Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 9, No. 3, pp. 443–461, July 2013

GLOBAL ALGORITHM FOR SOLVING STATIONARY POINTS FOR EQUILIBRIUM PROGRAMS WITH SHARED EQUILIBRIUM CONSTRAINTS*

LEI GUO AND GUI-HUA LIN†

Abstract: This paper focuses on solving various stationarity systems for some kind of equilibrium programs with equilibrium constraints (EPEC). Since the popular stationarity systems for EPECs involve some unknown index sets, we first reformulate these stationary systems as constrained equations and then we propose a globally and superlinearly convergent algorithm to solve these constrained equations. Numerical experience shows that the algorithm performs well.

Key words: *EPEC, EPEC-stationarity, Levenberg-Marquardt method, global algorithm, superlinear convergence*

Mathematics Subject Classification: *90C26, 90C30, 90C33*

---

1 Introduction

Consider the following equilibrium program with shared equilibrium constraints (EPEC) and some shared decision variables: For each $k = 1, \cdots, N$, player k solves the following parametric mathematical program with equilibrium constraints (MPEC) with independent decision variables $x^k \in \Re^{n_k}$ and shared decision variables $y \in \Re^l$:

$$\text{MPEC}(x^{-k}) \left\{ \begin{array}{ll} \min & \theta^k(x^k, y; x^{-k}) \\ \text{s.t.} & g^k(x^k, y; x^{-k}) \le 0,\ h^k(x^k, y; x^{-k}) = 0, \\ & 0 \le G(x^k, y; x^{-k}) \ \perp\ H(x^k, y; x^{-k}) \ge 0, \end{array} \right. \tag{1.1}$$

where $\theta^k : \Re^{n+l} \to \Re$, $g^k : \Re^{n+l} \to \Re^{p_k}$, $h^k : \Re^{n+l} \to \Re^{q_k}$, $G, H : \Re^{n+l} \to \Re^m$ are all twice differentiable functions with respect to (x^k, y) and their second-order derivatives are all locally Lipschitzian, $x^{-k} := (x^{k'})_{k'=1,k'\neq k}^N \in \Re^{n-n_k} (\forall k)$ with $n := \sum_{k=1}^N n_k$, and $a \perp b$ means that the vector a is perpendicular to the vector b. For simplicity, we denote by $p := \sum_{k=1}^N p_k$, $q := \sum_{k=1}^N q_k$, $x := (x^1, \ldots, x^N)$ and, in some cases, in order to emphasize the k-th player's variable within x, we denote by (x^k, x^{-k}) instead of x. This problem extends the classical Nash equilibrium problems by assuming that each player's strategy set may depends on the rival players' strategies and contains a system of common equilibrium constraints. In other words, the EPEC (1.1) is a problem to find equilibria that simultaneously solve several MPECs, each of which is parameterized by other independent decision variables.

*This work was supported in part by NSFC Grant #11071028.
†Corresponding author.

ISSN 1348-9151

The EPEC models play a very important role in many fields such as engineering design, economic equilibria, multi-leader-followers game, etc.; see for instance [3, 9, 11, 16, 18] for applications in studying the strategic behavior of generating firms in deregulated electricity markets and [5, 22, 21] for applications in economics such as forward-spot markets and moral-hazard problems. Optimality conditions for EPECs are studied in [8, 15, 17] and approximation methods are studied in [4, 11, 13, 10, 22], respectively. In particular, Hu [10] presents a diagonalization method that solves sequentially parametric MPECs of each player until an approximation equilibrium point is found. Su [22] proposes a sequential nonlinear equilibrium method that is based on a relaxation technique in [20]. The works [4, 11, 13] consider an approach that reformulates the strong Nash stationarity conditions [11] for EPECs as a mixed equilibrium system.

As is known to us, since MPECs are highly nonconvex, there are several kinds of stationarities defined for MPECs, in which popular stationarities include Clarke stationarity (C-stationarity), Mordukhovich stationarity (M-stationarity), Bouligand stationarity (B-stationarity), and strong stationarity (S-stationarity). Among these stationarities, the S-stationarity is most favorable. However, by the examples given in [14], M-stationary or C-stationary points may be better than S-stationary points in some cases and hence it is necessary to study the M-/C-stationarity systems. The purpose of this paper is to develop effective algorithms for solving various stationarity systems for the EPEC (1.1). As one can see in Section 2, since the C-/M-/S-stationarity systems involve some unknown index sets, we cannot solve these systems directly. Our strategy is similar to the recent work [14]. That is, we reformulate the stationarity systems as constrained equations and then propose a numerical algorithm to solve the equations. The main difference with [14] is that, by making use of a projection operator and an Armijo line search technique, the algorithm presented in this paper is globally and superlinearly convergent, whereas the algorithm given in [14] is locally and superlinearly convergent. Our numerical experience shows that the new algorithm performs quite well.

The following notation will be used later on. We denote by $\|\cdot\|$ the Euclidean vector norm and by $\mathcal{B}_\delta(x) := \{y \in \Re^n \mid \|y-x\| < \delta\}$ the open ball centered at x with radius $\delta > 0$. For a point $x \in \Re^n$ and a closed set $X \subseteq \Re^n$, $\mathrm{dist}(x, X)$ denotes the distance from x to X and $P_X(x) := \{s \in X \mid \|s - x\| = \mathrm{dist}(x, X)\}$ denotes the projection of x onto X. For a mapping $F : \Re^n \to \Re^l$ and a vector $x \in \Re^n$, $\nabla F(x)$ stands for the transposed Jacobian of F at x. In addition, for two vectors x and y, $\min(x, y)$ is taken componentwise.

2 Stationarities and Reformulations

Let $\mathcal{F}^k$ be the feasible region of MPEC(x^{-k}) for each $k \in \{1, \ldots, N\}$ and $\mathcal{F}$ be the feasible region of the EPEC (1.1), i.e.,

$$\mathcal{F} := \{(x, y) \mid (x^k, y) \in \mathcal{F}^k,\ k = 1, \ldots, N\}.$$

To facilitate the notation, we define the following index sets for a given $(x^*, y^*) \in \mathcal{F}$:

$$\begin{aligned}
I_g^{k*} &:= \{\, i \mid g_i^k(x^*, y^*) = 0\}, \quad k = 1, \ldots, N,\\
\mathcal{I}^* &:= \{\, i \mid G_i(x^*, y^*) = 0 < H_i(x^*, y^*)\},\\
\mathcal{J}^* &:= \{\, i \mid G_i(x^*, y^*) = 0 = H_i(x^*, y^*)\},\\
\mathcal{K}^* &:= \{\, i \mid G_i(x^*, y^*) > 0 = H_i(x^*, y^*)\}.
\end{aligned}$$

Obviously, $\{\mathcal{I}^*, \mathcal{J}^*, \mathcal{K}^*\}$ is a partition of $\{1, 2, \ldots, m\}$. For each $k = 1, \ldots, N$, the standard Lagrangian of MPEC(x^{-k}) is defined by

$$\begin{aligned} L^k(x, y, \lambda^k, \mu^k, \alpha^k, \beta^k, \zeta^k) \quad := \quad & \theta^k(x, y) + g^k(x, y)^T \lambda^k + h^k(x, y)^T \mu^k \\ & -G(x, y)^T \alpha^k - H(x, y)^T \beta^k + \zeta^k G(x, y)^T H(x, y) \end{aligned}$$

and the MPEC-Lagrangian of MPEC(x^{-k}) is defined by

$$\mathcal{L}^k(x, y, \lambda^k, \mu^k, u^k, v^k) := \theta^k(x, y) + g^k(x, y)^T \lambda^k + h^k(x, y)^T \mu - G(x, y)^T u^k - H(x, y)^T v^k.$$

Definition 2.1. A strategy $(x^*, y^*) \in \mathcal{F}$ is called a *global (local) equilibrium point* if, for each $k = 1, \ldots, N$, $(x^{*,k}, y^*)$ is a global (local) optimal solution of MPEC($x^{*,-k}$).

