MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Fast Market Clearing Algorithms

Raghunathan, A.U.; Curtis, F.E.; Takaguchi, Y.; Hashimoto, H.

TR2017-196 December 2017

Abstract

Real-time electricity markets are the main transaction platforms for providing necessary balancing services, where the market clearing (nodal or zonal prices depending on markets) is very close to real time operations of power systems. We present single and multiple time period decentralized market clearing models based on the DC power flow model. The electricity market we study consists of a set of Generation Companies (GenCos) and a set of Distribution System Operators (DSOs). The Independent System Operator (ISO) determines the market clearing generation and demand levels by coordinating with the market participants (GenCos and DSOs). We exploit the problem structure to obtain a decomposition of the market-clearing problem where the GenCos and DSOs are decoupled. We propose a novel semismooth Newton algorithm to compute the competitive equilibrium. Numerical experiments demonstrate that the algorithm can obtain several orders of magnitude speedup over a typical subgradient algorithm with no modification to the existing communication protocol between the ISO and market participants.

IMA Workshop Volume

Chapter 1
Fast Market Clearing Algorithms

Arvind U. Raghunathan, Frank E. Curtis, Yusuke Takaguchi and Hiroyuki Hashimoto

Abstract Real-time electricity markets are the main transaction platforms for providing necessary balancing services, where the market clearing (nodal or zonal prices depending on markets) is very close to real time operations of power systems. We present single and multiple time period decentralized market clearing models based on the DC power flow model. The electricity market we study consists of a set of Generation Companies (GenCos) and a set of Distribution System Operators (DSOs). The Independent System Operator (ISO) determines the market clearing generation and demand levels by coordinating with the market participants (GenCos and DSOs). We exploit the problem structure to obtain a decomposition of the market-clearing problem where the GenCos and DSOs are decoupled. We propose a novel semismooth Newton algorithm to compute the competitive equilibrium. Numerical experiments demonstrate that the algorithm can obtain several orders of magnitude speedup over a typical subgradient algorithm with no modification to the existing communication protocol between the ISO and market participants.

A. U. Raghunathan
Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, USA. e-mail: raghunathan@merl.com

F. E. Curtis
Dept. Industrial & Systems Engineering, Lehigh University, Bethlehem, PA 18015, USA. e-mail: frank.e.curtis@gmail.com

Y. Takaguchi and H. Hashimoto
Advanced Technology Center, Mitsubishi Electric Corporation, Hyogo 661-8661 Japan.

1.1 Introduction

Electricity markets are commodity markets where: (i) suppliers (electricity generators) and consumers (electricity customers) are spread across a network and (ii) the flow of the commodity (electricity) is dictated by physical laws [13]. Competition in electricity markets allows to establish a market price that is acceptable to all market participants, whereby the market is said to have reached *equilibrium*. The design of appropriate market or pricing mechanisms is governed by the theory of general equilibria. The non-existence of such an equilibrium implies the possibility of some market participants that can unilaterally affect the prices to their own advantage [28]. For example, competition [4] and active participation (eg. demand response) [27] in electricity markets are known to significantly enhance efficiency and reduce prices. Given the importance of an efficient and reliable grid infrastructure, the modeling and subsequent analysis of electricity markets has seen extensive research. Hobbs and Helman [11] study market equilibria via competitive equilibrium models. Oligopolistic price equilibria were investigated by Hobbs, Metzler and Pang [12] for Direct Current (DC) power flow networks using supply functions. Baldick [2] compares Cournot and supply function equilibrium models of bid-based electricity markets. Day, Hobbs and Pang [7] investigate conjectured supply function models of competition among power generators on a DC power flow network. The impact of network constraints on the market performance is analyzed in [4] under different game theory models. Weber and Overbye [29] study Nash equilibria for electricity markets by employing a full representation of the transmission system. Motto et al. [20] formulate a multi-period electricity auction market tool for a DC power flow network accounting for the transmission congestion, losses and unit commitment constraints as a mixed integer program. A mathematical framework to construct dynamic models for electricity markets and study their competitive equilibria using DC power flow models is provided in Wang et al [28].

A realistic market model is associated with important nonlinearities, arising from nonconvex utility functions and nonlinear network constraints [3]. For instance, the DC power flow model may not be a appropriate when voltage constraints or reactive power constraints are considered. Motto et al. [21] proposed a single time period decentralized electricity market clearing model that includes reactive power and demand responsiveness, based on the Alternating Current (AC) power flow network. More recently, Lavaei and Sojuodi [15] also investigate market efficiency for AC power flow networks by leveraging the zero duality gap of certain OPF formulations [16, 31].

While research has focused largely on aspects of electricity market design, there has been little work on algorithmic and computational aspects. This is especially important in the current context of grid infrastructure modernization and increased penetration of distributed generation. For instance, the DOE agency ARPA-E envisions a future grid infrastructure that incorporates

diverse, distributed generation and storage sources, and that is operated under a distributed control architecture [1]. In that context it is important to develop decentralized or distributed algorithms that scale with network size and have little overhead in communication. This serves as the motivation and focus of this chapter.

1.1.1 Our Focus

We consider a mutiple time period pool-based electricity market consisting of: generation companies (GenCo), load entities called the Distribution System Operators (DSO) and an Independent System Operator (ISO). We assume that: (a) the DC power flow model is used by the ISO to model the power flow in the transmission system, (b) the DSOs are modeled as a single node neglecting the underlying distribution network, (c) the DSOs have the ability to defer loads, and (d) the GenCos and DSOs are price-taking and unwilling to share their cost function to the ISO. Maintaining privacy of the individual market participants motivates the development of a decentralized framework whereby the ISO only transmits price signals to the individual participants and obtains price-sensitive optimal actions from them. Using such information, the ISO could employ a subgradient algorithm to converge to an equilibrium. However, the convergence rate for subgradient algorithms is known to be quite slow [10, 26] and hence, require significant number of message communications with the individual participants. This is undesirable in the current context of rapidly changing grid infrastructure which aims to incorporate intermittent distributed generation and distributed architectures for control and operation [1]. In such a distributed setting, reduction in communication overhead is important. Hence, fast convergence to equilibrium is desirable for robust grid operation.

In this chapter, we exploit the problem structure to obtain decentralized optimization problems in the context of multiple time period market clearing. In such a scheme, the ISO transmits a price signal to the individual participants, who in turn solve their individual optimization problems, the solutions of which are communicated back to the ISO so they may update the price. With this information, we propose that the ISO solves its market clearing problem by solving an implicit complementarity problem (ICP) as introduced in Curtis and Raghunathan [5]. To solve the ICP, Curtis and Raghunathan [5] propose a semismooth Newton algorithm for accelerating convergence when solving structured quadratic programs. We employ the same algorithm for solving the ISO's market clearing problem. The algorithm requires the computation of sensitivity of the market participants's solution to the price. We exploit the underlying communication protocol to additionally compute the sensitivity of their solution to changes in the price. Note that this requires no change in the computations performed by the GenCos and DSOs. We demon-

strate through numerical experiments that our approach leads to orders of magnitude fewer function evaluations as compared to a subgradient method. The authors previously investigated the approach for single time period market clearing in [24] in which the GenCos and DSOs were also assumed to provide the sensitivity information. The approach described in this chapter removes this assumption.

We note that a similar approach has been considered when an AC power flow model is used and only equality constraints are present; see Motto et al [20]. In this work, the authors propose applying dual decomposition to a non-convex nonlinear program for which the guarantees of finding a solution with zero duality gap do not exist. Further, [20] employs a pure Newton strategy which does not have global convergence guarantees [23]. By contrast, in this chapter we consider the simpler DC power flow model which is convex and hence, there exists no duality gap. Further, we allow for inequality constraints and also argue that from a computational standpoint the problem is better posed than the equality constrained formulation. We also present numerical results showing that a pure Newton strategy, such as in [20], is not robust in converging to the solution. Our approach can also be extended to consider AC power flow models as in [20]. In fact, [5] also proposed a semismooth Newton algorithm for solving non-convex structured quadratic programs using semismooth Newton algorithms. The framework of [5] can be extended to the case of AC power flow models and will be investigated in a separate study.

1.1.2 Organization of the Chapter

The rest of the paper is organized as follows. Models of the market participants and the notions of competitive equilibrium are presented in §1.2. An implicit complementarity formulation of the ISO's market-clearing problem is presented in §1.3. We describe the semismooth formulation and algorithm in §1.4. Numerical results demonstrating the efficacy of the method are presented in §1.5 followed by conclusions in §1.6.

