Applied Mathematical Sciences, Vol. 3, 2009, no. 39, 1943 - 1956

A Note on Smoothing Mathematical Programs with Equilibrium Constraints

M. A. Tawhid

Department of Mathematics and Statistics
School of Advanced Technologies and Mathematics
Thompson Rivers University
900 McGill Road, PO Box 3010
Kamloops, BC V2C 5N3 Canada
mtawhid@tru.ca

Abstract

Mathematical programs with equilibrium constrains (MPECs) in which the constraints are defined by a parametric variational inequality are considered. Recently, nonlinear programming solvers have been used to solve MPECs. Smoothing algorithms have been very successful. In this note, a smoothing approach based on neural network function to solve MPECs is proposed. The performance of the proposed smoothing approach on as set of well-known problems is tested. A very useful and efficient tool for practitioners to solve not only MPEC problems but also more general class of MPECs is provided.

Keywords: Mathematical programs with equilibrium constraints, smoothing approach, neural network function, online solvers.

1 Introduction

In this paper, we consider the following mathematical programs with equilibrium constraints:

$$\begin{array}{ll} \min & f(x,y) \\ s.t. & x \in X \\ & y \in S(x), \end{array} \tag{1}$$

where $f : R^{n+m} \to R$ is a continuously differentiable function, and X is a nonempty set in R^n, $S(x)$ is the solution set of the following parametric variational inequality problem (PVI): find $S(x)$ such that $y \in S(x)$ if and only if $y \in C(x)$ and

$$\langle v - y, F(x,y) \rangle \geq 0, \text{ for all } v \in C(x), \tag{2}$$

where $F : R^{n+m} \to R^m$ is a continuously differentiable function, and set valued mapping C is defined by

$$C(x) := \{y \in R^m : g_i(x,y) \geq 0, i = 1, 2, \cdots, l\}, \tag{3}$$

where $g_i : R^{n+m} \to R$ for all $i = 1, 2, \ldots, l$ are twice continuously differentiable and concave in the second variable. We note that if $l = m$ and $g(x, y) = y$, then the PVI is reduced to the following parametric nonlinear complementarity problem

$$\langle y, F(x,y)\rangle = 0. \tag{4}$$

One difficulty in dealing with MPECs problem is their combinatorial nature and feasible region are nonconvex and nonsmooth due to the variational inequality constraints, in general, there is no feasible point satisfying all inequality constraints strictly which implies the constraints qualification of the nonlinear programming such as Mangasarian-Fromovitz constraint qualification (MFCQ) is violated at any feasible point of a MPEC. Due to this, most of the well-developed theory for nonlinear programming can not be applied directly to MPEC problems.

Many practical problems in multilevel game, capacity enhancement in traffic networks, engineering design, dynamic pricing in telecommunication networks, and economic equilibrium that are modeled using the MPEC formulation, see [24], [26] . As a special case of MPEC, MPEC includes the so called bilevel programming problem, for further material on bilevel program and its applications; see [1], [8] and the references therein.

One of the powerful and effective approaches for solving general MPECs is a class of so-called smoothing algorithms (e.g., [5], [9], [13], [14], [15], [18], [19], [28]). Motivated by these methods, in this article we use the KKT conditions for the variational inequality constraints, we reformulate MPEC problem as a nonsmooth constrained optimization problem. Then by using the proposed smoothing function, we transfer this nonsmooth optimization problem to a smooth nonlinear programming problem. Our neural network approach provides a very useful and efficient tool for practitioners to solve not only MPEC problems but also more general class of MPECs since we use online software. Also, we compare our neural network approach with entropic smoothing approach based on entropic function. Numerical results are given for both practical and academic problems. Both of them enjoy favorable properties such as nonsingularity of the Jacobian, and so on.

2 Equivalent reformulation of MPEC

The following assumptions are needed [9], [27]:

A1 $C(x)$ is nonempty and convex for all $x \in A$, where A is an open bounded set in R^n such that A is containing X.

A2 $C(x)$ is uniformly compact on A, i.e., there exists an open set $B \subseteq R^n$ such that for all $x \in A$, $C(x) \subseteq B$.

A3 F is uniformly strongly monotone with respect y, i.e., there exists a constant $\mu > 0$ such that

$$\langle y_1 - y_2, F(x, y_1) - F(x, y_2)\rangle \geq \mu ||y_1 - y_2||^2 \text{ for all } y_1, y_2 \in C(x).$$

A4 $X \subseteq R^n$ is nonempty and compact.

A5 At each $x \in X$ and $y \in S$, the partial gradients $\nabla_y g_i(x, y)$ of the active constrained are linearly independent.