Since each MPEC(x^{-k}) is a nonconvex optimization problem, it is generally difficult to get a global equilibrium point. As in standard nonlinear programming theory, we may consider stationarity conditions for EPECs. Based on the MPEC theory, we define the stationarity for the EPEC (1.1) as follows:

Definition 2.2. A strategy $(x^*, y^*) \in \mathcal{F}$ is called a *weakly stationary point* of the EPEC (1.1) if there exist multipliers $(\lambda, \mu, u, v) \in \Re^p \times \Re^q \times \Re^{mN} \times \Re^{mN}$ satisfying

$$k = 1, \ldots, N: \quad \left\{ \begin{array}{l} \nabla_{x^k} \mathcal{L}^k(x^*, y^*, \lambda^k, \mu^k, u^k, v^k) = 0, \\ \nabla_y \mathcal{L}^k(x^*, y^*, \lambda^k, \mu^k, u^k, v^k) = 0, \\ \min(\lambda^k, -g^k(x^*, y^*)) = 0, \\ h^k(x^*, y^*) = 0, \\ \min(G(x^*, y^*), H(x^*, y^*)) = 0, \\ u_i^k G_i(x^*, y^*) = 0, \qquad i = 1, \ldots, m, \\ v_i^k H_i(x^*, y^*) = 0, \qquad i = 1, \ldots, m. \end{array} \right. \tag{2.1}$$

A strategy $(x^*, y^*) \in \mathcal{F}$ is called a *Clarke stationary point* or C-stationary point of the EPEC (1.1) if there exist multipliers $(\lambda, \mu, u, v) \in \Re^p \times \Re^q \times \Re^{mN} \times \Re^{mN}$ satisfying (2.1) and

$$u_i^k v_i^k \geq 0, \qquad i \in \mathcal{J}^*, \; k = 1, \ldots, N. \tag{2.2}$$

A strategy $(x^*, y^*) \in \mathcal{F}$ is called a *Mordukhovich stationary point* or M-stationary point of the EPEC (1.1) if there exist multipliers $(\lambda, \mu, u, v) \in \Re^p \times \Re^q \times \Re^{mN} \times \Re^{mN}$ satisfying (2.1) and

$$\text{either} \;\; u_i^k v_i^k = 0 \;\; \text{or} \;\; u_i^k > 0, v_i^k > 0, \qquad i \in \mathcal{J}^*, \; k = 1, \ldots, N. \tag{2.3}$$

A strategy $(x^*, y^*) \in \mathcal{F}$ is called a *strongly stationary point* or S-stationary point of the EPEC (1.1) if there exist multipliers $(\lambda, \mu, u, v) \in \Re^p \times \Re^q \times \Re^{mN} \times \Re^{mN}$ satisfying (2.1) and

$$u_i^k \geq 0, \; v_i^k \geq 0, \qquad i \in \mathcal{J}^*, \; k = 1, \ldots, N. \tag{2.4}$$

The relations among the above stationarities can be stated as follows:

S-stationarity $\Longrightarrow$ M-stationarity $\Longrightarrow$ C-stationarity $\Longrightarrow$ weak stationarity .

Although the B-stationarity in sense of [19] is a very good candidate for an optimal solution of EPECs, it is difficult to solve [19]. In fact, the B-stationarity is weaker than the S-stationarity but stronger than the M-stationarity [7]. Thus, we can approximate a B-stationary point by solving the M-stationarity system or S-stationarity system. Therefore, we concentrate on the C-/M-/S-stationarity systems in this paper.

Note that the C-/M-/S-stationarity systems contain an unknown index set $\mathcal{J}^*$. Therefore, we cannot apply the developed algorithms in standard nonlinear programming theory to solve these systems directly. From [14], we can obtain the following result immediately.

Theorem 2.3. *For any $(x^*, y^*) \in \mathcal{F}$, we have the following statements:*

(i) *Conditions (2.1) and (2.2) are equivalent to*

$$\text{(2.1) and } u_i^k v_i^k \geq 0, \quad i = 1, \ldots, m, \;\; k = 1, \ldots, N. \tag{2.5}$$

(ii) *Conditions (2.1) and (2.3) are equivalent to*

$$\text{(2.1) and } u_i^k v_i^k \geq 0, \;\; \max\{u_i^k, v_i^k\} \geq 0, \quad i = 1, \ldots, m, \;\; k = 1, \ldots, N. \tag{2.6}$$

(iii) *Conditions (2.1) and (2.4) are equivalent to*

$$k = 1, \ldots, N: \quad \begin{cases} \nabla_{x^k} L^k(x^*, y^*, \lambda^k, \mu^k, \alpha^k, \beta^k, \zeta^k) = 0, \\ \nabla_y L^k(x^*, y^*, \lambda^k, \mu^k, \alpha^k, \beta^k, \zeta^k) = 0, \\ \min(\lambda^k, -g^k(x^*, y^*)) = 0, \\ h^k(x^*, y^*) = 0, \\ \min(G(x^*, y^*), \alpha^k) = 0, \\ \min(H(x^*, y^*), \beta^k) = 0, \\ G(x^*, y^*)^T H(x^*, y^*) = 0. \end{cases} \tag{2.7}$$

Theorem 2.3 indicates that we can remove the unknown index set $\mathcal{J}^*$ from (2.2)–(2.4) and hence we may develop fast numerical algorithms to solve these stationarity systems directly. In what follows, for simplicity, we denote

$$g(x, y) := \begin{pmatrix} g^1(x, y) \\ \vdots \\ g^N(x, y) \end{pmatrix}, \quad h(x, y) := \begin{pmatrix} h^1(x, y) \\ \vdots \\ h^N(x, y) \end{pmatrix},$$

$$\Pi_G(x, y) := \begin{pmatrix} G(x, y) \\ \vdots \\ G(x, y) \end{pmatrix}, \quad \Pi_H(x, y) := \begin{pmatrix} H(x, y) \\ \vdots \\ H(x, y) \end{pmatrix},$$

$$\mathcal{L}(x, y, \lambda, \mu, u, v) := \begin{pmatrix} \nabla_{x^1} \mathcal{L}^1(x, y, \lambda^1, \mu^1, u^1, v^1) \\ \nabla_y \mathcal{L}^1(x, y, \lambda^1, \mu^1, u^1, v^1) \\ \vdots \\ \nabla_{x^N} \mathcal{L}^N(x, y, \lambda^N, \mu^N, u^N, v^N) \\ \nabla_y \mathcal{L}^N(x, y, \lambda^N, \mu^N, u^N, v^N) \end{pmatrix},$$

$$L(x, y, \lambda, \mu, \alpha, \beta, \zeta) := \begin{pmatrix} \nabla_{x^1} L^1(x, y, \lambda^1, \mu^1, \alpha^1, \beta^1, \zeta^1) \\ \nabla_y L^1(x, y, \lambda^1, \mu^1, \alpha^1, \beta^1, \zeta^1) \\ \vdots \\ \nabla_{x^N} L^N(x, y, \lambda^N, \mu^N, \alpha^N, \beta^N, \zeta^N) \\ \nabla_y L^N(x, y, \lambda^N, \mu^N, \alpha^N, \beta^N, \zeta^N) \end{pmatrix}.$$