1.2 Competitive Equilibrium

In this section, we describe the optimization problems related to each of the market participants: generation companies (GenCos), Distribution System Operators (DSOs), and the Independent System Operator (ISO). Based on these, we present the notion of competitive equilibrium and social welfare maximization. In what follows, $\mathcal{N}$ denotes the set of buses in the transmission network of the ISO while $\mathcal{N}^G$ and $\mathcal{N}^D$ (with $\mathcal{N} = \mathcal{N}^G \cup \mathcal{N}^D$) respectively

denote the nodes connected to GenCos and DSOs. Further, $\mathcal{L}$ denotes the set of lines in the transmission network. We assume there are T time periods and the set $\mathcal{T} = \{1, \ldots, T\}$ represents the set of time periods. The electricity price at a node $i \in \mathcal{N}$, time period $t \in \mathcal{T}$ is denoted by $\lambda_{i,t}$. We use boldface to denote vector quantities: $\boldsymbol{\lambda}_{i,\cdot} = (\lambda_{i,1}, \ldots, \lambda_{i,T}) \in \mathbb{R}^T$ is the vector of prices over all time periods at the node i, $\boldsymbol{\lambda}_{\cdot,t} = (\lambda_{1,t}, \ldots, \lambda_{|\mathcal{N}|,t})$ is the vector of prices over the entire set of nodes in the time period t, and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_{\cdot,1}, \ldots, \boldsymbol{\lambda}_{\cdot,T}) \in \mathbb{R}^{|\mathcal{N}|T}$ is the vector of all nodal prices for all time periods. Note that $(\boldsymbol{\lambda}_{1,\cdot}, \ldots, \boldsymbol{\lambda}_{|\mathcal{N}|,\cdot})$ is a different ordering of the vector $\boldsymbol{\lambda}$. The power injection into the network at the node i at time period t is denoted by $P^*_{i,t}(\boldsymbol{\lambda})$. Similarly, $\boldsymbol{P}^*_{i,\cdot} \in \mathbb{R}^T$ is the vector of power injections at a node i over all time periods and, $\boldsymbol{P}^*_{\cdot,t} \in \mathbb{R}^{|\mathcal{N}|}$ is the vector of all nodal power injections in the time period t.

1.2.1 Generation Company (GenCo)

The generation company located at node $i \in \mathcal{N}^G$ chooses its optimal power generation level $\boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}_{i,\cdot})$ over all time periods given the set of nodal prices over the time periods $\boldsymbol{\lambda}_{i,\cdot}$ from the ISO by solving the optimization problem

$$\boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}_{i,\cdot}) = \arg \min_{P_1,\ldots,P_T} \sum_{t \in \mathcal{T}} (c_i(P_t) - \lambda_{i,t} P_t) \tag{1.1a}$$

$$\text{s.t. } \underline{P}^G_i \leq P_t \leq \overline{P}^G_i, \forall\, t \in \mathcal{T} \tag{1.1b}$$

$$|P_{t+1} - P_t| \leq \Delta \overline{P}^G_i, \forall\, t \in \mathcal{T} \setminus \{T\} \tag{1.1c}$$

where $c_i(\cdot)$ is the cost of generation associated with the GenCo, $\underline{P}^G_i$ and $\overline{P}^G_i$ are respectively the mimimum and maximum generation levels and $\Delta \overline{P}^G_i$ represent the limit on the change in power generation over successive time periods. We assume the following on the cost function of the GenCo, which implies that (1.1) has a unique solution.

Assumption 1 *The function $c_i(\cdot)$ is strictly convex.*

The optimization problem in (1.1) assumes that the cost function is independent of time periods. This is done for sake of simplicity of exposition and is not a restriction of the approach. When considering multi-period operations GenCos schedule to operate additional units of generation which typically incurs a start-up cost. In addition, there are minimum down (up) periods for generation units once they are shut down (started up). Modeling such operations requires the introduction of binary variables which renders the GenCo problem non-convex. However, these non-convexities can be handled by relaxing the binary variables to be continuous and replacing the feasible region

by the convex hull. We do not pursue this further, but refer the interested reader to [6, 25].

1.2.2 Distribution System Operator (DSO)

The DSO located at node $i \in \mathcal{N}^D$ chooses its optimal power consumption level $-\boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}_{i,\cdot})$ over all time periods given the set of nodal prices over the time periods $\boldsymbol{\lambda}_{i,\cdot}$ from the ISO by solving the optimization problem

$$\boldsymbol{P}^*_i(\boldsymbol{\lambda}_{i,\cdot}) = \arg \min_{P_1,\ldots,P_T} \sum_{t\in\mathcal{T}} (-\lambda_{i,t} P_t - u_i(-P_t)) \tag{1.2a}$$

$$\text{s.t. } \underline{P}^D_i \leq -P_t \leq \overline{P}^D_i\,, \tag{1.2b}$$

$$-\sum_{t\in\mathcal{T}} P_t \geq \underline{P}^{D,\text{tot}}_i \tag{1.2c}$$

where $u_i(\cdot)$ is the utility function of the DSO, $\underline{P}^D_i$ and $\overline{P}^D_i$ are minimum and maximum power consumption levels in a time period and $\underline{P}^{D,\text{tot}}_i$ represents a minimum total power consumption over the multiple time periods. Note that $P^*_i(\boldsymbol{\lambda}_{i,\cdot})$ is negative since it represents power being withdrawn from the electrical network. We assume the following on the utility function of the DSO which ensures that (1.2) has a unique solution.

Assumption 2 *The function $u_i(\cdot)$ is strictly concave.*

The optimization problem in (1.2) assumes that the utility function is independent of time periods. This is done for sake of simplicity of exposition and is not a restriction of the approach. Under Assumption 2, DSO's optimization problem (1.2) is strictly convex and hence, has an unique solution.

1.2.3 Independent System Operator (ISO)

The ISO is responsible for maintaining balance between the GenCos and DSOs, and ensuring that the power flows in the network are within certain limits. Given a vector of nodal prices $\boldsymbol{\lambda}$ over all time periods, the ISO chooses the optimal power injection levels by solving the optimization problem

$$\boldsymbol{P}^{\text{ISO}}(\boldsymbol{\lambda}) = \arg \min_{(\boldsymbol{P}_{\cdot,1},\ldots,\boldsymbol{P}_{\cdot,T})} \sum_{t\in\mathcal{T}} \boldsymbol{\lambda}^T_{\cdot,t} \boldsymbol{P}_{\cdot,t} \tag{1.3a}$$

$$\text{s.t. } \mathbf{1}^T \boldsymbol{P}_{\cdot,t} = 0,\ \forall\, t \in \mathcal{T} \tag{1.3b}$$

$$-\overline{\boldsymbol{P}} \leq \boldsymbol{A}\boldsymbol{P}_{\cdot,t} \leq \overline{\boldsymbol{P}},\ \forall\, t \in \mathcal{T} \tag{1.3c}$$

where $\overline{\boldsymbol{P}} \in \mathbb{R}^{|\mathcal{L}|}$ denotes the vector of power limits on the lines in the network, $\mathbf{1} \in \mathbb{R}^{|\mathcal{N}|}$ is a vector of all ones, and $\boldsymbol{A}$ is the matrix of power distribution factors for the ISO's transmission network. The constraint (1.3b) imposes power balance between the GenCos and DSOs at each time period. The DC power flow model appears in (1.3c) through the power distribution factors [30].

1.2.4 Competitive Equilibrium

A pair $(\widehat{\boldsymbol{P}}, \widehat{\boldsymbol{\lambda}})$ is said to achieve *competitive (or Walrasian)* equilibrium for an electricity market if:

(a) $\widehat{\boldsymbol{P}}_{i,\cdot} = \boldsymbol{P}^*_{i,\cdot}(\widehat{\boldsymbol{\lambda}}_{i,\cdot})\ \forall\, i \in \mathcal{N}^G$,
(b) $\widehat{\boldsymbol{P}}_{i,\cdot} = \boldsymbol{P}^*_{i,\cdot}(\widehat{\boldsymbol{\lambda}}_{i,\cdot})\ \forall\, i \in \mathcal{N}^D$, and
(c) $\widehat{\boldsymbol{P}} = \boldsymbol{P}^{\mathrm{ISO}}(\widehat{\boldsymbol{\lambda}})$.

By the well-known first and second fundamental theorems of welfare economics [19], we have the following.