Under the Assumptions (A1-A3), the authors [17] showed that there exists a unique solution for the variational inequalities. The Assumption (A5) implies that for each solution vector $y \in S(x)$, the variational inequality problem (2) can be rewritten as a system of Karush-Kuhn-Tucker (KKT) conditions [10], i.e.,

$$\begin{aligned} &F(x, y) - \nabla_y g(x, y)^T \lambda = 0, \\ &g(x, y) \geq 0, \quad \lambda \geq 0, \quad \lambda^T g(x, y) = 0, \end{aligned} \tag{5}$$

where $\lambda \in R^l$ is a multiplier which is uniquely determined by Assumption A5. Therefore, under Assumptions A1-A5, the variational inequality constraints in (2) and its KKT conditions in (5) are equivalent, see [17]. MPEC problem (1) can be written as a standard nonlinear programming:

$$\begin{aligned} &\min f(x, y) \\ &\text{s.t. } \ x \in X, \\ &F(x, y) - \nabla_y g(x, y)^T \lambda = 0, \\ &g(x, y) \geq 0, \quad \lambda \geq 0, \quad \lambda^T g(x, y) = 0. \end{aligned} \tag{6}$$

In general, Problem (6) does not satisfy any standard constraint qualification(such as Mangasarian-Fromovitz constraint qualification). Moreover, the complementarity constraints are very complicated and difficult to handle [30].

We consider the following nonsmooth reformulation which is used by Facchinei et al. [9]

$$\begin{aligned} &\min f(x, y) \\ &\text{s.t. } \ x \in X, \\ &F(x, y) - \nabla_y g(x, y)^T \lambda = 0, \\ &g(x, y) - z = 0, \\ &\min\{z, \lambda\} = 0, \end{aligned} \tag{7}$$

where $z \in R^l$ and the "min" operator is taken component wise to the vectors z and λ. Problem (7) will be:

$$\begin{aligned} &\min f(x,y) \\ &\text{s.t. } \; x \in X, \\ &H(x,y,z,\lambda) = 0, \end{aligned} \tag{8}$$

where $H : R^{n+m+l+l} \rightarrow R^{m+l+l}$ defined as

$$H(w) := H(x,y,z,\lambda) = \begin{pmatrix} F(x,y) - \nabla_y g(x,y)^T \lambda \\ g(x,y) - z \\ \min\{z,\lambda\} \end{pmatrix}. \tag{9}$$

Note that MPEC Problem (1) is equivalent to (6), it is shown in [9] that (x^*, y^*) is a global/local solution of the MPEC problem in (1) if and only if there exists a vector (z^*, λ^*) such that $(x^*, y^*, z^*, \lambda^*)$ is a global/local solution of the problem in (6).

It is not easy to solve MPEC problem in (8) because its feasible set is nonsmooth or even disconnected. The min function in (9) is nonsmooth (nondifferentiable) function and makes MPEC problem in (8) difficult to handle.

In the field of complementarity problems, several researchers have proposed various smoothing functions to approximate the min function by a sequence of parameterized smooth (continuously differentiable) problems, and to trace the smooth path which leads to solutions. For example, Kanzow [20] used Chen-Harker-Kanzow-Smale (CHKS) smoothing function to solve complementarity problem [20] and variational inequalities [21]. Facchinei, Jiang, and Qi [9] applied Chen-Harker-Kanzow-Smale smoothing function to smooth MPEC problem (9) and their numerical experiments indicate that the smooth approach is very effective and efficient. Chen and Mangasarian [6], [7] proposed a class of parametric smooth functions, called plus-smooth functions, to approximate the min function which unified the smoothing functions studied in [3], [4], [20], [31], [32]. Roughly speaking, their smoothing functions are derived from double integrals of parameterized probability density functions. Chen-Mangasarian smoothing function has been used to approximate linear and convex inequalities, to solve complementarity problems [6], [7].

Fang and his collaborators used entropic regularization to solve various problems in optimization [11], [12], [23] (for example, min–max problems and semi–infinite programs). The entropic regularization function is defined as

$$\phi_\rho(a,b) = -\frac{1}{\rho} \ln\{e^{-\rho a} + e^{-\rho b}\}, \tag{10}$$

where $\rho > 0$ is the real parameter and $\phi_\rho : R^2 \rightarrow R$. It is easy to see that $\phi_\rho(a,b)$ is a C^∞ function and concave function for any $\rho > 0$.

In this paper, we consider a subclass from Chen-Mangasarian smoothing function to handle MPEC problems. This subclass is known as neural network smoothing function which is defined as

$$\phi(a, b, \epsilon) = b - \epsilon \ln(1 + e^{\frac{b-a}{\epsilon}}). \tag{11}$$

The following lemma summarizes the properties of the neural network. The proof can be found in [2].

