi(L_i, t) \) with Adjacent Tie-sets Evaluation Function Message (TEFM) to decide the process priority among tie-sets. If \( L_i \) gains the process priority, \( v^A_i \) sets its Tie-set Flag (TF) as \( \zeta(L_i) = 0 \), a tie-set based upon the current MV \( y_i(t) \) with certain predefined criteria.

TIE-SET BASED AUTONOMOUS DISTRIBUTED CONTROL (TADiC)

**INITIALIZE:**
Set TF as \( \zeta(L_i) = 0 \). Call SEND.

**SEND:**
Calculate TEF \( \Phi(L_i, t) \) based upon current MV \( y_i(t) \).
Write the value of \( \Phi(L_i, t) \) into TEFM.
Send TEFM to \( L_i^A \).

**RECEIVE:** (Called when received TEFM)
(Conducted after receiving TEFM from all of \( L_i^A \))
for each \( L_j \in L_i^A \) do
Compare \( \Phi(L_i, t) \) with \( \Phi(L_j, t) \).
end for
If \( \Phi(L_i, t) \) is larger than any other TEFs \( \Phi(L_j, t) \),
then set TF as \( \zeta(L_i) = 1 \); otherwise \( \zeta(L_i) = 0 \).
Call OPTIMIZE.

**OPTIMIZE:**
if \( \zeta(L_i) = 1 \) then
Conduct Algorithm 2.
Set TF as \( \zeta(L_i) = 0 \).
end if
Call NOTIFY.

**NOTIFY:**
Send TFS to \( L_i^A \) to notify that TF \( \zeta(L_i) = 0 \).

**CONFIRM:** (Called when received TFS)
(Conducted after receiving TFS from all of \( L_i^A \) when OPTIMIZE finished.)
Confirm that each TF of \( L_j \in L_i^A \) is \( \zeta(L_j) = 0 \).
Call STAND-BY.

**STAND-BY:**
Stand by for \( \Delta t \).
Call SEND.

**TABLE I**
NOTATIONS AND DEFINITIONS FOR TADiC

<table>
<thead>
<tr>
<th>AdT ( L_i^A )</th>
<th>Adjacent Tie-sets ( L_i^A = { L_j } ). If ( V_L \cap V_j \neq \emptyset ), ( L_j ) is an adjacent tie-set of ( L_i ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV ( y_i(t) )</td>
<td>Measurement Vector. MV ( y_i(t) ) contains various information about a node ( i \in V_L ) at time ( t ) such as loads and renewables.</td>
</tr>
<tr>
<td>TA</td>
<td>Tie-set Agent. An autonomous agent that constantly navigates a tie-set to bring the current MV ( y_i(t) ) with state info of ( L_i ) to its leader node.</td>
</tr>
<tr>
<td>TEF ( \Phi(L_i, t) )</td>
<td>Tie-set Evaluation Function. A function that evaluates the value of TEF ( \Phi(L_i, t) ) with adjacent tie-sets ( L_i^A ).</td>
</tr>
<tr>
<td>TEFM</td>
<td>Tie-set Evaluation Function Message. A message used to exchange the value of TEF ( \Phi(L_i, t) ) with adjacent tie-sets ( L_i^A ).</td>
</tr>
<tr>
<td>TF ( \zeta(L_i) )</td>
<td>Tie-set Flag. When ( \zeta(L_i) = 0 ), a tie-set ( L_i ) is standby; otherwise ( L_i ) is in process ( \zeta(L_i) = 1 ).</td>
</tr>
<tr>
<td>TFS</td>
<td>Tie-set Flag Signal. A signal to notify the state of TF ( \zeta(L_i) ).</td>
</tr>
</tbody>
</table>

Each tie-set \( L_i \in L_n^B \) has a leader node \( v^A_i \). TADiC is conducted in the leader \( v^A_i \) by communicating with leaders of Adjacent Tie-sets (AdT) \( L_i^A \) using a predefined routing table. Tie-set Agent (TA) constantly navigates each tie-set \( L_i \) to bring Measurement Vector (MV) \( y_i(t) \) to the leader node \( v^A_i \). Based on the MV \( y_i(t) \), \( v^A_i \) calculates the value of Tie-set Evaluation Function (TEF) \( \Phi(L_i, t) \) and exchanges it with AdT using Tie-set Evaluation Function Message (TEFM) to decide the process priority among tie-sets. If \( L_i \) gains the process priority, \( v^A_i \) sets its Tie-set Flag (TF) as \( \zeta(L_i) = 0 \), a tie-set based upon the current MV \( y_i(t) \) with certain predefined criteria.

Therefore, the procedure of TADiC starts out with Initialize, and repeats the steps from Send to Receive, Optimize, Notify, Confirm, and StandBy in each leader \( v^A_i \) as in Algorithm 1.