- A competitive equilibrium is Pareto optimal.
- Every Pareto optimal allocation can be decentralized as a competitive equilibrium.

By the second fundamental theorem of welfare economics [19, 28], a competitive equilibrium can be characterized by maximizing social welfare given as

$$\min_{\boldsymbol{P}} \sum_{t\in\mathcal{T}} \left(\sum_{i\in\mathcal{N}^G} c_i(P_{i,t}) - \sum_{i\in\mathcal{N}^D} u_i(-P_{i,t}) \right) \tag{1.4a}$$

$$\text{s.t. } \mathbf{1}^T \boldsymbol{P}_{\cdot,t} = 0,\ \forall\, t \in \mathcal{T} \tag{1.4b}$$

$$-\overline{\boldsymbol{P}} \le \boldsymbol{A}\boldsymbol{P}_{\cdot,t} \le \overline{\boldsymbol{P}},\ \forall\, t \in \mathcal{T} \tag{1.4c}$$

$$\underline{P}^G_i \le P_{i,t} \le \overline{P}^G_i,\ \forall\, i \in \mathcal{N}^G, t \in \mathcal{T} \tag{1.4d}$$

$$|P_{i,t+1} - P_{i,t}| \le \Delta\overline{P}^G_i,\ \forall\, i \in \mathcal{N}^G, t \in \mathcal{T} \setminus \{T\} \tag{1.4e}$$

$$\underline{P}^D_i \le -P_{i,t} \le \overline{P}^D_i,\ \forall\, i \in \mathcal{N}^D, t \in \mathcal{T} \tag{1.4f}$$

$$-\sum_{t\in\mathcal{T}} P_{i,t} \ge \underline{P}^D,\ \forall\, i \in \mathcal{N}^D. \tag{1.4g}$$

Social welfare maximization achieves Pareto optimal allocation only under certain assumptions. Any electricity dispatch and pricing system is Pareto optimal only if prices are "right" and maximizes welfare only if all the important costs and benefits are priced into the system. For instance, it is well known that electric generation shifts some costs to society such that they are not priced in this market. Furthermore, even when prices are right, welfare is

only maximized if willingness to pay is an accurate measure of utility. We do not delve further into these aspects but refer the reader to the texts [13, 19].

The social welfare maximization formulation in (1.4) is a *centralized formulation*. This does not lend itself to preserving privacy of the market participants. However, the formulation in (1.4) serves as the starting point for deriving the decentralized formulation which we do next.

1.3 Decentralized Market Formulation

We develop the decentralized market formulation based on Lagrange dualization of the coupling constraints in (1.4). For ease of presentation, we represent the power balance constraint in (1.4b) as two inequalities

$$-\mathbf{1}^T \boldsymbol{P}_{\cdot,t} \leq 0, \mathbf{1}^T \boldsymbol{P}_{\cdot,t} \leq 0 \; \forall \, t \in \mathcal{T}. \tag{1.4b'}$$

Introducing multipliers $\underline{\xi}_t, \overline{\xi}_t$ for the power balance constraints in (1.4b') and $\underline{\zeta}_{l,t}, \overline{\zeta}_{l,t} \; \forall \, l \in \mathcal{L}$ for the line limit constraints in (1.4c), the partial Lagrangian for (1.4) can be written as

$$\begin{aligned} & L(\boldsymbol{P}, \underline{\boldsymbol{\xi}}, \overline{\boldsymbol{\xi}}, \underline{\boldsymbol{\zeta}}, \overline{\boldsymbol{\zeta}}) \\ = & \sum_{t \in \mathcal{T}} \left(\sum_{i \in \mathcal{N}^G} c_i(P_{i,t}) - \sum_{i \in \mathcal{N}^D} u_i(-P_{i,t}) + (-\underline{\xi}_t + \overline{\xi}_t)(\mathbf{1}^T \boldsymbol{P}_{\cdot,t}) \right) \\ & + \sum_{t \in \mathcal{T}} \left(\underline{\boldsymbol{\zeta}}_t^T \left(-\overline{\boldsymbol{P}} - \boldsymbol{A}\boldsymbol{P}_{\cdot,t} \right) + \overline{\boldsymbol{\zeta}}_t^T \left(\boldsymbol{A}\boldsymbol{P}_{\cdot,t} - \overline{\boldsymbol{P}} \right) \right). \end{aligned} \tag{1.5}$$

The Lagrangian dualization is restricted to the constraints that fall under the purview of the ISO's optimization problem (1.3). Using the partial Lagrangian in (1.5) and duality theory of convex optimization [18] we can decentralize the welfare maximization problem in (1.4) as explained below. Define the Lagrange dual function as

$$\begin{aligned} g(\underline{\boldsymbol{\xi}}, \overline{\boldsymbol{\xi}}, \underline{\boldsymbol{\zeta}}, \overline{\boldsymbol{\zeta}}) = \min_{\boldsymbol{P}} \; & L(\boldsymbol{P}, \underline{\boldsymbol{\xi}}, \overline{\boldsymbol{\xi}}, \underline{\boldsymbol{\zeta}}, \overline{\boldsymbol{\zeta}}) \\ & \text{s.t. (1.4d)} - \text{(1.4g)}. \end{aligned} \tag{1.6}$$

From the definition of the partial Lagrangian in (1.5) it is easy to see that the objective function and constraints in (1.6) are separable by the GenCos and DSOs. Indeed, we can express the dual function as

$$
\begin{aligned}
& L(\boldsymbol{P}, \underline{\boldsymbol{\xi}}, \overline{\boldsymbol{\xi}}, \underline{\boldsymbol{\zeta}}, \overline{\boldsymbol{\zeta}}) \\
&= \sum_{t \in \mathcal{T}} \left(\sum_{i \in \mathcal{N}^G} (c_i(P_{i,t}) - \lambda_{i,t} P_{i,t}) + \sum_{i \in \mathcal{N}^D} (-\lambda_{i,t} P_{i,t} - u_i(-P_{i,t})) \right) \\
&\quad - \sum_{t \in \mathcal{T}} (-\underline{\boldsymbol{\zeta}}_t + \overline{\boldsymbol{\zeta}}_t)^T \overline{\boldsymbol{P}}
\end{aligned} \tag{1.7}
$$

where $\boldsymbol{\lambda}_{\cdot,t}$, the vector of nodal prices at time period t, is defined as

$$
\boldsymbol{\lambda}_{\cdot,t} = (\underline{\xi}_t - \overline{\xi}_t)\mathbf{1} + \boldsymbol{A}^T(\underline{\boldsymbol{\zeta}}_t - \overline{\boldsymbol{\zeta}}_t). \tag{1.8}
$$

With this definition of the vector of nodal prices $\boldsymbol{\lambda}_{\cdot,t}$, the optimization problem in (1.6) is precisely the set of optimization problems for GenCos (1.1) and DSOs (1.2). Thus, the Lagrangian dualization yields a decentralized formulation in which the ISO interacts with GenCos and DSOs through a price signal and, thereby allowing the market participants to maintain the privacy of their optimization data.

To obtain the solution to (1.4) we rely on convex duality [18] which states the equivalence between (1.4) and its dual optimization problem given as

$$
\begin{aligned}
&\max_{\underline{\boldsymbol{\xi}}, \overline{\boldsymbol{\xi}}, \underline{\boldsymbol{\zeta}}, \overline{\boldsymbol{\zeta}}} \; g(\underline{\boldsymbol{\xi}}, \overline{\boldsymbol{\xi}}, \underline{\boldsymbol{\zeta}}, \overline{\boldsymbol{\zeta}}) \\
&\quad \text{s.t. } (\underline{\boldsymbol{\xi}}, \overline{\boldsymbol{\xi}}, \underline{\boldsymbol{\zeta}}, \overline{\boldsymbol{\zeta}}) \geq 0.
\end{aligned} \tag{1.9}
$$