Lemma 1 *Properties of $\phi(a, b, \epsilon)$ where $\epsilon > 0$*

1. *For any fixed $\epsilon > 0$, the partial derivative satisfies*

$$(0,0) \leq \left(\frac{\partial \phi(a,b,\epsilon)}{\partial a}, \frac{\partial \phi(a,b,\epsilon)}{\partial b}\right) \leq (1,1), \tag{12}$$

   and $\phi(a, b, \epsilon)$ is k-times continuously differentiable for any positive integer k and for all $(a, b)^T \in R^2$.

2. *For any fixed $(a, b)^T \in R^2$, $\phi(a, b, \epsilon)$ is continuously differentiable, monotonically decreasing and concave with respect to ϵ. In particular, for $\epsilon_1 \geq \epsilon_2 \geq 0$*

$$0 \leq \phi(a, b, \epsilon_2) - \phi(a, b, \epsilon_1) \leq \kappa(\epsilon_1 - \epsilon_2), \tag{13}$$

   where $\kappa = \max\{\kappa_1, \frac{1}{\sqrt{2}}\}$. Furthermore, $\phi(a, b, \epsilon) \to -\infty$ as $\epsilon \to \infty$.

3. *The limit*

$$\lim_{\epsilon \downarrow 0}\left(\frac{\partial \phi(a,b,\epsilon)}{\partial a}, \frac{\partial \phi(a,b,\epsilon)}{\partial b}\right) \text{ exists for any fixed } (a,b)^T \in R^2. \tag{14}$$

The smooth reformulation of Problem (8) will be

$$\begin{array}{l} \min f(x, y) \\ \text{s.t. } \; x \in X, \\ H_\epsilon(x, y, z, \lambda) = 0, \end{array} \tag{15}$$

where $H_\epsilon : R^{n+m+l+l} \to R^{m+l+l}$ defined as

$$H_\epsilon(w) := H_\epsilon(x, y, z, \lambda) = \begin{pmatrix} F(x, y) - \nabla_y g(x, y)^T \lambda \\ g(x, y) - z \\ \Phi_\epsilon(z, \lambda) \end{pmatrix} \tag{16}$$

where $\Phi_\epsilon(z, \lambda) := (\Phi_\epsilon(z_1, \lambda_1), \cdots, \Phi_\epsilon(z_l, \lambda_l))^T \in R^l$.

Since Problem (15) is a regular nonlinear programming, we can solve it by online software. It is known that these software solvers converge to the local minimizers unless certain assumptions need to be imposed on f and X

such as convexity. Since we do not have such assumptions, we expect a local convergence. We minimize the constraint violations (a feasible solution for (15) is not a feasible solution for the original constraints in (8). Indeed, several constrained optimization softwares such as SNOPT [16] try to minimize the constraint violations. We minimize the constraint violations between (15) and (8) by taking $\epsilon \to 0$.

As a consequence of the above lemma, one can prove both Φ_ϵ and $H_\epsilon(x, y, z, \lambda)$ are regular and locally Lipschtiz. In a recent paper, Ralph and Wright [29] have established certain regularity condition under which some of the reformulations (e.g., the product and penalty reformulations) can generate an local optimal solution to the original MPEC.

Remarks

Clearly by Part (1) of Lemma 1, $H_\epsilon(x, y, z, \lambda)$ is continuously differentiable for any fixed $\epsilon > 0$ and by Part (2) of Lemma 1, $H_\epsilon(x, y, z, \lambda)$ bounded by the smooth parameter ϵ.

The Jacobian of $H_\epsilon(x, y, z, \lambda)$ is given by

$$\nabla H_\epsilon(w) := \nabla H_\epsilon(x, y, z, \lambda) = \begin{pmatrix} Q & 0 & -\nabla_y g(x,y)^T \\ \nabla_y g(x,y) & -I & 0 \\ 0 & D_1 & D_2 \end{pmatrix} \qquad (17)$$

where

$$\begin{array}{l} Q := \nabla_y F(x,y) - \sum_{i \in I} \lambda_i \nabla_y^2 g_i(x,y), \\ D_1 := diag(\frac{\partial \phi_\epsilon(z_1,\lambda_1)}{\partial z_1}, \cdots, \frac{\partial \phi_\epsilon(z_l,\lambda_l)}{\partial z_l}) \\ D_2 := diag(\frac{\partial \phi_\epsilon(z_1,\lambda_1)}{\partial \lambda_1}, \cdots, \frac{\partial \phi_\epsilon(z_l,\lambda_l)}{\partial \lambda_l}) \end{array} \qquad (18)$$

and I is the l-dimensional identity matrix. Part (3) of Lemma 1 provides the limiting behavior of the Jacobian $\nabla H_\epsilon(x, y, z, \lambda)$ as smooth parameter $\epsilon \to 0$. This is useful for designing locally fast convergent algorithms.