Then, the systems (2.5)–(2.7) can be rewritten as the following formulations, where $\circ$ means the Hadamard product:

(a) C-stationarity:

$$\begin{cases} \mathcal{L}(x,y,\lambda,\mu,u,v) = 0, \\ \min(G(x,y), H(x,y)) = 0, \\ \min(\lambda, -g(x,y)) = 0, \ h(x,y) = 0, \\ u \circ \Pi_G(x,y) = 0, \ v \circ \Pi_H(x,y) = 0, \\ u \circ v \geq 0. \end{cases}$$

(b) M-stationarity:

$$\begin{cases} \mathcal{L}(x,y,\lambda,\mu,u,v) = 0, \\ \min(G(x,y), H(x,y)) = 0, \\ \min(\lambda, -g(x,y)) = 0, \ h(x,y) = 0, \\ u \circ \Pi_G(x,y) = 0, \ v \circ \Pi_H(x,y) = 0, \\ u \circ v \geq 0, \ \max(u,v) \geq 0. \end{cases}$$

(c) S-stationarity:

$$\begin{cases} L(x,y,\lambda,\mu,\alpha,\beta,\zeta) = 0, \\ \min(\lambda, -g(x,y)) = 0, \ h(x,y) = 0, \\ \min(\Pi_G(x,y), \alpha) = 0, \ \min(\Pi_H(x,y), \beta) = 0, \\ G(x,y)^T H(x,y) = 0. \end{cases}$$

By introducing some slack and auxiliary variables, the above systems can be reformulated as smooth constrained equations in the form

$$F(w) = 0, \quad w \in W, \tag{2.8}$$

where F is a smooth mapping and the constraint set $W := \{w \in \Re^r \mid w_i \geq 0, i \in \mathcal{I}\}$ with $\mathcal{I}$ to be a fixed index set. Specifically, F and W in (2.8) have the following form, where $\Pi_{z_2} := (z_2, \ldots, z_2)$ and $\Pi_{z_3} := (z_3, \ldots, z_3)$:

(a') C-stationarity:

$$F(x,y,s,z_1,z_2,z_3,\lambda,\mu,u,v) := \begin{pmatrix} \mathcal{L}(x,y,\lambda,\mu,u,v) \\ \lambda^T z_1 \\ z_1 + g(x,y) \\ h(x,y) \\ z_2 - G(x,y) \\ z_3 - H(x,y) \\ z_2^T z_3 \\ u \circ \Pi_{z_2} \\ v \circ \Pi_{z_3} \\ s - u \circ v \end{pmatrix} \tag{2.9}$$

and

$$W := \Big\{(x,y,s,z_1,z_2,z_3,\lambda,\mu,u,v) \,\Big|\, s \geq 0; z_i \geq 0 (1 \leq i \leq 3); \lambda \geq 0\Big\}. \tag{2.10}$$

(b') M-stationarity:

$$F(x,y,s_1,s_2,s_3,s_4,z_1,z_2,z_3,\lambda,\mu,u,v) := \begin{pmatrix} \mathcal{L}(x,y,\lambda,\mu,u,v) \\ \lambda^T z_1 \\ z_1 + g(x,y) \\ h(x,y) \\ z_2 - G(x,y) \\ z_3 - H(x,y) \\ z_2^T z_3 \\ u \circ \Pi_{z_2} \\ v \circ \Pi_{z_3} \\ s_1 - u \circ v \\ s_3^T s_4 \\ s_2 - s_3 - u \\ s_2 - s_4 - v \end{pmatrix} \tag{2.11}$$

and

$$W := \Big\{ (x,y,s_1,s_2,s_3,s_4,z_1,z_2,z_3,\lambda,\mu,u,v) \Big| s_i \geq 0 (1 \leq i \leq 4); z_i \geq 0 (1 \leq i \leq 3); \lambda \geq 0 \Big\}. \tag{2.12}$$

(c') S-stationarity:

$$F(x,y,z_1,z_2,z_3,\lambda,\mu,\alpha,\beta,\zeta) := \begin{pmatrix} L(x,y,\lambda,\mu,\alpha,\beta,\zeta) \\ \lambda^T z_1 \\ z_1 + g(x,y) \\ h(x,y) \\ z_2 - G(x,y) \\ z_3 - H(x,y) \\ z_2^T z_3 \\ \alpha^T \Pi_{z_2} \\ \beta^T \Pi_{z_3} \end{pmatrix} \tag{2.13}$$

and

$$W := \Big\{ (x,y,z_1,z_2,z_3,\lambda,\mu,\alpha,\beta,\zeta) \;\Big|\; z_i \geq 0 (1 \leq i \leq 3); \lambda \geq 0; \alpha \geq 0; \beta \geq 0 \Big\}. \tag{2.14}$$

In the next section, we focus on developing effective algorithms for solving these constrained equations.

3 Algorithms for Constrained Equations

Consider the constrained equation

$$F(w) = 0, \quad w \in W, \tag{3.1}$$

where $F : \Re^r \to \Re^\nu$ is a differentiable function and W is a nonempty closed and convex subset of $\Re^r$. Denote by W^* the solution set of (3.1).

Obviously, solving the constrained equation (3.1) is equivalent to solving the constrained optimization problem

$$\begin{aligned} \min \quad & \theta(w) := \frac{1}{2}\|F(w)\|^2 \\ \text{s.t.} \quad & w \in W. \end{aligned} \tag{3.2}$$

Lin et al. [14] present the following Levenberg-Marquardt method for solving problem (3.1):

LM Algorithm:

Step 1: Choose $w^0 \in W$, $\eta > 0$, and set $k := 0$.

Step 2: If $F(w^k) = 0$, then stop. Otherwise, set $\eta_k := \eta\|F(w^k)\|$ and solve the problem

$$\begin{array}{ll} \min & \theta_k(d) := \frac{1}{2}\|F(w^k) + \nabla F(w^k)^T d\|^2 + \frac{\eta_k}{2}\|d\|^2 \\ \text{s.t.} & w^k + d \in W \end{array} \tag{3.3}$$

to get d^k.

Step 3: If $d^k = 0$, then stop. Otherwise, let $w^{k+1} = w^k + d^k$, $k := k+1$, and go to Step 2.

Note that, since (3.3) is a strongly convex program, the iteration is well-defined. The following assumption is the same as assumed in [14]:

Assumption 3.1. There exist some $w^* \in W^*$ and positive constants $\{c, \delta, \gamma\}$ with $\delta \in (0, 1]$ and $\gamma \in [\frac{1}{2}, 2)$ such that

(A1) both F and ∇F are Lipschitz continuous in $\mathcal{B}_{2\delta}(w^*)$ with Lipschitz constant L;

(A2) there holds

$$c\,\text{dist}^{\,1/\gamma}(w, W^*) \le \|F(w)\|, \quad w \in \mathcal{B}_\delta(w^*) \cap W. \tag{3.4}$$

We make a few comments on Assumption 3.1. Since we assume that the involved functions are twice differentiable and their derivatives are locally Lipshitzian, the assumption (A1) holds immediately. The assumption (A2) is actually a local error bound condition and it generally requires some kind of regularity; see [14] for some sufficient conditions for (A2) to hold.