The solution to (1.9) could be obtained through a subgradient algorithm [10, 26] which is stated in Algorithm 1. For ease of presentation we introduce

$$
\begin{gathered}
\boldsymbol{\nu}_t = \begin{pmatrix} \underline{\xi}_t \\ \overline{\xi}_t \\ \underline{\boldsymbol{\zeta}}_t \\ \overline{\boldsymbol{\zeta}}_t \end{pmatrix}, \boldsymbol{F}_t(\boldsymbol{\nu}) = \begin{pmatrix} -\mathbf{1}^T \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}) \\ \mathbf{1}^T \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}) \\ \boldsymbol{A}\boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}) + \overline{\boldsymbol{P}} \\ -\boldsymbol{A}\boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}) + \overline{\boldsymbol{P}} \end{pmatrix} \; \forall\, t \in \mathcal{T} \\
\boldsymbol{\nu} = (\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_T), \boldsymbol{F}(\boldsymbol{\nu}) = (\boldsymbol{F}_1(\boldsymbol{\nu}), \ldots, \boldsymbol{F}_T(\boldsymbol{\nu}))
\end{gathered} \tag{1.10}
$$

where $\boldsymbol{F}$ denotes the vector of dualized constraints and $\boldsymbol{\nu}$ the corresponding multipliers. Note that $\boldsymbol{F}_t$ is denoted as a function of $\boldsymbol{\nu}$, multipliers over all time periods, since the optimization problem for GenCos (1.1) and DSOs (1.2) are coupled across time periods. The update step for the multipliers in Algorithm 1 only requires access to the optimal solution of the GenCos and DSOs. Thus, they are quite simple to implement and fit the decentralized framework very well. The typical number of iterations required for convergence of the algorithm to a solution that is within ϵ of the optimal solution is $O(\frac{1}{\epsilon^2})$. Thus, a large number of communication rounds are required between the ISO and the market participants (GenCos and DSOs) to achieve convergence. This renders the algorithm quite slow in practice.

Algorithm 1: Subgradient Algorithm

Let $\epsilon \in (0,1)$ be a desired convergence tolerance
Choose $\boldsymbol{\nu}_t^0 = (\underline{\xi}_t^0, \overline{\xi}_t^0, \underline{\boldsymbol{\zeta}}_t^0, \overline{\boldsymbol{\zeta}}_t^0)$ for all $t \in \mathcal{T}$ as the initial guess.
Set $k = 0$.
repeat
  Set $\boldsymbol{\lambda}^k$ according to (1.8)
  For $i \in \mathcal{N}^G$, solve (1.1) with $\boldsymbol{\lambda}_{i,\cdot} = \boldsymbol{\lambda}_{i,\cdot}^k$
  For $i \in \mathcal{N}^D$, solve (1.2) with $\boldsymbol{\lambda}_{i,\cdot} = \boldsymbol{\lambda}_{i,\cdot}^k$
  Choose γ^k (typically $\gamma^k = \frac{a}{k+1}$ for some some $a > 0$)
  Set $\boldsymbol{\nu}^{k+1} = \max\left(0, \boldsymbol{\nu}^k - \gamma^k \boldsymbol{F}(\boldsymbol{\nu}^k)\right)$
  Set $e^{k+1} = \| \min(\boldsymbol{\nu}^k, \boldsymbol{F}(\boldsymbol{\nu}^k))\|_\infty$
  Set $k = k + 1$
until $e^k \leq \epsilon$

1.4 Semismooth Equation Approach

We describe the semismooth equation approach of Curtis and Raghunathan [5] for computing the competitive equilibrium. The optimality conditions [18] for the ISO's problem for all $t \in \mathcal{T}$ are

$$\boldsymbol{\lambda}_{\cdot,t} = (\underline{\xi}_t - \overline{\xi}_t)\mathbf{1} + \boldsymbol{A}^T(\underline{\boldsymbol{\zeta}}_t - \overline{\boldsymbol{\zeta}}_t) \tag{1.11a}$$

$$0 \leq \underline{\xi}_t \perp (\mathbf{1}^T \boldsymbol{P}_{\cdot,t}) \geq 0 \tag{1.11b}$$

$$0 \leq \overline{\xi}_t \perp (-\mathbf{1}^T \boldsymbol{P}_{\cdot,t}) \geq 0 \tag{1.11c}$$

$$0 \leq \underline{\boldsymbol{\zeta}}_t \perp (\boldsymbol{A}\boldsymbol{P}_{\cdot,t} + \overline{\boldsymbol{P}}) \geq 0 \tag{1.11d}$$

$$0 \leq \overline{\boldsymbol{\zeta}}_t \perp (-\boldsymbol{A}\boldsymbol{P}_{\cdot,t} + \overline{\boldsymbol{P}}) \geq 0 \tag{1.11e}$$

where for a pair of vectors $\{a, b\}$ the expression $0 \leq a \perp b \geq 0$ represents the conditions $a_i \geq 0$, $b_i \geq 0$, and $a_i b_i = 0$ for all i. The constraints in (1.11b)–(1.11e) are the so-called *complementarity constraints* [18]. Following the definition in §1.2.4, competitive equilibrium is attained when the conditions in (1.11) hold for $\boldsymbol{P} = \boldsymbol{P}^*(\boldsymbol{\lambda})$. Following [5], we pose the ISO's market-clearing problem as the following *implicit complementarity problem* (ICP)

$$0 \leq \boldsymbol{\nu} \perp \boldsymbol{F}(\boldsymbol{\nu}) \geq 0 \tag{1.12}$$

where $(\boldsymbol{\nu}, \boldsymbol{F}) \in \mathbb{R}^m \times \mathbb{R}^m$ are as defined in (1.10) with $m = (2 + 2|\mathcal{L}|)T$. We call the complementarity problem in (1.12) as *implicit* since $\boldsymbol{P}^*(\boldsymbol{\lambda})$, which appears in computation of $\boldsymbol{F}(\boldsymbol{\nu})$, is obtained by solving a set of optimization problems. Observe that the evaluation of $\boldsymbol{P}^*(\boldsymbol{\lambda})$ only requires communication with the GenCos and DSOs through transmission of the price vector $\boldsymbol{\lambda}$. Thus, the ICP (1.12) has the desired property of decoupling by participants and allows the participants to mainatain *privacy* of their optimization problem.

The following theorem formalizes the equivalence between the ICP (1.12) and the competitive equilibrium.

Theorem 1. *The following are equivalent:*

(a) $(\widehat{\boldsymbol{P}}, \widehat{\boldsymbol{\lambda}})$ *is a competitive equilibrium;*
(b) $\widehat{\boldsymbol{\nu}}$ *solves the ICP* (1.12) *with* $\widehat{\boldsymbol{\lambda}}_{\cdot,t} = (\underline{\widehat{\xi}}_t - \overline{\widehat{\xi}}_t)\mathbf{1} + \boldsymbol{A}^T(\underline{\widehat{\boldsymbol{\zeta}}}_t - \overline{\widehat{\boldsymbol{\zeta}}}_t)$.

Proof. First, we show that (a) implies (b). Suppose (a) holds. From the definition of competitive equilibrium in §1.2.4, $\widehat{\boldsymbol{P}} = \boldsymbol{P}^*(\widehat{\boldsymbol{\lambda}})$. Since $\widehat{\boldsymbol{P}}$ solves the ISO's problem (1.3), we have that there exists multipliers $(\underline{\widehat{\xi}}, \overline{\widehat{\xi}}, \underline{\widehat{\boldsymbol{\zeta}}}, \overline{\widehat{\boldsymbol{\zeta}}})$ satisfying the optimality conditions in (1.11) with $\boldsymbol{P} = \boldsymbol{P}^*(\widehat{\boldsymbol{\lambda}})$. Thus, (b) holds. Now, suppose (b) holds. By the preceding arguments we have that first order stationarity conditions of the ISO's problem (1.3) hold. Since (1.3) is convex, a first order stationary point is also a minimizer [18]. This completes the proof.

To solve the ICP we rewrite the complementarity system using the Fischer operator as

$$\boldsymbol{\Phi}^{\mathrm{FB}}(\boldsymbol{\nu}) = \begin{pmatrix} \phi(\nu_1, F_1(\boldsymbol{\nu})) \\ \vdots \\ \phi(\nu_m, F_m(\boldsymbol{\nu})) \end{pmatrix}, \tag{1.13}$$

where, given scalars a and b, the Fischer-Burmeister function [9] has the form

$$\phi(a, b) = \sqrt{a^2 + b^2} - a - b. \tag{1.14}$$

Clearly, this latter function satisfies

$$\phi(a, b) = 0 \Longleftrightarrow \{a \geq 0,\ b \geq 0,\ \text{and } ab = 0\}. \tag{1.15}$$

The articles [8, 22] discuss regularity properties and sophisticated implementations of semismooth Newton algorithms for complementarity problems using the Fischer-Burmeister function. However, our formulation here is different in the sense that, in our context, the complementarity components $\boldsymbol{\nu}$ and $\boldsymbol{F}(\boldsymbol{\nu})$ are both functions of $\boldsymbol{\nu}$; hence, our formulation is somewhat more straightforward.