The following theorem shows the nonsingularity of $\nabla H_\epsilon(x, y, z, \lambda)$ which plays a very important rule in proving the convergence of the algorithm.

Theorem 1 *Given any $\epsilon \neq 0$ and $(x, y, z, \lambda) \in \Omega_\epsilon$ where $\Omega_\epsilon \subset R^{n+m+2l}$ is the feasible set of problem (15), the Jacobian of H_ϵ with respect to the variables (y, z, λ) is nonsingular.*

Proof. Let us show the nonsingularity of ∇H_ϵ in (17). Since $F(., x)$ is strongly monotone, $\nabla_y F(x, .)$ is positive definite. The Hessian matrices $\nabla_y^2 g_i(., x)$ are negative semidefinite for $i \in I$ because all functions $g_i(x, .)$ are concave. Thus, Q in (18) is positive definite. Using (12) for all $i = 1, 2, \cdots, l$, the diagonal matrices D_1 and D_2 in (18) are positive definite. To prove the nonsingularity of ∇H_ϵ, suppose that $\nabla H_\epsilon w = 0$ for some vector $w = (w^{(1)}, w^{(2)}, w^{(3)}) \in$

$R^m \times R^l \times R^l$. It is easy to show w is a zero vector. Thus, ∇H_ϵ is nonsingular. □

Now we give the following examples to illustrate the formulation with neural network smoothing.

Example 1.

The following example is a bilevel program. Since the lower level program is convex, it is equivalent to its optimality condition, which can be formulated as a variational inequality problem.

$$f(x,y) = \tfrac{1}{2}[(y_1-3)^2 + (y_2-4)^2 + (y_3-1)^2]; \qquad X = [0,10],$$

$$F(x,y) = \begin{pmatrix} (1+0.2x)y_1 - (3+1.333x) - 0.333y_3 + 2y_1y_4 - y_5 \\ (1+0.1x)y_2 - x + y_3 + 2y_2y_4 - y_6 \\ 0.333y_1 - y_2 + 1 - 0.1x \\ 9 + 0.1x - y_1^2 - y_2^2 \\ y_1 \\ y_2 \end{pmatrix},$$

$$g_1(x,y) = y_3, \quad g_2(x,y) = y_4, \quad g_3(x,y) = y_5, \quad g_4(x,y) = y_6.$$

The corresponding problem in (6) can be derived as

$$\begin{array}{ll} \min & \tfrac{1}{2}((y_1-3)^2 + (y_2-4)^2) \\ \text{s.t.} & 0 \le x_1 \le 10, \\ \multicolumn{2}{l}{(1+0.2x)y_1 - (3+1.333x) - 0.333y_3 + 2y_1y_4 - y_5 = 0,} \\ \multicolumn{2}{l}{(1+0.1x)y_2 - x + y_3 + 2y_2y_4 - y_6 = 0, \qquad 0.333y_1 - y_2 + 1 - 0.1x - \lambda_1 = 0,} \\ \multicolumn{2}{l}{9 + 0.1x - y_1^2 - y_2^2 - \lambda_2 = 0, y_1 - \lambda_3 = 0, \qquad y_2 - \lambda_4 = 0,} \\ \multicolumn{2}{l}{y_3 \ge 0,\ y_4 \ge 0,\ y_5 \ge 0,\ y_6 \ge 0, \qquad \lambda_1, \lambda_2, \lambda_3, \lambda_4 \ge 0,} \\ \multicolumn{2}{l}{\lambda_1 y_3 = 0,\ \lambda_2 y_4 = 0,\ \lambda_3 y_5 = 0,\ \lambda_4 y_6 = 0} \end{array}$$