The main convergence result of the LM algorithm can be stated as follows.

Theorem 3.2 ([14])**.** *Let $\{w^k\}$ be a sequence generated by the LM algorithm. Suppose that Assumption 3.1 holds with $\gamma > \frac{2}{3}$. Then there exist $\delta_0 > 0$ and $\kappa > 0$ such that, if $w^0 \in \mathcal{B}_{\delta_0}(w^*)$, there holds*

$$\text{dist}(w^{k+1}, W^*) \le \kappa\,\text{dist}^{\,\tau}(w^k, W^*) \tag{3.5}$$

for each k, where $\tau := \frac{2\gamma}{2-\gamma} > 1$, and $\{w^k\} \subseteq \mathcal{B}_\delta(w^*)$ converges superlinearly to a solution of (3.1), where δ is given in Assumption 3.1 .

We next employ an Armijo-type line search technique to present a global algorithm for solving (3.1).

Global Algorithm:

Step 0: Choose $w^0 \in W$, $\eta > 0$, $\rho > 0$, $\ell > 1$, $\sigma \in (0, 1)$, $\varrho \in (0, 1)$, $\varsigma \in (0, 1)$, and set $k := 0$.

Step 1: If $F(w^k) = 0$ or $w^k = P_W(w^k - \nabla\theta(w^k))$, then stop. Otherwise, set $\eta_k := \eta\|F(w^k)\|$ and solve (3.3) to get a solution d^k.

Step 2: If $d^k = 0$, then stop.

Step 3: If d^k satisfies

$$\|F(w^k + d^k)\| \leq \varsigma \|F(w^k)\|, \tag{3.6}$$

set $w^{k+1} := w^k + d^k$ and $k := k+1$, go to Step 1.

Step 4: If d^k satisfies

$$\nabla\theta(w^k)^T d^k \leq -\rho \|d^k\|^\ell, \tag{3.7}$$

then compute a stepsize $t_k := \max\{\varrho^i |\ i = 0, 1, 2, \dots\}$ such that

$$\theta(w^k + t_k d^k) \leq \theta(w^k) + \sigma t_k \nabla\theta(w^k)^T d^k, \tag{3.8}$$

set $w^{k+1} := w^k + t_k d^k$ and $k := k+1$, go to Step 1. Otherwise, compute a projected gradient stepsize $t_k := \max\{\varrho^i |\ i = 0, 1, 2, \dots\}$ such that

$$\theta(w^k(t_k)) \leq \theta(w^k) + \sigma \nabla\theta(w^k)^T (w^k(t_k) - w^k), \tag{3.9}$$

where $w^k(t) = P_W(w^k - t\nabla\theta(w^k))$, set $w^{k+1} := w^k(t_k)$ and $k := k+1$, go to Step 1.

We make a few remarks on the global algorithm. First of all, since the solution d^k of (3.3) always exists uniquely and it is always a descent direction of problem (3.2), the Armijo-type stepsize in (3.8) always exists, whereas the projected gradient stepsize in (3.9) always exists if $w^k \neq P_W(w^k - \nabla\theta(w^k))$. Moreover, the acceptability test (3.6) not only gives a chance to accept the full stepsize but plays an important role in the following convergence analysis. In addition, in Step 4, if (3.7) is not satisfied, we switch to the antigradient of the merit function, which ensures that the search direction is sufficiently descendant.

We next show the global convergence of the global algorithm.

Theorem 3.3. *Suppose that $w^0 \in W$ and F is continuously differentiable. Let $\{w^k\}$ be a sequence generated by the global algorithm. Then any accumulation point w^* of $\{w^k\}$ is a stationary point of problem (3.2). Furthermore, if Assumption 3.1 holds at w^* with $\gamma \geq \frac{4}{5}$, then the whole sequence $\{w^k\}$ converges to a solution of (3.1) superlinearly with order no less than $\tau := \frac{2\gamma}{2-\gamma} > 1$.*

Proof. First of all, it is obvious that $\{w^k\} \subseteq W$. Assume without loss of generality that $w^k \to w^*$. If there exists a subsequence $\{w^{k_j}\}$ such that, for each j, w^{k_j} implements the projected gradient step (3.9), we have from [2, Proposition 2.3.3] that w^* is a stationary point of problem (3.2). Therefore, without loss of generality, we may assume that the global algorithm does not implement the projected gradient step (3.9).

It is obvious that $\{\theta(w^k)\}$ is monotonically decreasing and bounded below. Assume that $\theta(w^k) \to a$. If $a = 0$, then $\theta(w^*) = 0$ by the continuity of θ. If $a > 0$, it means that (3.6) holds for only finitely many times. Without loss of generality, we assume that the global algorithm implements the Armijo-type line search (3.8) for all iterations. It follows from (3.8) that

$$0 \leftarrow \theta(w^{k+1}) - \theta(w^k) \leq \sigma t_k \nabla\theta(w^k)^T d^k,$$

from which we have $t_k \nabla\theta(w^k)^T d^k \to 0$. We consider the following two cases.

(a) The sequence $\{t_k\}$ is bounded away from 0. It follows from Step 4 of the global algorithm that

$$\|d^k\| \le \Big(-\frac{1}{\rho}\nabla\theta(w^k)^T d^k\Big)^{\frac{1}{\ell}} \to 0, \quad k \to \infty.$$

(b) There exists a subsequence $\{t_{k_j}\}$ converging to 0. According to the acceptance rule of stepsize in (3.8), we have

$$\frac{\theta(w^{k_j} + \varrho^{-1}t_{k_j}d^{k_j}) - \theta(w^{k_j})}{\varrho^{-1}t_{k_j}} > \sigma\nabla\theta(w^{k_j})^T d^{k_j}, \quad \forall j. \tag{3.10}$$

Since $\{d^{k_j}\}$ is bounded by $\ell > 1$, we may assume that $d^{k_j} \to d^*$ as $j \to \infty$. Taking a limit in (3.10), we have from the continuous differentiability of θ that

$$\nabla\theta(w^*)^T d^* \ge \sigma\nabla\theta(w^*)^T d^*,$$

which implies $\nabla\theta(w^*)^T d^* \ge 0$ by $0 < \sigma < 1$. On the other hand, if $d^* \ne 0$, it follows from Step 4 of the global algorithm that $\nabla\theta(w^*)^T d^* < 0$, which gives a contradiction. Thus, we have $d^k \to d^* = 0$.

In consequence, we have from (a) and (b) that $d^k \to 0$. Since d^k is the unique solution of (3.3), we have

$$(\nabla F(w^k)(F(w^k) + \nabla F(w^k)^T d^k) + \eta_k d^k)^T(w - w^k - d^k) \ge 0, \quad w \in W.$$

Taking a limit, we get

$$(\nabla F(w^*)F(w^*))^T(w - w^*) \ge 0, \quad w \in W,$$

which implies that w^* is a stationary point of problem (3.2).