At each iteration k of the semismooth Newton algorithm [14] the step $d\boldsymbol{\nu}^k$ is obtained as the solution of

$$\boldsymbol{\Phi}^{\mathrm{FB}}(\boldsymbol{\nu}^k) + H^k d\boldsymbol{\nu}^k = 0, \tag{1.16}$$

where H^k represents the first-order variation of the function $\boldsymbol{\Phi}^{\mathrm{FB}}$ at the point $\boldsymbol{\nu}^k$. We postpone the discussion on the computation of the matrix H^k to §1.4.3 and instead focus on the local convergence properties and algorithmic details. The step $d\boldsymbol{\nu}^k$ obtained by solving (1.16) is called the *Semismooth Newton* step.

1.4.1 Fast Local Convergence

Semismooth functions such as Φ^{FB} are almost everywhere differentiable [14]. Further, at points of non-differentiability, Φ^{FB} is directionally differentiable and can be approached through a sequence of differentiable points. Consequently, for any sequence of directions $d\boldsymbol{\nu} \to 0$ with associated Jacobians $H \in \partial\Phi(\boldsymbol{\nu} + d\boldsymbol{\nu})$ and directional derivatives $(\Phi^{\mathrm{FB}})'(\boldsymbol{\nu}; d\boldsymbol{\nu})$, we have that

$$\|Hd\boldsymbol{\nu} - (\Phi^{\mathrm{FB}})'(\boldsymbol{\nu}; d\boldsymbol{\nu})\| = o(\|d\boldsymbol{\nu}\|). \tag{1.17}$$

This Taylor-series-like property is sufficient to show that iterations defined by (1.16) can converge locally superlinearly.

Theorem 2 ([14]). *Suppose that $\boldsymbol{F}$ is continuously differentiable and $\boldsymbol{\nu}^*$ satisfies $\Phi^{\mathrm{FB}}(\boldsymbol{\nu}^*) = 0$ such that all $H \in \partial\Phi^{\mathrm{FB}}(\boldsymbol{\nu}^*)$ are non-singular. Then, for any $\boldsymbol{\nu}^k$ in a sufficiently small neighborhood of $\boldsymbol{\nu}^*$, it follows that $\|\boldsymbol{\nu}^{k+1} - \boldsymbol{\nu}^*\| \leq C\|\boldsymbol{\nu}^k - \boldsymbol{\nu}^*\|^{1+\gamma}$ for some $C > 0$ and $\gamma > 0$.*

In the present setting, $\boldsymbol{F}$ is not continuously differentiable, only piecewise differentiable (PC^1) since $\boldsymbol{P}^*_{i,\cdot}(\cdot)$ are PC^1 [5]. The main result in [5] proves local superlinear convergence for $\boldsymbol{F} \in PC^1$. Hence, the semismooth Newton iteration [5] converges fast locally, unlike a conventional subgradient approach. We provide numerical evidence for this in §1.5.

1.4.2 Algorithm

To promote global convergence, we employ a line-search based on the merit function $\Psi^{\mathrm{FB}}(\boldsymbol{\nu}) := \|\Phi^{\mathrm{FB}}(\boldsymbol{\nu})\|_2^2$, the 2-norm of the vector $\Phi^{\mathrm{FB}}(\boldsymbol{\nu})$. Observe that the minimum of $\Psi^{\mathrm{FB}}(\boldsymbol{\nu})$ is 0 corresponding to a solution of the ICP (1.12). Thus, reduction of the merit function $\Psi^{\mathrm{FB}}(\boldsymbol{\nu})$ can be used to certify that the steps of the algorithm ultimately decrease the distance to a solution of the ICP. Given a direction $d\boldsymbol{\nu}^k$, the step length α^k is determined as the largest $\alpha^k \in (0, 1]$ such that the sufficient decrease condition

$$\Psi^{\mathrm{FB}}(\boldsymbol{\nu}^k + \alpha^k d\boldsymbol{\nu}^k) \leq \Psi^{\mathrm{FB}}(\boldsymbol{\nu}^k) + \eta\alpha^k \nabla\Psi^{\mathrm{FB}}(\boldsymbol{\nu}^k)^T d\boldsymbol{\nu}^k \tag{1.18}$$

holds where $\eta \in (0, 1)$; e.g., one typically chooses $\eta = 10^{-4}$. The step-length α^k may be obtained using a backtracking line-search starting from 1 and multiplying by a constant factor $\rho \in (0, 1)$ until the sufficient decrease condition holds (1.18). The complete steps of the algorithm are provided in Algorithm 2. At each iteration of the algorithm the ISO computes the price vector $\boldsymbol{\lambda}$ (Step 5) and communicates the nodal price vector $\boldsymbol{\lambda}^k_{i,\cdot}$ to the GenCos and DSOs to obtain their optimal power generation and consumption levels (Steps 6 and 7). The sensitivity of these power levels to the nodal prices are

computed by finite difference in Step 8. To compute the sensitivity of a particular participant $i \in \mathcal{N}$ requires $2 \cdot T$ calls to the participant to solve the respective optimization problems (1.1) or (1.2) for different perturbations of the price vector. This operation can be performed in parallel for each participant $i \in \mathcal{N}$. We emphasize again that the computation of the sensitivity does not require any modification in the optimization problems of the market participants. The computation of the matrix H^k in Step 9 is described in §1.4.3.

Algorithm 2: Semismooth Newton Algorithm

1 Choose a convergence tolerance $\epsilon \in (0, 1)$.
2 Choose an initial guess $\boldsymbol{\nu}_t^0 = (\underline{\xi}_t^0, \overline{\xi}_t^0, \underline{\boldsymbol{\zeta}}_t^0, \overline{\boldsymbol{\zeta}}_t^0)$ for all $t \in \mathcal{T}$. Choose $\{\eta, \rho\} \subset (0, 1)$.
3 Set $k = 0$.
4 **repeat**
5 Set $\boldsymbol{\lambda}^k$ according to (1.8).
6 For $i \in \mathcal{N}^G$, compute $\boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}^k_{i,\cdot})$ from (1.1)
7 For $i \in \mathcal{N}^D$, compute $\boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}^k_{i,\cdot})$ from (1.2).
8 For $i \in \mathcal{N}, t \in \mathcal{T}$ compute $\frac{\partial \boldsymbol{P}^*_{i,\cdot}}{\partial \lambda_{i,t}}$ as

$$\text{Set } \lambda^{\pm}_{i',t'} = \begin{cases} \lambda_{i',t'} & \text{for } i' \neq i, t' \neq t \\ \lambda_{i,t} \pm \delta & \text{for } i' = i, t' = t \end{cases} \text{ for some } \delta > 0.$$

$$\text{Compute } \boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}^+_{i,\cdot}), \boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}^-_{i,\cdot}) \text{ from (1.1) for } i \in \mathcal{N}^G \text{ or (1.2) for } i \in \mathcal{N}^D$$

$$\text{Set } \frac{\partial \boldsymbol{P}^*_{i,\cdot}}{\partial \lambda_{i,t}} = \frac{\boldsymbol{P}^*_{i,\cdot}(\boldsymbol{\lambda}^+_{i,\cdot}) - \boldsymbol{P}^*(\boldsymbol{\lambda}^-_{i,\cdot})}{2\delta}.$$

9 Compute H^k using (1.19) and $d\boldsymbol{\nu}^k$ using (1.16).
10 Find the smallest integer $n \geq 0$ such that (1.18) holds for $\alpha^k = \rho^n$.
11 Set $\boldsymbol{\nu}^{k+1} = \boldsymbol{\nu}^k + \alpha^k d\boldsymbol{\nu}^k$ and $k = k + 1$
12 **until** $\|\Phi^{\mathrm{FB}}(\boldsymbol{\nu}^k)\|_\infty \leq \epsilon$