In this paper, TEF is defined as the average value of battery cost functions in a tie-set as follows:

\[
\Phi(L_i, t) = \frac{\sum_{i \in V_L} b_i(p_i(t))}{|V_L|}.
\] (9)

The tie-sets with larger value of \( \Phi(L_i, t) \) gains the process priority to conduct Algorithm 2 in the next section.

C. Decentralized Algorithm for Optimal Real-Time Power Flow

We optimize the objective function (6) in a tie-set \( L_i \) at every time step \( t \). A node power \( p_i(t) \) has two types of variables; the CGF generation \( g_i(t) \) and the net power injection \( F_i(t) \). Our algorithmic solution is designed to first optimize the cost (6) focusing on CGF generations, and then optimize flows in a tie-set as described in Algorithm 2. Necessary information in a tie-set \( L_i \) is provided by MV \( y_i(t) \) included in TA and stored in its leader node \( v^A_i \). After Initialization, \( v^A_i \) conducts the following procedures.

1) **CGF Optimization:** In Step 1, the optimum CGF power \( g_i^*(t) \) at \( i \in V_L \) is first calculated without dealing with the net power injection \( F_i(t) \). Let us define \( p_i(g_i(t), t) \) as

\[
p_i(g_i(t), t) := p_i(t - 1) + d_i(t) + g_i(t), \quad \text{for } i \in V_L \tag{10}
\]

where \( V_L \) is a set of all the nodes in a tie-set \( L_i \). Solving for \( g_i(t) \), we get:

\[
g_i(t) = p_i(g_i(t), t) - p_i(t - 1) + d_i(t).
\]

Now we have some constraints on \( g_i(t) \) and \( p_i(g_i(t), t) \) as in (3) and (4). On the one hand, since \( p_i(g_i(t), t) \leq \overline{p}_i \) and \( g_i(t) \leq \overline{g}_i \), we have

\[
g_i(t) \leq \min \{ \overline{p}_i - p_i(t - 1) + d_i(t), \ overline{g}_i \} \tag{11}
\]

On the other hand, we know that \( g_i(t) \geq 0 \) and \( p_i(g_i(t), t) \geq 0 \). Therefore, we have

\[
g_i(t) \geq \max \{ d_i(t) - p_i(t - 1), 0 \} \tag{12}
\]
Let us define $\psi(g_i(t), t)$ on the basis of (6) as

$$\psi(g_i(t), t) = c_i(g_i(t), t) + b_i(p_i(g_i(t), t)) = \frac{1}{2} \gamma_i(t) g_i^2(t) + \cdots$$

The CGF optimization problem is now to find the value of $g_i^*(t)$ that minimizes $\psi(g_i(t), t)$ at each node $i \in V_\lambda$ subject to the above constraints (11) and (12). We take a derivative of $\psi$ to the above constraints (11) and (12). We take a derivative of $\psi$ with respect to $g_i(t)$:

$$\frac{\partial \psi}{\partial g_i(t)}(g_i(t), t) = 0$$

i.e.,

$$g_i(t) = \frac{\alpha_i}{\gamma_i(t)}$$

Let us define $g_{\min} = \max \{d_i(t) - p_i(t-1), 0\}$ and $g_{\max} = \min \{p_i - p_i(t-1) + d_i(t), \bar{g}_i\}$, and then the optimal CGF $g_i^*(t)$ for each $i \in V_\lambda$ at $t$ is:

$$g_i^*(t) = \begin{cases} \frac{\alpha_i}{\gamma_i(t)} & \text{if } g_{\min} \leq \frac{\alpha_i}{\gamma_i(t)} \leq g_{\max} \\ g_{\min} & \text{if } \frac{\alpha_i}{\gamma_i(t)} < g_{\min} \\ g_{\max} & \text{if } \frac{\alpha_i}{\gamma_i(t)} > g_{\max} \end{cases}$$

2) Flow Optimization: After deciding CGF variables $g_j(t)$ in $V_\lambda$, the flow values $f$ on $L_\lambda$ are optimized in Step 2 considering the battery cost function with $g_j^*(t)$ decided above. Let $\phi_\lambda$ be

$$\phi_\lambda = \sum_{j \in V_\lambda} \alpha_j \left( \bar{p}_j - (p_j(g_j^*(t), t) - F_j(t)) \right)$$