and the problem in (15) can be derived as

$$\begin{array}{ll} \min & \tfrac{1}{2}((y_1-3)^2 + (y_2-4)^2) \\ \text{s.t.} & 0 \le x_1 \le 10 \\ \multicolumn{2}{l}{(1+0.2x)y_1 - (3+1.333x) - 0.333y_3 + 2y_1y_4 - y_5 = 0,} \\ \multicolumn{2}{l}{(1+0.1x)y_2 - x + y_3 + 2y_2y_4 - y_6 = 0,} \\ \multicolumn{2}{l}{0.333y_1 - y_2 + 1 - 0.1x - \lambda_1 = 0,} \\ \multicolumn{2}{l}{9 + 0.1x - y_1^2 - y_2^2 - \lambda_2 = 0} \\ \multicolumn{2}{l}{y_1 - \lambda_3 = 0, \quad y_2 - \lambda_4 = 0, \quad y_3 - z_1 = 0,} \\ \multicolumn{2}{l}{y_4 - z_2 = 0, \quad y_5 - z_3 = 0, \quad y_6 - z_4 = 0,} \\ \multicolumn{2}{l}{\lambda_1 - \epsilon * ln(1 + e^{(\lambda_1 - z_1)/\epsilon}) = 0, \quad \lambda_2 - \epsilon * ln(1 + e^{(\lambda_2 - z_2)/\epsilon}) = 0,} \\ \multicolumn{2}{l}{\lambda_3 - \epsilon * ln(1 + e^{(\lambda_3 - z_3)/\epsilon}) = 0, \quad \lambda_4 - \epsilon * ln(1 + e^{(\lambda_4 - z_4)/\epsilon}) = 0.} \end{array}$$

Example 2. This example has been used in [9].

$$f(x,y) = 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 + 100\left[\max\{0, x_1 + x_2 + y_1 - 2y_2 - 40\}\right]^2;$$
$$X = [0,50] \times [0,50],$$

$$F(x,y) = \begin{pmatrix} 2y_1 - 2x_1 + 40 \\ 2y_2 - 2x_2 + 40 \end{pmatrix}$$

$$\begin{array}{lll} g_1(x,y) = y_1 + 10, & g_2(x,y) = -y_1 + 20, & g_3(x,y) = y_2 + 10, \\ g_4(x,y) = -y_2 + 20, & g_5(x,y) = x_1 - 2y_1 - 10, & g_6(x,y) = x_2 - 2y_2 - 10. \end{array}$$

The corresponding problem in (6) can be derived as

$$\begin{array}{l} \min \quad 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 + 100\,[\max\{0, x_1 + x_2 + y_1 - 2y_2 - 40\}]^2 \\ \text{s.t.} \quad 0 \leq \ x_1, x_2 \leq 50, \\ y_1 + 10 \geq 0, \quad -y_1 + 20 \geq 0, \quad y_2 + 10 \geq 0, \\ -y_2 + 20 \geq 0, \quad x_1 - 2y_1 - 10 \geq 0, \quad x_2 - 2y_2 - 10 \geq 0, \\ 2y_1 - 2x_1 + 40 - \lambda_1 + \lambda_2 + 2\lambda_5 = 0, \qquad 2y_2 - 2x_2 + 40 - \lambda_3 + \lambda_4 + 2\lambda_6 = 0, \\ \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6 \geq 0, \quad \lambda_1(y_1 + 10) = 0, \quad \lambda_2(-y_2 + 20) = 0, \quad \lambda_3(y_2 + 10) = 0, \\ \lambda_4(-y_2 + 20) = 0, \quad \lambda_5(x_1 - 2y_1 - 10) = 0, \quad \lambda_6(x_2 - 2y_2 - 10) = 0 \end{array}$$

and the problem in (15) can be derived as

$$\begin{array}{l} \min \quad 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60 + 100\,[\max\{0, x_1 + x_2 + y_1 - 2y_2 - 40\}]^2 \\ \text{s.t.} \quad 0 \leq \ x_1, x_2 \leq 50, \\ 2y_1 - 2x_1 + 40 - \lambda_1 + \lambda_2 + 2\lambda_5 = 0, \qquad 2y_2 - 2x_2 + 40 - \lambda_3 + \lambda_4 + 2\lambda_6 = 0, \\ y_1 + 10 - z_1 = 0, \ -y_1 + 20 - z_2 = 0, \ y_2 + 10 - z_3 = 0, \\ -y_2 + 20 - z_4 = 0, \ x_1 - 2y_1 - 10 - z_5 = 0, \ x_2 - 2y_2 - 10 - z_6 = 0, \\ \lambda_1 - \epsilon * ln(1 + e^{(\lambda_1 - z_1)/\epsilon}) = 0, \qquad \lambda_2 - \epsilon * ln(1 + e^{(\lambda_2 - z_2)/\epsilon}) = 0, \\ \lambda_3 - \epsilon * ln(1 + e^{(\lambda_3 - z_3)/\epsilon}) = 0, \qquad \lambda_4 - \epsilon * ln(1 + e^{(\lambda_4 - z_4)/\epsilon}) = 0, \\ \lambda_5 - \epsilon * ln(1 + e^{(\lambda_5 - z_5)/\epsilon}) = 0, \qquad \lambda_6 - \epsilon * ln(1 + e^{(\lambda_6 - z_6)/\epsilon}) = 0. \end{array}$$

3 Numerical results with SNOPT

Our main goal in this paper is to apply our proposed approach to MPEC problems by using online available software. We show the performance of the neural network approach on a set of well-known problems. We used AMPL as our modeling language and took advantage of the web-based submission interface available on the new NEOS 5.0 server.