1.4.3 Computing H^k

The matrix H^k is defined as

$$H^k = D^k_{\boldsymbol{\nu}} + D^k_{\boldsymbol{F}} \nabla_{\boldsymbol{\nu}} \boldsymbol{F}(\boldsymbol{\nu}^k)^T \tag{1.19}$$

where

$$\nabla_{\boldsymbol{\nu}} \boldsymbol{F}(\boldsymbol{\nu}^k) = [\nabla_{\boldsymbol{\nu}} F_1(\boldsymbol{\nu}^k) \cdots \nabla_{\boldsymbol{\nu}} F_m(\boldsymbol{\nu}^k)] \tag{1.20}$$

with $F_j(\cdot)$ in (1.20) denoting the j-th component of $\boldsymbol{F}$, and $\nabla_{\boldsymbol{\nu}} F_j(\cdot)$ is the gradient of $F_j(\cdot)$ with respect to $\boldsymbol{\nu}$. Note that $F_j(\cdot)$ is different from the boldface notation $\boldsymbol{F}_j(\cdot)$ used in (1.10) and is being used to simplify the presentation of the matrices $D^k_{\boldsymbol{\nu}}, D^k_{\boldsymbol{F}}$. Likewise, ν_j represents the j-th component of the m-dimensional vector $\boldsymbol{\nu}$ and is different from the boldface notation $\boldsymbol{\nu}_j$ in (1.10). The matrices $D^k_{\boldsymbol{\nu}}$ and $D^k_{\boldsymbol{F}}$ are diagonal and are defined as described next. Introducing the set $\beta^k = \{j \mid \nu^k_j = 0 = F_j(\boldsymbol{\nu}^k)\}$, the diagonal matrices can be obtained as

$$[D^k_{\boldsymbol{\nu}}]_{jj} = \begin{cases} \left(\dfrac{\nu^k_j}{\|(\nu^k_j, F_j(\boldsymbol{\nu}^k))\|} - 1\right) & \forall\ j \notin \beta^k \\ \left(\dfrac{z_j}{\|(z_j, z^T \nabla F_j(\boldsymbol{\nu}^k))\|} - 1\right) & \forall\ j \in \beta^k \end{cases}$$

$$[D^k_{\boldsymbol{F}}]_{jj} = \begin{cases} \left(\dfrac{F_j(\boldsymbol{\nu}^k)}{\|(\nu^k_j, F_j(\boldsymbol{\nu}^k))\|} - 1\right) & \forall\ j \notin \beta^k \\ \left(\dfrac{z^T \nabla F_j(\boldsymbol{\nu}^k)}{\|(z_j, z^T \nabla F_j(\boldsymbol{\nu}^k))\|} - 1\right) & \forall\ j \in \beta^k \end{cases}$$

where z is chosen such that $z_j = 1$ for $j \in \beta^k$ and 0 otherwise [17].

To present the expression for the matrix $\nabla_{\boldsymbol{\nu}} \boldsymbol{F}(\boldsymbol{\nu}^k)^T$ we recall from (1.8) and (1.10) that the vectors $\boldsymbol{\nu}$ and $\boldsymbol{F}$ have the following structure

$$\begin{aligned} &\boldsymbol{\nu} = (\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_T),\ \boldsymbol{\lambda}_{\cdot,t} = \boldsymbol{B}\boldsymbol{\nu}_t, \\ &\text{and } \boldsymbol{F}(\boldsymbol{\nu}) = (\boldsymbol{F}_1(\boldsymbol{\nu}), \ldots, \boldsymbol{F}_T(\boldsymbol{\nu})),\ \boldsymbol{F}_t(\boldsymbol{\nu}) = \boldsymbol{B}^T \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}) + \mathbf{b}, \\ &\text{where } \boldsymbol{B} = \begin{bmatrix} -\mathbf{1}\ \mathbf{1}\ -\boldsymbol{A}^T\ \boldsymbol{A}^T \end{bmatrix},\ \mathbf{b}^T = \begin{bmatrix} 0\ 0\ \overline{\boldsymbol{P}}^T\ \overline{\boldsymbol{P}}^T \end{bmatrix}. \end{aligned} \tag{1.21}$$

Then the change in the function $\boldsymbol{F}_t(\boldsymbol{\nu}^k)$ due to a perturbation $\Delta\boldsymbol{\nu}$ in the variables $\boldsymbol{\nu}^k$ can be approximated to the first order as

$$\begin{aligned} &\boldsymbol{F}_t(\boldsymbol{\nu}^k + \Delta\boldsymbol{\nu}) - \boldsymbol{F}_t(\boldsymbol{\nu}^k) = \boldsymbol{B}^T(\boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}^k + \Delta\boldsymbol{\lambda}) - \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}^k)) \\ &\approx \boldsymbol{B}^T \left(\sum_{s=1}^{T} \nabla_{\boldsymbol{\lambda}_{\cdot,s}} \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}^k)^T \Delta\boldsymbol{\lambda}_{\cdot,s}\right) = \boldsymbol{B}^T \left(\sum_{s=1}^{T} \frac{\partial \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}^k)}{\partial \boldsymbol{\lambda}_{\cdot,s}} \boldsymbol{B}\Delta\boldsymbol{\nu}_{\cdot,s}\right) \end{aligned}$$

where in the last equality we have used $\Delta\boldsymbol{\lambda}_{\cdot,s} = \boldsymbol{B}\Delta\boldsymbol{\nu}_s$ by (1.21) and $\frac{\partial \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}^k)}{\partial \boldsymbol{\lambda}_{\cdot,s}}$ is a diagonal matrix with $\left[\frac{\partial \boldsymbol{P}^*_{\cdot,t}(\boldsymbol{\lambda}^k)}{\partial \boldsymbol{\lambda}_{\cdot,s}}\right]_{jj} = \frac{\partial P^*_{j,t}(\boldsymbol{\lambda}^k)}{\partial \lambda_{j,s}}$ for $j = 1, \ldots, |\mathcal{N}|$ and is obtained using the computed sensitivities (step 8 in Algorithm 2). Thus, the Jacobian $\nabla_{\boldsymbol{\nu}} F(\boldsymbol{\nu}^k)^T$ can be expressed as

$$\nabla_{\boldsymbol{\nu}} \boldsymbol{F}(\boldsymbol{\nu}^k)^T = \begin{bmatrix} \boldsymbol{B}^T \frac{\partial \boldsymbol{P}^*_{\cdot,1}(\boldsymbol{\lambda}^k)}{\partial \boldsymbol{\lambda}_{\cdot,1}} \boldsymbol{B} & \cdots & \boldsymbol{B}^T \frac{\partial \boldsymbol{P}^*_{\cdot,1}(\boldsymbol{\lambda}^k)}{\partial \boldsymbol{\lambda}_{\cdot,T}} \boldsymbol{B} \\ \vdots & \ddots & \vdots \\ \boldsymbol{B}^T \frac{\partial \boldsymbol{P}^*_{\cdot,T}(\boldsymbol{\lambda}^k)}{\partial \boldsymbol{\lambda}_{\cdot,1}} \boldsymbol{B} & \cdots & \boldsymbol{B}^T \frac{\partial \boldsymbol{P}^*_{\cdot,T}(\boldsymbol{\lambda}^k)}{\partial \boldsymbol{\lambda}_{\cdot,T}} \boldsymbol{B} \end{bmatrix}.$$

The matrix H^k is dense and there is no discernible structure that can be exploited in the solution of the linear system (1.16).

1.5 Numerical Results

We consider IEEE networks for testing the performance of the Algorithms 1 and 2. The algorithms were implemented in MATLAB and executed on a machine with 3.2 GHz Intel Core i7-3930K CPU, 32 GB RAM. Table 1.1 presents information on the number of GenCos, DSOs and lines in the different test cases. The rest of the section is organized as follows: §1.5.1 presents the results for single time period market clearing while §1.5.2 presents the multiple time period market clearing.

Name	$\lvert\mathcal{N}^G\rvert$	$\lvert\mathcal{N}^D\rvert$	$\lvert\mathcal{L}\rvert$
case9	3	3	9
case14	5	11	20
case30	6	20	41
case39	10	21	46
case57	7	42	80
case118	54	99	186
case300	69	191	411

Table 1.1 Problem size information for the test instances.

1.5.1 Single Period Market Clearing

We choose the cost function for the GenCos as a strictly convex quadratic function, $c_i(P) = c_{1i}P + c_{2i}P^2$ where $c_{2i} > 0$. The values for the coefficients c_{1i} and c_{2i} are generated randomly. The utility function for the DSOs is chosen as a strictly concave quadratic function, $u_i(-P) = u_{i1}(-P) + u_{2i}P^2$ where $u_{2i} < 0$. The coefficient values u_{1i} and u_{2i} are generated randomly. The demands at the buses are allowed to vary between 80% and 120% of the nominal demand specified in the test cases available in MATPOWER [32]. Table 1.2 summarizes the performance statistics of Algorithm 2 versus Algorithm 1 (a subgradient algorithm) for a single time period ($T = 1$) market clearing problem. The reported numbers are averaged over 10 different runs

in which the DSO's utility functions and demands are varied. The convergence tolerance for both algorithms was set to $\epsilon = 10^{-6}$. Note that the error measures used for Algorithms 1 and 2 are distinct but equivalent measures of the error in satisfying the ICP (1.12). The subgradient algorithm hits the iteration limit of 100000 on most instances, whereas Algorithm 2 solves the problems in very few iterations with modest function evaluation counts. Further, Algorithm 2 is 2-3 orders of magnitude faster than the subgradient algorithm in terms of CPU time. The number of function evaluations in Table 1.2 also includes those required for the sensitivity matrices $\partial \boldsymbol{P}^*_{\cdot,t'}/\partial \boldsymbol{\lambda}_{\cdot,t}$ in Step 8 of Algorithm 2.