We next show the second part of the theorem. To this end, we first show that (3.6) holds for each k sufficiently large. Note that Assumption 3.1 implies $w^* \in W^*$. Let $\{\delta_0, \kappa, \tau\}$ be the constants given in Theorem 3.2. Since $\{w^k\}$ converges to w^* and $\tau\gamma > 1$ by $\gamma \ge \frac{4}{5}$, there exists $w^{\bar{k}} \in \mathcal{B}_{\delta_0}(w^*)$ such that

$$\kappa L c^{-\tau\gamma}\|F(w^{\bar{k}})\|^{\tau\gamma-1} \le \varsigma.$$

Let $b^0 := w^{\bar{k}}$ and $b^{k+1} := b^k + d^k$, $k = 0, 1, 2, \ldots$. It follows from Theorem 3.2 that $b^k \in \mathcal{B}_\delta(w^*)$ and hence, for each k,

$$\mathrm{dist}(b^{k+1}, W^*) \le \kappa\,\mathrm{dist}^{\,\tau}(b^k, W^*). \tag{3.11}$$

It is easy to see that $\hat{b^k} \in \mathcal{B}_{2\delta}(w^*)$, where $\hat{b^k}$ stands for some element in $P_{W^*}(b^k)$. It follows from Assumption 3.1 and (3.11) that, for each k,

$$\begin{aligned}\frac{\|F(b^{k+1})\|}{\|F(b^k)\|} &\le \frac{L\|b^{k+1} - \widehat{b^{k+1}}\|}{c\|b^k - \hat{b^k}\|^{1/\gamma}} \\ &\le \kappa L c^{-1}\|b^k - \hat{b^k}\|^{\tau - 1/\gamma} \\ &\le \kappa L c^{-\tau\gamma}\|F(b^k)\|^{\tau\gamma-1}.\end{aligned}$$

It is not difficult to see that, for each k,

$$\|F(b^{k+1})\| \le \varsigma \|F(b^k)\|.$$

From the definition of $\{b^k\}$ and the mathematical induction, we have

$$\|F(w^k + d^k)\| \le \varsigma \|F(w^k)\|, \quad k \ge \bar{k},$$

which implies that the full stepsize is accepted in the global algorithm. Therefore, the global algorithm becomes the LM algorithm when k is sufficiently large. As a result, we get the desired results from Theorem 3.2 immediately. □

We will test the global algorithm on (2.8) with the mappings F defined by (2.9), (2.11), (2.13) and the constraint set W defined by (2.10), (2.12), (2.14) respectively.

4 Numerical Results

Now we report our numerical experience with some examples. First of all, we emphasize that, since the EPEC (1.1) reduces to an MPEC when $N = 1$, the global algorithm can be regarded as a generalization of the LM algorithm proposed for solving MPECs. In order to compare the behavior of these two algorithms, we selected 48 MPEC examples from [6, 12]. We further tested the global algorithm on several EPECs selected from [13, 23, 24].

In our tests, the parameters are set by

$$\eta = 10^{-4},\ \rho = 10^{-8},\ p = 2.0,\ \sigma = 10^{-4},\ \varrho = 0.9,\ \varsigma = 0.99995,$$

respectively and we terminated the iterations if one of the following conditions were satisfied:

- $k \ge 100$;
- $\min\{t_k, \|d^k\|\} \le 10^{-12}$;
- $\min\{\|\Phi(w^k)\|, \|w^k - P_W(w^k - \nabla\Psi(w^k))\|\} \le 10^{-6}$.

In addition, we chose all starting points to be $(10, 10, \dots, 10)$ with suitable dimensions.

The computational results for MPECs are reported in Tables 1–4. In the tables, *Iter* denotes the number of iterations by the global algorithm or the LM algorithm, *ErrObjective* denotes the error between the value of the objective function at the approximation solution and the real optimal value, and *ResEquation* denotes the residual of the constrained equations at the approximation solution. From Tables 1–4, we can obtain the following observation:

- There are 14 (30/29) test problems whose real optimal solutions were able to be obtained by solving the S-stationarity (M-/C-stationarity) systems only by the global algorithm, whereas the numbers for the LM algorithm are 14/29/29 respectively.

Based on the above observation, we may have the following conclusions:

Here we mean that the values of |ErrObjective| and $\min\{\text{ResEquation}, \|w^k - P_W(w^k - \nabla\Psi(w^k))\|\}$ are both less than 10^{-6}, where the latter value can be regarded as a measure of feasibility of the current point to the original MPEC.

Table 1: Numerical results for MPECs

Problems	Systems	Global Algorithm			LM Algorithm		
		Iter	ErrObjective	ResEquation	Iter	ErrObjective	ResEquation
bard1	C	20	5.0590	1.2081	41	5.0590	1.2081
	M	21	5.0583	1.2082	51	5.0583	1.2082
	S	59	8.0000	4.9306e-006	68	8.0000	8.7514e-007
bard3	C	16	0.0000	3.1485e-007	16	0.0000	3.1485e-007
	M	100	0.0002	2.3050e-004	100	0.0002	2.3050e-004
	S	100	0.4232	0.0522	100	0.4232	0.0522
bilevel1	C	100	82.3166	0.6758	100	82.3166	0.6758
	M	100	75.6691	0.6957	100	75.6691	0.6957
	S	12	65.0000	1.8838e-006	13	65.0000	6.3106e-007
bilevel3	C	100	0.0002	4.7205e-004	100	0.0002	4.7205e-004
	M	23	0.0000	4.2716e-005	100	0.0000	1.0055e-005
	S	100	0.0012	0.0017	100	0.0012	0.0017
dempe	C	100	4.9217	0.3539	100	4.9217	0.3539
	M	100	3.0044	0.0011	100	3.0044	0.0011
	S	100	2.9990	2.0175e-004	100	2.9990	2.0175e-004
desilva	C	100	0.2502	3.8741e-004	100	0.2502	3.4331e-004
	M	100	1.3665	0.2169	100	1.3665	0.2169
	S	100	0.5552	0.0466	100	0.5033	0.0494
df1	C	100	2.9897e-004	6.0260e-004	100	2.9780e-004	6.0024e-004
	M	100	3.1556e-004	6.3817e-004	100	3.1445e-004	6.3592e-004
	S	100	2.9897e-004	6.0261e-004	100	0.2697	0.0549
ex9.1.1	C	100	4.3028	0.3422	100	4.3283	0.3422
	M	68	7.0000	6.1884e-005	100	7.0000	4.3350e-005
	S	100	8.0583	0.8767	100	8.0583	0.8767
ex9.1.2	C	10	0.0000	1.1593e-007	10	0.0000	2.0671e-007
	M	10	0.0000	6.2350e-005	100	0	4.3620e-005
	S	100	2.1669	0.2276	100	2.1669	0.2276
ex9.1.3	C	100	7.5573	0.7060	100	7.5573	0.7060
	M	100	-2.7851	0.5833	100	-2.7851	0.5833
	S	100	6.3376	0.7118	100	6.3376	0.7118
ex9.1.4	C	100	-11.3622	0.6444	100	-15.4333	0.6444
	M	100	-15.2170	0.6443	100	-18.1163	0.6436
	S	100	-6.0239	0.3318	100	-6.0239	0.3318
ex9.1.5	C	22	0.0000	1.3902e-009	22	0.0000	1.3902e-009
	M	100	0.0000	8.0110e-005	100	0.0000	8.0110e-005
	S	8	0.0000	1.4972e-008	100	0.0000	1.4972e-008

(1) To a certain extent, the global algorithm is more effective than the LM algorithm. In particular, in our tests, the LM algorithm usually used more CPU time than the global algorithm.

(2) Although the S-stationarity system is usually thought to be the best optimality system in the MPEC world, it is not a good way to consider this system only. Our suggestion is that, if it is not too expensive to solve the systems, one may solve the S-/M-/C-stationarity systems and then choose a candidate by comparing their objective values.