We use the set of test problems which have been used by the authors in [9], [25], [27]. We compare our neural network with the entropic approach.

The user must set a parameter value prior to the execution of the solver. We used a standard value of $1e-4$, adjusting it as necessary to produce usable results.

We will refer to entropic regularization approach using smoothing function in (10) as ERA . Our neural network smoothing function will be abbreviated NNA (neural network approach using neural network function in (11)).

In the following table, the first column numbers the test problems. Column two shows the starting points used to test the various problems, while columns three and six give the best objective function values f^* reported by the solver, with smaller values being better solutions. Columns four and seven give the optimum solutions which correspond to the objective functions. Finally, columns five and eight give the number of iterations each approach takes to converge to the solution, and column 9 reports the value of the parameter that was used by our proposed functions to obtain a result. SNOPT 6.2 solver is one of the best-known SQP solvers available. SNOPT employs a sparse sequential quadratic programming algorithm with Quasi-Newton approximations. In particular, SNOPT allows the nonlinear constraints to be violated (if necessary) and minimizes the sum of such violations. We used the default parameter settings for the SNOPT solver.

For problems $1-5$, both approaches produced comparable answers and provided comparable total iterations over all five problems. Also, for all these problems SNOPT produced the following error when using the NNA approach with an ϵ value which is too small: "SNOPT $6.2-2$: Numerical error: the general constraints cannot be satisfied accurately".

In problem 6, both approaches converged to the same result. In problem 7, both ERA and NNA had trouble converging to the best reported value. In fact, the parameter ρ for the ERA approach had to be reduced to 1.0 in order to produce a usable result. The NNA approach converged to the best value in only one of the two cases while the ERA approach could not converge to the best result from either starting point. Finally, when using the NNA approach from the $(50, 50)$ starting point, a value of ρ which was too small produced the following error: "SNOPT $6.2-2$: The current point cannot be improved."

In problem 8, the values that were obtained corresponded to the optimum solutions posted on the MacMPEC website [22]. We are unsure as to the cause of this discrepancy at this point. In terms of iterations, both ERA and the proposed NNA approach performed the most efficiently, with both of them performing almost identically in terms of iterations.

Problem 9 caused more problems for the ERA approach, with three of the five points tested failing to reach the optimum solution. The NNA approach proposed in this paper fared the best, converging to the optimum solution from four out of five of the tested starting points.

In problem 10, both the ERA and NNA converged to the optimum values at every starting point while the remaining two could not converge to the optimum value from any starting point. The approach did reach the optimum objective. Both of the proposed approaches performed efficiently on this problem, except when using the zero vector as a starting point for the NNA approach. In this case the smoothing function is equal to $-\epsilon\ \ln(2)$ and, excessive iterations are spent to minimize the violation of the equality constraint. When $(0.0001, 0.0001, 0.0001, 0.0001)$ was used instead of $(0, 0, 0, 0)$ with the NNA approach, the number of iterations dropped dramatically from 141 to 53, while still producing the same result. Finally, problem 11 saw all approaches reaching the optimum solution from all starting points.