Figure 1.1 plots the typical progress of the error ($\|\Phi^{\mathrm{FB}}(\boldsymbol{\nu}^k)\|_2$) in satisfying ICP (1.12) against the iteration index. The semismooth Newton algorithm dominates the subgradient method for all tolerance levels. Further, the convergence rate is indeed superlinear as predicted by Theorem 2 and is key to explaining the observed acceleration in convergence over the subgradient method.

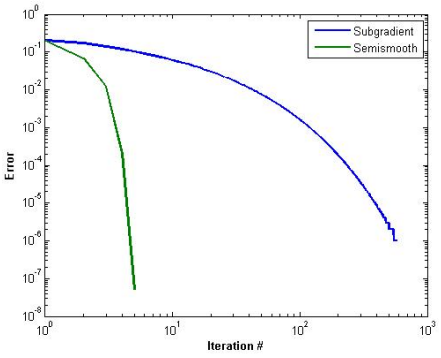

Fig. 1.1 Plot of error against iteration index for the algorithms.

Name	m	Algo. 2 - Semismooth			Algo. 1 - Subgradient	
		Avg. #Iters.	Avg. #Fcn.	Avg. CPU (s)	Avg. #Iters.	Avg. CPU (s)
case9	20	5.4	28.7	0.03	100000	1.8
case14	42	5.7	59.0	0.06	100000	2.1
case30	84	5.2	26.5	0.05	100000	3.1
case39	94	10.0	109.7	0.13	43262	1.6
case57	162	6.8	33.1	0.12	100000	2.7
case118	374	6.2	42.0	0.86	100000	4.5
case300	824	7.2	28.7	4.03	100000	20.1

Table 1.2 Results for the single time period ($T = 1$) market clearing problem using Algorithms. m - size of the vector $\boldsymbol{\nu}$, Avg. #Iters. - average number of iterations, Avg. #Fcn. - Average number of function evaluations, Avg. CPU (s) - average CPU time in seconds.

As mentioned in the introduction, Motto et al [20] had also proposed an approach that is quite similar to the implicit equation approach. The authors employed a pure Newton strategy without any line-search. Table 1.3 presents the results using the pure Newton algorithm of [20]. The algorithm was set a limit of 1000 iterations. Table 1.3 clearly shows that employing the pure Newton strategy is not robust. The algorithm of [20] stops on attaining the iteration limit on all instances of case14, case57, case118 and on 9 of the 10 instances of case30. On the test cases where all the instances were solved - case9, case39 and case300 - the iteration count is comparable to that of the semismooth Newton algorithm proposed in this paper. Thus, it is quite evident that in the single period market clearing the semismooth Newton algorithm (Algorithm 2) based on the implicit complementarity (ICP) formulation (1.12) is computationally efficient and robust in its convergence.

Name	Algorithm in Motto et al. [20]		
	Avg. #Iters.	Avg. #Fcn.	Avg. CPU (s)
case9	5.0	16.0	0.02
case14	1000.0	3000.0	4.71
case30	901.4	2705.2	6.58
case39	8.0	25.0	0.07
case57	1000.0	3000.0	13.91
case118	1000.0	3000.0	83.31
case300	24.1	73.3	13.64

Table 1.3 Results for the single time period ($T = 1$) market clearing problem. Avg. #Iters. - average number of iterations, Avg. #Fcn. - Average number of function evaluations, Avg. CPU (s) - average CPU time in seconds.

1.5.2 Multiperiod Market Clearing

We now explore the computational performance of the multiperiod market clearing problems as the number of time periods is varied. We consider 5 different time periods $T \in \{2, 4, 8, 16, 32\}$. In the multiperiod setting we impose that $\Delta\overline{P}_i^G = 0.25(\overline{P}_i^G - \underline{P}_i^G)$ and $\underline{P}_i^{D,\text{tot}}$ to be the nominal demand specified in the input file multiplied by the number of time periods. Table 1.4 lists the size of the vector of unknowns $\boldsymbol{\nu} \in \mathbb{R}^m$ in the implicit complementarity formulation (1.12) for the different problem instances and time periods. The size of the problem m dictates the number of floating point operations required to solve the linear system in (1.16) in order to compute the Newton step $d\boldsymbol{\nu}^k$ at each iteration of Algorithm 2. Since the matrix H^k is expected to be dense, the number of number floating point operations required scales as m^3 and will be reflected in the computational time of the algorithm. We will highlight this aspect in our discussion on CPU times.

Tables 1.5 and 1.6 list the number of iterations and function evaluations taken by Algorithm 2 on the different instances and time periods. From the tables it is clear that the number of iterations of Algorithm 2 is independent of the increase in the number of time periods. This is a very desirable feature for practical algorithms. However, the number of function evaluations scales linearly with the number of time periods.

Table 1.7 lists the CPU time in seconds taken by the algorithm on the different problem instances and time periods. The reported times include the time performing the step computation in (1.16) and also for the function evaluations. The number reported in the parenthesis is the percentage of time that is spent in computing the sensitivity matrices $\partial \boldsymbol{P}_i^* / \partial \boldsymbol{\lambda}_i$. In our implementation the sensitivity computations for all the participants are performed serially. If these computations are performed in parallel as will be the case in a practical implementation, then the expected speed-ups are reported in Table 1.8. The speed up is computed as

$$\text{speedup} = \frac{\tau_{cpu}}{\tau_{cpu} - \tau_{sen} + \tau_{sen}/|\mathcal{N}|}$$

where τ_{cpu} is the total CPU time taken by Algorithm 2 as reported in Table 1.7 and τ_{sen} is the CPU time spent in sensitivity evaluation. Note that this computation does include the communication overheads that are typically involved in a parallel computing framework. From Table 1.8 it is evident that we can attain almost an order of magnitude speedup up to time periods $T \leq 4$ on the larger instances. However as the number of time periods increases the time involved in the step computation (1.16) dominates the overall CPU time and as a consequence the speedups are not significant.

Name	$T = 2$	$T = 4$	$T = 8$	$T = 16$	$T = 32$
case9	40	80	160	320	640
case14	84	168	336	672	1344
case30	168	336	672	1344	2688
case39	188	376	752	1504	3008
case57	324	648	1296	2592	5184
case118	748	1496	2992	5984	11968
case300	1648	3296	6592	13184	26368

Table 1.4 Summary of the number of constraints in the implicit complementarity problem (ICP) formulation (1.12) for the different instances and time periods.

1.6 Conclusions

In this paper, we have presented a novel semismooth Newton algorithm for multiperiod electricity markets. The approach is decentralized in that it only

Name	$T = 2$	$T = 4$	$T = 8$	$T = 16$	$T = 32$
case9	5	5	5	5	5
case14	6	6	6	5	5
case30	5	5	5	4	4
case39	10	10	10	10	10
case57	4	4	4	4	4
case118	5	5	5	5	5
case300	7	6	6	6	6

Table 1.5 Summary of the iterations taken by the semismooth Newton algorithm (Algorithm 2) to solve the multiperiod market clearing problem.

Name	$T = 2$	$T = 4$	$T = 8$	$T = 16$	$T = 32$
case9	36	76	156	316	636
case14	94	142	238	367	687
case30	41	81	161	258	514
case39	154	234	394	714	1354
case57	29	61	125	253	509
case118	49	89	169	329	649
case300	74	115	211	403	787

Table 1.6 Summary of the function evaluations taken by the semismooth Newton algorithm (Algorithm 2) to solve the multiperiod market clearing problem.