The EPEC test problems in Table 5 can be found in [13]. Among these problems, **ex-001** and **outrata4** have equilibrium solutions, but **ex-4** and **outrata3** do not. The computational results are reported in Table 5, in which `ApproxSolution` denotes the approximation equilibrium solution solved by the global algorithm. The results for the first 3 problems

Table 2: Numerical results for MPECs (continued)

Problems	Systems	Global Algorithm			LM Algorithm		
		Iter	ErrObjective	ResEquation	Iter	ErrObjective	ResEquation
ex9.1.6	C	10	0.0000	2.0086e-009	10	0.0000	2.0087e-009
	M	11	0	6.5113e-006	9	0.0000	5.4996e-007
	S	100	20.0825	0.7298	100	20.0825	0.7298
ex9.1.7	C	100	-3.6819	0.6281	100	-3.6819	0.6281
	M	100	-1.6888	0.5725	100	-1.6888	0.5725
	S	100	7.5580	0.4720	100	7.5580	0.4720
ex9.1.8	C	21	0.0000	1.5413e-006	22	0.0000	1.8268e-007
	M	76	0.0000	6.2605e-005	100	0.0000	4.7827e-005
	S	13	0.0000	3.1430e-006	15	0.0000	3.3893e-007
ex9.1.9	C	9	0.0000	4.2956e-008	9	0.0000	4.2957e-008
	M	9	0.0000	1.7204e-005	100	0.0000	4.5959e-005
	S	10	0.0000	1.0334e-006	11	0.0000	1.0993e-007
ex9.1.10	C	30	0.0000	2.2093e-006	31	0.0000	1.2520e-007
	M	81	0.0000	6.0631e-005	100	0.0000	2.6035e-005
	S	29	0.0000	3.9856e-006	33	0.0000	6.7620e-007
ex9.2.1	C	28	0.0000	6.1880e-007	26	0.0000	6.4131e-007
	M	93	0.0000	6.2404e-005	100	0.0000	6.4272e-005
	S	49	0.0000	3.1018e-006	54	0.0000	8.9377e-007
ex9.2.2	C	18	0.0000	2.9096e-007	17	0.0000	2.8221e-007
	M	9	0.0000	7.2306e-007	9	0.0000	1.0153e-007
	S	100	3.6439	0.2615	100	3.6439	0.2615
ex9.2.3	C	11	91.0095	0.6164	100	91.0095	0.6164
	M	12	91.0072	0.6185	100	91.0072	0.6186
	S	100	71.3477	0.5642	100	71.3477	0.5643
ex9.2.4	C	10	0.0000	1.6132e-006	11	0.0000	2.1572e-007
	M	14	0.0000	1.8069e-006	15	0.0000	4.5816e-007
	S	9	3.5000	1.6286e-006	10	3.5000	2.1884e-007
ex9.2.5	C	18	3.0000	3.2535e-006	20	3.0000	5.3059e-007
	M	24	3.0000	4.0460e-006	27	3.0000	7.0950e-007
	S	100	2.1472	0.0954	100	2.1472	0.0955
ex9.2.6	C	100	0.9963	0.7048	100	0.9963	0.7048
	M	100	0.9992	0.7064	100	0.9992	0.7064
	S	100	0.0001	2.2117e-004	100	0.0001	2.2117e-004
ex9.2.7	C	28	0.0000	6.1880e-007	26	0.0000	6.4131e-007
	M	93	0.0000	6.2405e-005	100	0.0000	6.4272e-005
	S	49	0.0000	3.1018e-006	54	0.0000	8.9377e-007

Table 3: Numerical results for MPECs (continued)

Problems	Systems	Global Algorithm			LM Algorithm		
		Iter	ErrObjective	ResEquation	Iter	ErrObjective	ResEquation
ex9.2.8	C	100	1.9950	0.4915	100	1.9950	0.4915
	M	29	0.0000	2.1720e-008	29	0.0000	2.1720e-008
	S	21	0.0000	3.1018e-006	22	0.0000	6.6802e-007
ex9.2.9	C	9	0.0000	2.1223e-008	9	0.0000	2.1223e-008
	M	12	0.0000	4.4449e-007	12	0.0000	4.4449e-007
	S	100	3.0094	0.1542	100	3.0094	0.1542
flp2	C	100	7.6895e-007	0.0013	100	7.6895e-007	0.0013
	M	75	1.0293e-012	6.2396e-005	100	4.4482e-013	4.7302e-005
	S	100	0.8809	0.1503	100	0.8809	0.1503
gauvin	C	8	0.0000	6.1725e-007	7	0.0000	4.8888e-007
	M	9	0.0000	2.0914e-011	7	0.0000	2.7310e-011
	S	100	2.5621e+002	1.4837	100	2.5621e+002	1.4837
jr1	C	8	0.0000	2.5589e-008	8	0.0000	2.5589e-008
	M	8	0.0000	5.6723e-007	8	0.0000	5.6723e-007
	S	46	0.0000	2.0948e-006	49	0.0000	9.5313e-007
jr2	C	10	0.0000	1.2336e-010	10	0.0000	1.2336e-010
	M	9	0.0000	8.8926e-007	9	0.0000	8.8926e-007
	S	100	0.0037	0.0036	100	0.0037	0.0036
kth1	C	7	5.1028e-012	7.4938e-007	7	5.1028e-012	7.4938e-007
	M	84	9.8813e-324	6.2073e-005	100	9.8813e-324	5.1591e-005
	S	100	0.1385	0.0048	100	0.1385	0.0048
kth2	C	7	4.1345e-010	2.6614e-009	7	4.1345e-010	2.6614e-009
	M	7	7.1674e-010	4.5914e-009	7	7.1674e-010	4.5914e-009
	S	8	-1.1626e-015	6.2816e-013	8	-1.1626e-015	6.2816e-013
kth3	C	9	0.0000	9.7090e-007	9	0.0000	9.7090e-007
	M	9	0.0000	6.0865e-008	9	0.0000	6.0865e-008
	S	7	0.0000	2.2261e-007	7	0.0000	2.2261e-007
nash1a	C	12	3.8059e-012	9.5387e-007	12	3.8059e-012	9.5387e-007
	M	80	2.6463e-012	6.2238e-005	100	1.3268e-012	4.9606e-005
	S	100	13.0842	1.0843	100	13.0842	1.0843
outrata31	C	21	0.0000	1.4428e-006	21	0.0000	5.5470e-007
	M	21	0.0000	1.5393e-006	22	0.0000	7.1682e-007
	S	100	1.0554	0.0877	100	1.0554	0.0877
outrata32	C	22	0.2681	9.7882e-007	19	0.2681	8.2409e-007
	M	14	0.2681	1.6971e-006	15	0.2681	6.6307e-007
	S	100	1.1835	0.1712	100	1.1835	0.1713

Table 4: Numerical results for MPECs (continued)