		ERA			NNA			
No.	Start	f^*	x^*	It.	f^*	x^*	It.	ρ
1	0	3.2151	4.0780	25	3.2151	4.0780	26	1e-2
	10	3.2151	4.0780	24	3.2151	4.0780	29	1e-2
2	0	3.4494	5.1536	23	3.4494	5.1536	29	1e-2
	10	3.4494	5.1536	18	3.4494	5.1536	25	1e-2
3	0	4.6043	2.3894	21	4.6041	2.3895	26	1e-2
	10	4.6043	2.3894	19	4.6043	2.3894	19	1e-2
4	0	6.5927	1.3731	17	6.5927	1.3731	17	1e-2
	10	6.5927	1.3731	19	6.5927	1.3731	19	1e-2
5	(0,0)	-0.99999	(0.5005,0.5005)	15	-0.99999	(0.5005,0.5005)	13	1e-4
	(2,2)	-0.99999	(0.5005,0.5005)	20	-0.99999	(0.5005,0.5005)	17	1e-4
6	0	-3266.6667	93.3333	5	-3266.6667	93.3333	5	1e-4
	100	-3266.6667	93.3333	6	-3266.6667	93.3333	6	1e-4
	200	-3266.6667	93.3333	6	-3266.6667	93.3333	6	1e-4
7	(25,25)	5.1676	(24.9757,30.1322)	34	4.9995	(25,30)	33	1e-4
	(50,50)	5.1676	(24.9757,30.1322)	37	5.1676	(24.9757,30.1322)	39	1e-1
8.1	0	-230.8232	47.036	19	-230.8232	47.036	19	1e-1
	150	-230.8232	47.036	21	-230.8232	47.036	21	1e-1
8.2	0	-129.9119	34.9942	18	-129.9119	34.9942	18	1e-1
	150	-129.9119	34.9942	21	-129.9119	34.9942	21	1e-1
8.3	0	-36.9331	18.1332	17	-36.9331	18.1332	17	1e-1
	150	-36.9331	18.1332	22	-36.9331	18.1332	22	1e-1
8.4	0	-7.06178	7.55197	19	-7.06178	7.55197	19	1e-1
	150	-7.06178	7.55197	24	-7.06178	7.55197	24	1e-1
8.5	0	-0.1790	1.0663	19	-0.1790	1.0663	19	1e-1
	150	-0.1790	1.0663	23	-0.1790	1.0663	23	1e-1
8.6	0	-354.7021	50	17	-354.7021	50	17	1e-1
	50	-354.7021	50	14	-354.7021	50	14	1e-1
8.7	0	-241.4420	40	17	-241.4420	40	17	1e-1
	40	-241.4420	40	14	-241.4420	40	14	1e-1
8.8	0	-90.7491	25.2584	20	-90.7491	25.2584	20	1e-1
	30	-90.7491	25.2584	19	-90.7491	25.2584	19	1e-1
8.9	0	-25.7478	13.2974	21	-25.7478	13.2974	21	1e-1
	25	-25.7478	13.2974	20	-25.7478	13.2974	20	1e-1
8.10	0	-6.1176	6.3655	20	-6.1176	6.3655	20	1e-1
	20	-6.1176	6.3655	19	-6.1176	6.3655	19	1e-1
9	(0,0)	5.97398e-11	(9.0962,5.9038)	33	1.5705e-12	(5,9)	15	1e-4
	(5,5)	1.1987e-12	(5,9)	11	4.1063e-13	(5,9)	11	1e-4
	(10,10)	4.4995e-11	(9.0977,5.9023)	34	4.4995e-11	(9.0977,5.9023)	34	1e-2
	(10,0)	8.3173e-11	(9.0938,5.9062)	33	4.1691e-13	(5,9)	11	1e-4
	(0,10)	6.1944e-12	(5,9)	19	6.1927e-12	(5,9)	18	1e-4
10	(0,0,0,0)	-6600.00	(7,3,12,18)	139	-6600.00	(7,3,12,18)	141	1e-1
	(0,5,0,20)	-6600.00	(7,3,12,18)	56	-6600.00	(7,3,12,18)	59	1e-1
	(5,0,15,10)	-6600.00	(7,3,12,18)	72	-6600.00	(7,3,12,18)	54	1e-1
	(5,5,15,15)	-6600.00	(7,3,12,18)	60	-6600.00	(7,3,12,18)	58	1e-1
	(10,5,15,10)	-6600.00	(7,3,12,18)	50	-6600.00	(7,3,12,18)	49	1e-1
11	(0,0)	-12.6787	(0,2)	21	-12.6787	(0,2)	21	1e-1
	(0,2)	-12.6787	(0,2)	21	-12.6787	(0,2)	21	1e-1
	(2,0)	-12.6787	(0,2)	27	-12.6787	(0,2)	28	1e-1

Table 1: The NNA with ERA using **SNOPT**

Acknowledgments This research is supported in part by the Natural Sciences and Engineering Research Council of Canada.

References

[1] J. F. Bard, *Practical Bilevel Optimization Algorithms and applications* Kluwer Academic Publishers, Dordrecht, 1998.

[2] B. Chen and X. Chen, A global and local superlinear continuation-smoothing method for P_0 and R_0 NCP or monotone NCP, SIAM on Journal Optimization, 9 (1999), 624-645.

[3] B. Chen P.T. Harker, A non-interior-point continuation method for linear complementarity problems, SIAM on Journal Matrix Analysis and Applications, 14 (1993), 1168-1190.

[4] X. Chen and L. Qi, A paramtereized Newton method and a Broyden-like method for solving nonsmooth equation, Computational Optimization and Applications, 3 (1994), 157-179.

[5] X. Chen and M. Fukushima, A smoothing method for a mathematical program with P-matrix linear complementarity constraints, Computational Optimization and Applications, 27 (2004), 223-246.

[6] C. Chen and Q. L. Mangasarian, Smoothing methods for convex inequalities and linear complementarity problems, Mathematical Programming, 71 (1995), 51-69.