Name	$T = 2$	$T = 4$	$T = 8$	$T = 16$	$T = 32$
case9	0.2 (93.6%)	0.5 (96.9%)	1.0 (98.0%)	2.2 (97.5%)	5.0 (96.9%)
case14	0.7 (86.7%)	1.6 (92.0%)	3.4 (93.8%)	6.3 (93.6%)	14.6 (90.7%)
case30	0.9 (96.3%)	2.1 (96.4%)	4.9 (95.7%)	9.3 (93.8%)	22.9 (86.6%)
case39	2.1 (90.1%)	4.9 (93.2%)	11.3 (92.7%)	26.5 (88.6%)	67.9 (79.2%)
case57	1.4 (97.6%)	3.8 (95.6%)	8.9 (92.8%)	22.0 (85.6%)	70.9 (74.7%)
case118	7.3 (96.0%)	20.8 (93.3%)	49.5 (88.4%)	169.9 (77.4%)	736.4 (64.1%)
case300	36.3 (93.7%)	75.0 (88.2%)	278.2 (78.4%)	1253.0 (64.9%)	9182.8 (37.4%)

Table 1.7 Summary of the CPU time in seconds taken by the semismooth Newton algorithm (Algorithm 2) to solve the multiperiod market clearing problem. The number in the parenthesis is the percentage of time spent in evaluating the sensitivities.

Name	$T = 2$	$T = 4$	$T = 8$	$T = 16$	$T = 32$
case9	4.54	5.19	5.46	5.33	5.19
case14	5.34	7.27	8.29	8.17	6.70
case30	13.48	13.73	12.49	10.21	5.96
case39	7.78	10.19	9.69	7.00	4.29
case57	22.83	15.72	11.02	6.19	3.73
case118	21.73	13.64	8.22	4.32	2.75
case300	15.12	8.24	4.56	2.83	1.59

Table 1.8 Summary of the potential speedup in computations when parallel computations are taken into consideration.

requires the GenCos and DSOs to communicate their optimal response to the price signal from the ISO. The proposed approach is shown to be robust in converging to a tight tolerance of 10^{-6}. For the single period market clearing the proposed algorithm requires about 4 orders of magnitude fewer func-

tion evaluations than a subgradient algorithm. Our numerical experiments demonstrate that the algorithm scales very well with the number of time periods. The communication requirement for the semismooth Newton algorithm (Algorithm 2) is identical to that of that of the subgradient algorithm (Algorithm 1). Hence, the proposed approach can be readily implemented in practice.

There are a number of extensions for this work. We outline some of them below.

- In the current paper, the GenCo problem (1.1) does not include startup or shutdown costs and minimum up or down time for generators. Modeling such operations requires the introduction of binary variables which renders the GenCo problem non-convex. Our algorithm can be easily extended to the GenCo problem resulting from relaxing the binary variables to be continuous and replacing the feasible region by the convex hull [25, 6].
- The current chapter assumes a lumped model for DSOs and no distributed generation. The proposed approach to DSOs where the electrical network of the DSO is also modeled and distributed generation is included. We believe this is a straightforward extension.
- We will also investigate the applicability of the approach when the DSO's power flow is modeled using AC power flow equations. In this context, we will also explore the convex SDP relaxation [15] which has shown to have zero duality gap in a number of instances.

Acknowledgements We are grateful to the referees for a careful reading of the manuscript and bringing to our attention the subtleties of social welfare maximization.

References

1. Advanced Research Project Agency - Energy (ARPA-E). Network Optimized Distributed Energy Systems (NODES). Technical Report DE-FOA-0001289, ARPA-E, 2015.
2. R. Baldick. Electricity market equilibrium models: the effect of parametrization. *IEEE Transactions on Power Systems*, 17(4):1170–1176, 2002.
3. G. Bautista, M. F. Anjos, and A. Vannelli. Formulation of oligopolistic competition in AC power networks: an NLP approach. *IEEE Transactions on Power Systems*, 22(1):105–115, 2007.
4. E. Bompard, Y. C. Ma, R. Napoli, G. Gross, and T. Guler. Comparative analysis of game theory models for assessing the performances of network constrained electricity markets. *IET Generation, Transmission & Distribution*, 4(3):386–399, 2010.
5. F. E. Curtis and A. U. Raghunathan. Solving Nearly-Separable Quadratic Optimization Problems as Nonsmooth Equations. *Computational Optimization and Applications*, 67(2):317–360, 2017.
6. P. Damcı-Kurt, S. Küçükyavuz, D. Rajan, and A. Atamtürk. A polyhedral study of production ramping. *Mathematical Programming*, 158(1-2):175–205, 2016.

7. C. J. Day, B. F. Hobbs, and J. S. Pang. Oligopolistic competition in power networks: a conjectured supply function approach. *IEEE Transactions on Power Systems*, 17(3):597–607, 2002.
8. F. Facchinei, A. Fischer, and C. Kanzow. Regularity properties of a semismooth reformulation of variational inequalities. *SIAM Journal on Optimization*, 8:850–869, 1998.
9. A. Fischer. A special Newton-type optimization method. *Optimization*, 24:269–284, 1992.
10. J.-L. Goffin. On the convergence rate of subgradient optimization methods. *Mathematical Programming*, 13:329–347, 1977.
11. B. F. Hobbs and U. Helman. Complementarity-Based Equilibrium Modeling for Electric Power Markets. In *Modeling Prices in Competitive Electricity Markets*, Wiley Series in Financial Economics, pages 69–98. Wiley, 2004.
12. B. F. Hobbs, C. B. Metzler, and J. S. Pang. Strategic gaming analysis for electric power systems: an MPEC approach. *IEEE Transactions on Power Systems*, 15(2):638–645, 2000.
13. D. S. Kirschen and G. Strbac. *Fundamentals of Power System Economics*. John Wiley & Sons, 2004.
14. L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58:353–368, 1993.
15. J. Lavaei and S. H. Low. Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems*, 27(1):92–107, 2012.
16. J. Lavaei and S. Sojoudi. Competitive Equilibria in Electricity Markets with Nonlinearities. In *Americal Control Conference*, pages 3081–3088, 2012.
17. T. De Luca, F. Facchinei, and C. Kanzow. A semismooth equation approach to the solution of nonlinear complementarity problems. *Mathematical Programming*, 75:407–439, 1996.
18. O.L. Mangasarian. *Nonlinear Programming*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1969.
19. A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1991.
20. A. L. Motto, F. D. Galiana, A. J. Conejo, and J M. Arroyo. Network constrained multiperiod auction for a pool-based electricity market. *IEEE Transactions on Power Systems*, 17(3):646–653, 2002.
21. A. L. Motto, F. D. Galiana, A. J. Conejo, and M. Huneault. On Walrasian equilibrium for pool-based electricity markets. *IEEE Transactions on Power Systems*, 17(3):774–781, 2002.
22. T. S. Munson, F. Facchinei, M. Ferris, A. Fischer, and C. Kanzow. The Semismooth algorithm for large scale complementarity problems. *Journal on Computing*, 13:294–311, 2001.
23. J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer series in operations research and financial engineering. Springer, 1999.
24. A. U. Raghunathan, F. E. Curtis, Y. Takaguchi, and H. Hashimoto. Accelerating Convergence to Competitive Equilibrium in Electricity Markets. In *IEEE Power & Energy Society General Meeting*, pages PESGM2016–000221, 2016.
25. D. Rajan and S. Takriti. Minimum up/down polytopes of the unit commitment problem with start-up costs. Technical Report IBM Research Report RC23628, IBM, Yorktown Heights, NY, 2005.
26. N. Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag, Berlin, 1985.
27. C. L. Su and D. Kirschen. Quantifying the effect of demand response on electricity markets. *IEEE Transactions on Power Systems*, 24(3):1199–1207, 2009.
28. G. Wang, M. Negrete-Pincetic, A. Kowli, E. Shafieepoorfard, S. Meyn, and U. V. Shanbhag. Dynamic competitive equilibria in electricity markets. In Aranya

Chakrabortty and Marija D. Ili, editors, *Control and Optimization Methods for Electric Smart Grids*, volume 3 of *Power Electronics and Power Systems*, pages 35–62. Springer New York, 2012.

29. J. D. Weber and T. J. Overbye. An individual welfare maximization algorithm for electricity markets. *IEEE Transactions on Power Systems*, 17(3):590–596, 2002.
30. A. J. Wood and B. F. Wollenberg. *Power Generation Operation and Control*. Wiley-Interscience, 2nd edition, 1996.
31. B. Zhang and D. Tse. Geometry of the Injection Region of Power Networks. *IEEE Transactions on Power Systems*, 28(2), 2013.
32. R. D. Zimmerman, C. E. Murillo-Snchez, and R. J. Thomas. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.