According to (2), $F_j(t)$ at node $j$ in a tie-set $L_\lambda$ is

$$F_j(t) = f_{jk}(t) - f_{ij}(t),$$

where $\{(j, k), (i, j)\} \subseteq L_\lambda$. On the basis of (18), $\sum_{j \in V_\lambda} \alpha_j F_j(t)$ of $\phi_\lambda$ in (17) is

$$\sum_{j \in V_\lambda} \alpha_j F_j(t) = \alpha_1(f_{12}(t) - f_{11}(t)) + \alpha_2(f_{23}(t) - f_{12}(t)) + \cdots$$

$$= (\alpha_1 - \alpha_2)f_{12}(t) + (\alpha_2 - \alpha_3)f_{23}(t) + \cdots + (\alpha_1 - \alpha_{\Gamma})f_{11}(t)$$

where $V_\lambda = \{1, 2, \ldots, \Gamma\}$ and $L_\lambda = \{(1, 2), (2, 3), \ldots, (\Gamma, 1)\}$. Therefore, $\phi_\lambda$ in (17) is transformed into a function of a tie-set flow vector $f$:

$$\phi_\lambda(f) = A^T f + B^T (C - D)$$

where $f = \begin{bmatrix} f_{12}(t) \\ f_{23}(t) \\ \vdots \\ f_{11}(t) \end{bmatrix}$, $A = \begin{bmatrix} \alpha_1 - \alpha_2 \\ \alpha_2 - \alpha_3 \\ \vdots \\ \alpha_1 - \alpha_\Gamma \end{bmatrix}$, $B = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_\Gamma \end{bmatrix}$, $C = \begin{bmatrix} p_1(g_1^*(t), t) \\ p_2(g_2^*(t), t) \\ \vdots \\ p_\Gamma(g_\Gamma^*(t), t) \end{bmatrix}$, and $D = \begin{bmatrix} \bar{p}_1 \\ \bar{p}_2 \\ \vdots \\ \bar{p}_\Gamma \end{bmatrix}$.

Each element of the tie-set flow vector satisfies (5). In addition, $p_j(t)$ satisfies (3). Therefore, we solve the following problem:

$$\min A^T f + B^T (C - D)$$

subject to

$$D - C \leq f' \leq D$$

$$-\bar{f} \leq f \leq \bar{f}$$

As the edges in a tie-set form a linear structure, we can solve (21) by linear programming.

Algorithm 2 Decentralized Algorithm for ORPF

**STEP 0:** Initialization

for each $i \in V_\lambda$ do

\quad $p_i(t-1)$ is preserved in $v_i$ of $L_\lambda$ at $t$.\n\quad $d_i(t) := L_i(t) - r_i(t)$ is provided by TA.\n\quad Set $g_i(t) = 0$.\nend for

**STEP 1:** CGF Optimization

for each $i \in V_\lambda$ do

\quad Calculate $g_i^*(t)$ according to (16).\nend for

**STEP 2:** Flow Optimization

Calculate $f$ according to (21) - (23).
Distribute flows in $L_\lambda$.

IV. SIMULATION AND ANALYSIS

Simulation and experiments were conducted to verify the proposed method for solving the ORPF problem and to analyze the solution and its behavior. The results described below confirm that at every node the useful battery life is maintained by the procedure described for calculating CGF generation and in-and-out power flows. As the network becomes larger, the stability of behavior of overall battery levels does not change, which demonstrates the effectiveness of the proposed method in terms of scalability.

In each run of our simulation, the network is configured to be a biconnected network in which each node has at least 2 link connections. Although links are directed, power flow can pass bidirectionally. Each node has a function that produces its load and renewable power generation over time. In addition, each node $i$ has a battery whose capacity is set as $\bar{p}_i = 20$ MW (Megawatt). The generation capacity is $\bar{g}_i = 100$ MW, and the link capacity is $\bar{f}_{ij} = 100$ MW. The initial powers are uniformly set as $p_i(0) = 5$, $\alpha_i$ of the storage cost $b_i(p_i(t)) = \alpha_i(\bar{p}_i - p_i(t))$ is assigned between 1 to 2 at random; $\gamma_i(t) \equiv 1$. The loads $l_i(t)$ and renewable injections $r_i(t)$ are randomly generated with a time interval of 15 min, where $T = \{0, 15, \ldots\}$. The communications interval $\Delta t$ in TADiC as in Table 1 is set as $\Delta t = 1$ min. MV $y_\lambda(t)$ is constantly reported to the leader of each tie-set by TA.
A. Power Injections at a Node

We first analyze the simulated behavior of various power injections at a node. The simulation network $G = (V, E)$ has 100 nodes and 200 links. Since $G$ has 200 links, the number of tie-sets is $\mu(G) = |E| - |V| + 1 = 101$. The height of the tree is 7. The loads and renewable injections are bounded by the penetration rate of renewables so that the expected net power injection ($F_i(t)$) is intended to be 50%. We show the results at two nodes: an arbitrary node and the root node of the spanning tree in $G$ used to generate the tie-sets.