Problems	Systems	Global Algorithm			LM Algorithm		
		Iter	ErrObjective	ResEquation	Iter	ErrObjective	ResEquation
outrata33	C	15	0.0000	5.0379e-007	15	0.0000	5.0379e-007
	M	16	0.0000	2.3177e-007	16	0.0000	2.3177e-007
	S	100	9.9783	0.2497	100	9.9783	0.2497
outrata34	C	99	2.4037	5.0907e-006	100	2.4037	4.6131e-006
	M	100	2.4037	6.0525e-005	100	2.4037	6.0525e-005
	S	100	-0.2117	0.2127	100	-0.2117	0.2127
ralph1	C	7	-5.3493e-020	1.0612e-010	7	-5.3493e-020	1.0612e-010
	M	6	-1.3422e-013	8.4445e-008	6	-1.3422e-013	8.4445e-008
	S	6	-9.5213e-013	1.3309e-012	6	-9.5213e-013	1.3309e-012
ralph2	C	100	-3.0755e-008	2.2293e-005	100	-3.0755e-008	2.2293e-005
	M	100	-1.0569e-005	1.0969e-004	100	-1.0569e-005	1.0969e-004
	S	100	-4.3541e-004	2.2406e-004	100	-4.3541e-004	2.2406e-004
scholtes3	C	10	0.5000	4.6032e-009	10	0.5000	4.6032e-009
	M	11	0.0000	9.3072e-011	11	0.0000	9.3072e-011
	S	11	0.0000	8.7258e-007	11	0.0000	8.7258e-007
scholtes4	C	26	3.4725e-010	1.4375e-009	26	3.4725e-010	1.4375e-009
	M	27	-1.7985e-009	1.0049e-006	27	4.9010e-024	1.2643e-007
	S	100	0.9980	0.0727	100	0.9979	0.0727
scholtes5	C	8	0.0000	3.6761e-008	8	0.0000	3.6761e-008
	M	8	0.0000	3.2950e-008	8	0.0000	3.2950e-008
	S	11	1.5000	2.4975e-006	12	1.5000	5.5246e-007
scale1	C	100	-1.0000	0.0087	100	-1.0000	0.0087
	M	100	-0.9999	0.0088	100	-0.9999	0.0088
	S	100	-0.0639	0.0057	100	0.0639	0.0057
scale2	C	11	0.0000	6.4265e-010	11	0.0000	6.4265e-010
	M	11	0.0000	3.6495e-010	11	0.0000	3.6495e-010
	S	8	0.0000	9.5056e-009	100	0.0000	1.2895e-005
scale3	C	100	-1.0000	0.0086	100	-1.0000	0.0086
	M	100	-1.0000	0.0087	100	-1.0000	0.0087
	S	100	-0.9993	0.0084	100	-0.9993	0.0084
sl2	C	10	0.0000	3.1773e-008	10	0.0000	3.1773e-008
	M	10	0.0000	1.3082e-007	10	0.0000	1.3082e-007
	S	100	1.0000	8.7818e-006	100	1.0000	8.7818e-006
sl4	C	8	0.0000	4.7318e-011	8	0.0000	4.7318e-011
	M	8	0.0000	5.0859e-011	8	0.0000	5.0859e-011
	S	7	0.0000	2.5027e-008	7	0.0000	2.5027e-008

Table 5: Numerical results for EPECs

Problems	Systems	Iter	ApproxSolution	ResEquation
ex-001	C	8	(-1.0000, -1.0000)	3.7951e-009
	M	83	(-1.0000, -1.0000)	6.2172e-005
	S	100	(-0.4157, 0.5134)	0.0354
ex-4	C	100	(0.4904, 0.5029)	0.7051
	M	100	(0.0226, 1.0000)	0.0010
	S	100	(0.0921, 1.0000)	0.0033
outrata3	C	100	(8.8731, 9.9999, -0.0001, -0.0001)	0.0851
	M	100	(9.9926, 1.0000, -0.0000, -0.0000)	0.0904
	S	100	(-0.0003, 0.6198, 9.9999, 0.42223)	0.5535
outrata4	C	93	(0.9117, 0.0700, 0.0978, 0.0978)	1.2678
	M	100	(0.9117, 0.0700, 0.0979, 0.0978)	1.2678
	S	100	(1.7061, 0.4378, 0.6678, 0.6623)	0.1546

coincide basically with our expectation, but the results for the last problem does not. Actually, the point obtained by solving the C-stationarity system is a stationary point of problem (3.2).

We next consider an electricity market model with a three-node lossless direct current network from [24, 23] as indicated in Figure 1. Demand occurs at each node but there

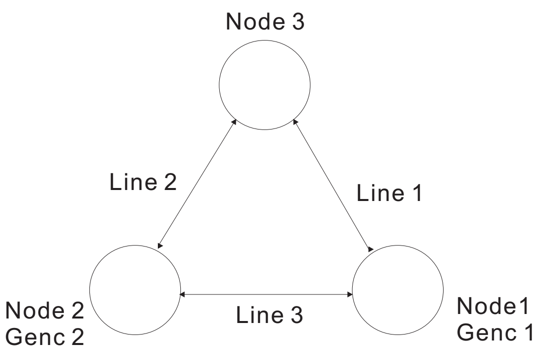

Figure 1: A three-node example with two generators

are only two generators located in node 1 and node 2. There exists a pool operated by an *Independent System Operator* (ISO), which serves as a broker and makes decisions on the market clearing price and power transactions. The ISO leases the transmission system from the network owners and controls the power flows in order to maintain the feasibility of the transmission network. Given the generators' production decisions q_1 and q_2, the ISO solves

Table 6: Payoff parameters at the network

node	a_k	b_k	α_k	β_k	$\bar{q}_k$
k=1	25	1.2	5	1	10
k=2	26.5	1.3	5	1.5	10
k=3	27.9	1.5	–	–	–

the parametric problem

$$\begin{array}{ll} \max\limits_{r_1,r_2,r_3} & \sum\limits_{k=1}^{2}\int_0^{r_k+q_k}(a_k-b_kt_k)\,dt_k+\int_0^{r_3}(a_3-b_3t_3)\,dt_3-\sum\limits_{k=1}^{2}(\alpha_kq_k+\beta_kq_k^2) \\ \text{s.t.} & r_1+r_2+r_3=0, \\ & r_1+q_1\geq 0,\ r_2+q_2\geq 0,\ r_3\geq 0, \\ & -K_l\leq D_{l1}r_1+D_{l2}r_2+D_{l3}r_3\leq K_l,\ l=1,2,3, \end{array} \tag{4.1}$$

where D_{lk} is the *power transfer distribution factor* (PTDF) to calculate the flows on lines; see [24] for details. It is obvious that problem (4.1) is a convex program with linear constraints. Let ϱ, η, and λ^-, λ^+ denote the Lagrange multipliers corresponding to the constraints in problem (4.1). Then, solving problem (4.1) is equivalent to solve its KKT system

$$\begin{cases} a_k-b_k(r_k+q_k)-\varrho+\eta_k-\sum_{l=1}^3 D_{lk}(\lambda_l^+-\lambda_l^-)=0, \quad k=1,2, \\ a_3-b_3r_k-\varrho+\eta_3-\sum_{l=1}^3 D_{l3}(\lambda_l^+-\lambda_l^-)=0, \\ r_1+r_2+r_3=0, \\ 0\leq\eta_k\perp r_k+q_k\geq 0\ k=1,2,\ 0\leq\eta_3\perp r_3\geq 0, \\ 0\leq\lambda_l^-\perp K_l+D_{l1}r_1+D_{l2}r_2+D_{l3}r_3\geq 0, \quad l=1,2,3, \\ 0\leq\lambda_l^+\perp K_l-D_{l1}r_1-D_{l2}r_2-D_{l3}r_3\geq 0, \quad l=1,2,3. \end{cases}$$