[7] C. Chen and Q. L. Mangasarian, A class of smoothing functions for nonlinear and mixed complementarity problems, Computational Optimization and Applications, 5 (1996), 97-138.

[8] S. Dempe, Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints, Optimization, 52 (2003), 333-359.

[9] F. Facchinei, H. Jiang, L. Qi, A smoothing method for mathematical programs with equilibrium constraints, Mathematical Programming, 85 (1999), 107-134.

[10] F. Facchinei and J.-S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Vol. I and II, Springer Series in Operations Research, Springer-Verlag, New York, 2003.

[11] S.-C. Fang, J. R. Rajasekera, H.-S. J. Tsao, Entropy Optimization and Mathematical Programming, International Series in Operations Research & Management Science, 8. Kluwer Academic Publishers, Boston, MA, 1997.

[12] S.-C. Fang and S.-Y Wu, Solving min-max problems and linear semi-infinite programs, Computers & Mathematics with Applications, 32 (1996), 87-93.

[13] R. Fletcher and S. Leyffer, Solving mathematical programs with complementarity constraints as nonlinear programs, Optimization Methods and Software, 19 (2004), 15-40.

[14] M. Fukushima and J.-S. Pang, Convergence of a smoothing continuation method for mathematical programs with complementarity constraints, Ill-Posed Variational Problems and Regularization Techniques (Trier, 1998), 99-110 (1999), Lecture Notes in Econom. and Math. Systems, 477, Springer, Berlin.

[15] M. Fukushima, Z-Q Luo, and J.-S. Pang, A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints, Computational Optimization and Applications, 10 (1998), 5-34.

[16] P.E. Gill, W. Murray, M.A. Saunders, A. Drud, and E. Kalvelagen, GAMS/SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization, 2000. http://www.games.om/docs/solver/snopt.pdf.

[17] P.T. Harker and J.-S. Pang, Finite dimension variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications, Mathematical Programming, 48 (1990), 161-220.

[18] H. Jiang and D. Ralph, Smooth SQP methods for mathematical programs with nonlinear complementarity constraints, SIAM Journal on Optimization, 10 (2000), 779-808.

[19] H. Jiang and D. Ralph, Extension of quasi-Newton methods to mathematical programs with complementarity constraints, Computational Optimization and Applications, 25 (2003), 123-150.

[20] C. Kanzow, Some noninterior continuation methods for linear complementarity problems, SIAM Journal on Matrix Analysis and Applications, 17 (1996), 851-868.

[21] C. Kanzow and H. Jiang, A continuation method for (strongly) monotone variational inequalities, Mathematical Programming, 81 (1998), 109-125.

[22] S. Leyffer, MacMPEC: AMPL collection of MPECs, April 2004, http://www.mcs.anl.gov/leyffer/MacMPEC.

[23] X.-S Li, Xing-Si and S.-C Fang, On the entropic regularization method for solving min-max problems with applications, Mathematical Methods of Operations Research, 46 (1997), 119-130.

[24] Z.-Q. Luo, J.-S. Pang and D. Ralph, Mathematical Programs with Equilibrium Constraints, Cambridge University Press, Cambridge, 1996.

[25] J. Outrata, On optimization problems with variational inequality constraints, SIAM Journal on Optimization, 4 (1994), 340-357.

[26] J. Outrata and J. Zowe, Nonsmooth Approach to Optimization Problems with Equilibrium Constraints, Theory, Applications and Numerical Results, Nonconvex Optimization and its Applications, 28. Kluwer Academic Publishers, Dordrecht, 1998.

[27] J. Outrata and J. Zowe, A numerical approach to optimization problems with variational inequality constraints, Mathematical Programming, 68 (1995), 105-130.

[28] D. Ralph, Sequential quadratic programming for mathematical programs with linear complementarity constraints, Computational Techniques and Applications: CTAC95 (Melbourne, 1995), pp. 663-668, World Sci. Publishing, River Edge, NJ, 1996.

[29] D. Ralph and S. J. Wright, Some properties of regularization and penalization schemes for MPECs, Optimization Methods and Software, 19 (2004), 527-556.

[30] S.M. Robinson, Stability theory for systems of inequalities. II. Differentiable nonlinear systems, SIAM Journal on Numerical Analysis, 13 (1976), 497-513.

[31] S. Smale, Algorithms for Solving Equations, Proceedings of International Congress of Mathematicians (Berkeley, California), pp. 172-195, 1986.

[32] I. Zang, A smoothing-out technique for min-max optimization, Mathematical Programming, 19 (1980), 61-71.

Received: January, 2009