Fig. 2 shows the behavior of battery level $p_i(t)$, load $l_i(t)$, renewable generation $r_i(t)$, CGF power generation $g_i(t)$, and net power injection $F_i(t)$ at a typical node $i \in V$. The values are shown at 15 minute intervals from $t = 0$ to 600. As shown in Fig. 2, the battery level $p_i(t)$ varies in the range $0 < p_{i0}(t) \leq 20$ (MW). Shortly after the sudden load peaks occur, node $i$ makes up for those peaks with increased CGF generation and increased power inflows. At node $i$, the net power injection $F_i(t)$ can be either positive or negative depending on the power levels and the value of $\alpha_j$ at other nodes $V_{\Lambda}$ within a tie-set. The value of $\alpha_i$ is 1.601.

Fig. 3 shows the behavior at the root node. At this node, $\alpha_i = 1.120$. Since this node is the root of the tree, it belongs to many tie-sets, so it is involved in many optimization procedures conducted in those tie-sets. As can be seen in Fig. 3, the root node produces more CGF power than the node in Fig. 2, with the most of the produced CGF power distributed to other nodes. This suggests that the nodes in a grid can be divided into two categories: nodes that obtain power from other nodes, or the nodes that distribute power to other nodes. This category to which a node $i$ belongs depends in part on the number of tie-sets to which it belongs and the value of $\alpha_i$.

B. Convergence of the Overall Power Levels with Different Renewable Penetrations

We analyze the convergence behavior of the overall power levels, taking their average value to see if the proposed scheme is capable of coping with different values of the renewable penetration rate. We calculate $\overline{P}(t)$, the average value of the power levels at time $t$, as follows:

$$\overline{P}(t) = \frac{\sum_{i \in V} P_i(t)}{|V|}.$$  \hfill (24)

Convergence can be measured by looking at the changing values of $\overline{P}(t)$ over time. Convergence is achieved from time $t_s$ to $t_e$ provided $0.9 \leq \overline{P}(t+1)/\overline{P}(t) \leq 1.1$ is satisfied during the time interval $t_s \leq t \leq t_e$. We define the maximum load $\overline{l}_i$ and the maximum renewable generation $\overline{r}_i$ by

$$0 \leq l_i(t) \leq \overline{l}_i, 0 \leq r_i(t) \leq \overline{r}_i.$$  \hfill (25)

The renewable penetration rate can be adjusted by fixing the load and changing the maximum renewable generation as follows:

- 100%: $\overline{l}_i = 20$ MW, $\overline{r}_i = 20$ MW
- 75%: $\overline{l}_i = 20$ MW, $\overline{r}_i = 15$ MW
- 50%: $\overline{l}_i = 20$ MW, $\overline{r}_i = 10$ MW
- 25%: $\overline{l}_i = 20$ MW, $\overline{r}_i = 5$ MW
Experimental data was obtained for each value of the renewable penetration rate by running the simulation 10 times and averaging the results.

Fig. 4 shows the result of the simulations in a 100-node grid. From the figure, we see that convergence occurs for all values of the renewable penetration rate, and that as the renewable penetration rate increases the convergence line moves closer to the capacity $\bar{P}_t$. For all renewable penetration rates, after 100 minutes the average battery levels are converging in the range between 15 and 20, and they remain in this range once they have reached it.

C. Analysis of CGFs with Different Penetration Rates of Renewables

We analyze the behavior of the total CGF $\sum_{i \in V} g_i(t)$ in a 100-node grid from over a 600-minute period with the same simulation conditions as previously. Fig. 5 shows the result of these simulations. There is an initial spike in the CGF power injection, as power is injected to match the load. After about 30 minutes, the total CGF power becomes stable as $\bar{P}(t)$ starts to converge. As the renewable penetration rate is lowered, the amount of CGF power injection increases.

D. Scalability

To assess scalability, we conducted simulation on networks where the number of nodes varied from 100 to 500 in increments of 100. In these simulations, we held the renewable penetration rate constant at 50%. The results are shown in Fig. 6. The convergence results are similar to those seen in Fig. 4 above, as in all cases convergence occurred after no more than 100 minutes. As can be seen in Fig. 6, as the number of nodes becomes large the final average battery levels approach battery capacity. Although the reason is not clear, this could be because the size of a tie-set (the number of nodes in a tie-set) becomes large and many nodes conduct optimizations in parallel. This behavior suggests that the proposed method will be well suited to future large-scale power networks.

V. CONCLUSION

In this paper, we have formulated an optimal real-time power flow problem to integrate renewable energy generation and energy storage systems. We have proposed a decentralized algorithm to solve the problem, using a fundamental system of tie-sets that divides a grid into a set of independent loops. Our algorithm minimizes the costs of fossil fuel generation and battery storage within each tie-set. Simulation results suggest that reliable operation of batteries may be possible with optimal power productions and flow controls based on the system of tie-sets. The proposed method shows promise as a cost-effective and sustainable grid management strategy that compensates for fluctuations in load and renewable generation in a complex large-scale future power grid.

REFERENCES