Assume that each generator aims at maximizing its own profit and can anticipate the impact of its production on ISO's decision making and its competition's response. Then, we can formulate the generators' optimal decision making problem as follows: For each $k=1,2$,

$$\begin{array}{ll} \max\limits_{q_k,r_k,\varrho,\eta,\lambda} & (a_k-b_k(r_k+q_k))q_k-(\alpha_kq_k+\beta_kq_k^2) \\ \text{s.t.} & 0\leq q_k\leq\bar{q}_k, \\ & a_k-b_k(r_k+q_k)-\varrho+\eta_k-\sum\limits_{l=1}^{3}D_{lk}(\lambda_l^+-\lambda_l^-)=0, \quad k=1,2, \\ & a_3-b_3r_k-\varrho+\eta_3-\sum\limits_{l=1}^{3}D_{l3}(\lambda_l^+-\lambda_l^-)=0, \\ & r_1+r_2+r_3=0, \\ & 0\leq\eta_k\perp r_k+q_k\geq 0\ k=1,2,\ 0\leq\eta_3\perp r_3\geq 0, \\ & 0\leq\lambda_l^-\perp K_l+D_{l1}r_1+D_{l2}r_2+D_{l3}r_3\geq 0, \quad l=1,2,3, \\ & 0\leq\lambda_l^+\perp K_l-D_{l1}r_1-D_{l2}r_2-D_{l3}r_3\geq 0, \quad l=1,2,3. \end{array}$$

The required parameters in our test come from [23] and are given in Tables 6–7 and the computational results are reported in Table 8. From Table 8, we can observe that a C-stationary point and an M-stationary point were obtained by solving the C-stationarity and

Table 7: Power transfer factors at the network

node k	D_{lk}	D_{2k}	D_{3k}
k=1	0.5	-0.5	-0.5
k=2	0.5	-0.5	0.5
K_l	5	5	5

Table 8: Numerical results for electricity market

Systems	Iter	ApproxSolution	ApproxObject[a]	ResEquation
C	25	(6.8110, 6.0750)	(60.4195, 58.1340)	9.7555e-009
M	15	(7.4720, 6.2145)	(58.7083, 55.9762)	3.2262e-007
S[b]	10	(-0.0000, 0.0000)	(1.1847e-005, -3.2152e-005)	4.6425e-006

[a] `ApproxObject` denotes the value of objective functions at the approximate solution.
[b] Iteration terminated due to that a stationary point of problem (3.2) was obtained.

M-stationarity systems respectively, while a stationary point of problem (3.2) was obtained by solving the S-stationarity system.

5 Conclusions

We have reformulated the S-/M-/C-stationarity systems for the EPEC (1.1) as some equations with simple constraints and, furthermore, we have proposed a numerical algorithm for solving these constrained equations. We have shown that the new algorithm is globally and superlinearly convergent, whereas the LM algorithm given in the recent work [14] is locally and superlinearly convergent. However, our computational experience indicates that, compared with the LM algorithm, the global algorithm was not as fast as expected. Perhaps, it would be different for large scale problems. We leave it as a future work.

Acknowledgements. The authors are grateful to the two anonymous referees for their helpful comments and suggestions.

References

[1] R. Behling and A. Fischer, A unified local convergence analysis of inexact constrained Levenberg-Marquardt methods, *Optimization Letters*, to appear.

[2] D.P. Bertsekas, *Nonlinear Programming*, second edition, Athena Scientific, Belmont, Massachusetts, 1999.

[3] J.B. Cardell, C.C. Hitt and W.W. Hogan, Market power and strategic interaction in electricity networks, *Resource and Energy Economics* 19 (1997) 109–137.

[4] A. Ehrenmann, Equilibrium problems with equilibrium constraints and their application to electricity markets, PhD thesis, Fitzwilliam College, 2004.

[5] A. Ehrenmann, Manifolds of mutli-leaders cournot equilibria, *Operations Research Letters* 32 (2004) 121–125.

[6] R. Fletcher, S. Leyffer, D. Ralph and S. Scholtes, Local convergence of SQP methods for mathematical programs with equilibrium constraints, *SIAM Journal on Optimization* 17 (2006) 259–286.

[7] L. Guo and G.H. Lin, Notes on Some Constraint Qualifications for Mathematical Programs with Equilibrium Constraints, *Journal of Optimization Theory and Applications* 156 (2013) 600–616.

[8] R. Henrion and W. Romisch, On M-stationary points for a stochastic equilibrium problem under equilibrium constraints in electricity spot market modeling, *Applications of Mathematics* 52 (2007) 473–494.

[9] B.F. Hobbs, C.B. Metzler and J.S. Pang, Strategic gaming analysis for electric power networks: An MPEC approach, *IEEE Transactions on Power Systems* 15 (2000) 638–645.

[10] X. Hu, Mathematical programs with complementarity constraints and game theory models in electricity markets, Ph.D. Thesis, Department of Mathematics and Statistics, University of Melbourne, 2002.

[11] X. Hu and D. Ralph, Using EPECs to model bilevel games in restructured electricity markets with locational prices, *Operations Research* 55 (2007) 809–827.

[12] S. Leyffer, MacMPEC-ampl collection of mathematical programs with equilibrium constraints, http://wiki.mcs.anl.gov/leyffer/index.php/MacMPEC.

[13] S. Leyffer and T. Munson, Solving multi-leader-common-follower games, *Optimization Methods and Software* 25 (2010) 601–623.

[14] G.-H. Lin, L. Guo and J.J. Ye, Solving mathematical programs with equilibrium constraints as constrained equations, submmited for publication.

[15] B.S. Mordukhovich, Optimization and equilibrium problems with equilibrium constraints in infinite-dimensional spaces, *Optimization* 57 (2008) 715–741.

[16] F.H. Murphy and Y. Smeers, Generation capacity expansion in imperfectly competitive restructured electricity markets, *Operations Research* 53 (2005) 646–661.

[17] J.V. Outrata, A note on a class of equilibrium problems with equilibrium constraints, *Kybernetika* 40 (2004) 585–594.

[18] J.S. Pang and M. Fukushima, Quasi-variational inequalities, generalized Nash equilibrium, and multi-leader-follower games, *Computational Management Science* 2 (2005) 21–56.

[19] H.S. Scheel and S. Scholtes, Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity, *Mathematics of Operations Research* 25 (2000) 1–22.

[20] S. Scholtes, Convergence properties of a regularization scheme for mathematical programs with complementarity constraints, *SIAM Journal on Optimization* 11 (2001) 918–936.

[21] C.L. Su, Analysis on the forward market equilibrium model, *Operations Research Letters* 35 (2007) 74–82.

[22] C.L. Su, Equilibrium problems with equilibrium constraints: Stationarities, algorithms, and applocations, Ph.D. Thesis, Department of Management Science and Engineering, Stanford University, Stanford, 2005.

[23] H. Xu and D. Zhang, Stochastic nash equilibrium problems: Sample average approximation and applications, http://eprints.soton.ac.uk/id/eprint/145459.

[24] J. Yao, I. Adler and S.S. Oren, Modeling and computing two-settlement oligopolistic equilibrium in a congested electricity network, *Operations Research* 56 (2008) 34–47.

[25] J.J. Ye, Constraint qualifications and necessary optimality conditions for optimization problems with variational inequality constraints, *SIAM Journal on Optimization* 10 (2000) 943–962.

[26] J.J. Ye, Necessary and sufficient optimality conditions for mathematical programs with equilibrium constraints, *Journal of Mathematical Analysis and Applications* 307 (2005) 350–369.

Manuscript received 5 February 2012
revised 5 July 2012
accepted for publication 16 August 2012

Lei Guo
School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
E-mail address: guolayne@gmail.com

Gui-Hua Lin
School of Management, Shanghai University, Shanghai 200444, China
E-mail address: guihualin@shu.edu.cn