Expert Systems with Applications 38 (2011) 231–234

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A neural network approach for solving mathematical programs with equilibrium constraints ☆

Yibing Lv [a,*], Zhong Chen [a], Zhongping Wan [b]

[a] School of Information and Mathematics, Yangtze University, Jingzhou 434023, PR China
[b] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, PR China

ARTICLE INFO

Keywords:
Equilibrium constraints
Neural network
Asymptotic stability
Optimal solution

ABSTRACT

A neural network approach is presented for solving mathematical programs with equilibrium constraints (MPEC). The proposed neural network is proved to be Lyapunov stable and capable of generating approximal optimal solution to the MPEC problem. The asymptotic properties of the neural network are analyzed and the condition for asymptotic stability, solution feasibility and solution optimality are derived and the transient behavior of the neural network is simulated and the validity of the network is verified with numerical examples.

1. Introduction

The mathematical program with equilibrium constraints (MPEC) is an optimization problem, whose constraints include variational inequalities (Harker & Pang, 1988). Such problems play an important role, for example, in the design of transportation network (Harker & Pang, 1990), in economic modelling (Tobin, 1992), and in shape optimization (Haslinger & Neittaanmaki, 1988). Further material on the MPEC problem and its applications can also be found in Luo, Pang, and Ralph (1996).

We are interested in algorithms for solving the MPEC problem, which is known to be a very difficult problem, being non-smooth and non-convex also under very favourable assumptions. In fact, few successful numerical methods have been proposed to solve this kind of problem (Facchinei, Jiang, & Qi, 1999).

In recent years, neural computing has become an important means to provide real-time solutions to some optimization problems, especially large-scale problems. In fact, there have been various types of neural networks proposed for solving linear programs, nonlinear programs, variational inequalities et al., we cite for example (Chen, Leung, & Leung, 2002; Kennedy & Chua, 1998; Lan & Wen, 2007; Xia & Wang, 1998; Zhang & Constantinides, 1992). Compared with classical optimization approaches, the prominent advantage of neural computing is that it can converge to the equilibrium point (optimal solution) rapidly, and this advantage motivates us to propose an efficient algorithm, which is based on the neural network approach, for the MPEC problem. It is noted that there are few reports on solving the MPEC problem using neural network approach.

Our strategy can be outlined as follows. By using the Kuhn–Tucker optimality conditions for the variational inequality constraints, we reformulate the MPEC problem as a normal non-smoothly constraints optimization problem. Then, to avoid the difficulty of the non-smooth constraints, we smooth the non-smoothly constraints optimization problem and propose a novel neural network for the smoothed program. Towards these ends, the rest of the paper is organized as follows. In Section 2, we will firstly give some preliminaries. In Section 3, a smoothing technique is introduced by which the MPEC problem is approximated by a sequence of smooth optimization problems. Section 4 is devoted to propose a neural network for solving the smoothed problem and derive the conditions for asymptotic stability, solution feasibility and solution optimality. Numerical examples are given in Section 5. Finally we conclude our paper.

2. Problem statement and preliminaries

We consider the following mathematical program with equilibrium constraints

$$\begin{aligned} &\min f(x,y)\\ &s.t. \quad h(x) \leqslant 0\\ &\qquad y \in S(x) \end{aligned} \tag{1}$$

where $x \in R^n$, $y \in R^m$, and f: $R^{n\times m} \to R$, h: $R^n \to R^p$ are continuous differentiable, for each $x \in X_{ad} = \{x \in R^n: h(x) \leqslant 0\}$ and for a continuously differentiable function F: $R^{n+m} \to R^m$, $S(x)$ is the solution set of the variational inequality (VI) defined by the pair $(F(x,\cdot),C(x))$, i.e., $y \in S(x)$ if and only if $y \in C(x)$ and

☆ Supported by the National Natural Science Foundation of China (10926168, 70771080).

* Corresponding author.
E-mail address: Lvyibing2001@gmail.com (Y. Lv).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.06.050

$$(v-y)^T F(x,y) \geqslant 0 \quad \text{for all } v \in C(x) \tag{2}$$

$C(x)$ is defined by

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

with g: $R^{n+m} \to R^l$ twice continuously differentiable and concave in the variable y. We indicate by $I(x,y)$ the set of active constraints, i.e.,

$$I(x,y) = \{i : g_i(x,y) = 0\}$$

We make the following assumptions (see also Facchinei et al., 1999):

A1. $X_{ad} \subseteq R^n$ is nonempty and compact and $C(x) \notin \emptyset$ for all $x \in A$, where A is an open set containing X_{ad}.

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

$$C(x) \subseteq B$$

A3. F is uniformly strongly monotone with respect to y on $A \times B$, i.e., there exists a constant $\alpha > 0$ such that for all $(x,y) \in A \times B$ and $d \in R^m$,

$$d^T \nabla_y F(x,y) d \geqslant \alpha \|d\|^2$$

A4. At each $x \in X_{ad}$ and $y \in S(x)$, the partial gradients $\nabla_y g_i(x,y)$, $i \in I(x,y)$ are linearly independent.

Based on the above assumptions, the MPEC problem (1) can be reformulated as the following standard nonlinear program (Facchinei et al., 1999)

$$\begin{aligned} &\min\ f(x,y)\\ &s.t.\quad h(x) \leqslant 0\\ &\qquad F(x,y) - \nabla_y g(x,y)\lambda = 0\\ &\qquad g(x,y) \geqslant 0, \quad \lambda \geqslant 0, \quad \lambda^T g(x,y) = 0 \end{aligned} \tag{4}$$

Problem (4) is non-convex and non-differentiable, moreover the regularity assumptions which are needed to successfully handle smooth optimization problems are never satisfied (Luo et al., 1996), which brings great obstacle to use the neural network approach to solve Problem (4).

We are then forced to further reformulate the MPEC problem (1). To this end we consider the following non-smooth equivalent reformulation of Problem (4)

$$\begin{aligned} &\min\ f(x,y)\\ &s.t.\quad h(x) \leqslant 0\\ &\qquad F(x,y) - \nabla_y g(x,y)\lambda = 0\\ &\qquad g(x,y) - z = 0\\ &\qquad -2\min(z,\lambda) = 0 \end{aligned} \tag{5}$$

where $z \in R^l$ and the min operator is applied componentwise to the vectors z and λ. The reason for the introduce of the multiplicative factor -2 before min operator will become apparent shortly. In the following content, we will introduce a smooth method for Problem (5).

3. Smoothing problem

Let $\mu \in R$ be a parameter. Define the function ϕ_μ: $R^2 \to R$ by

$$\phi_\mu(a,b) = \sqrt{(a-b)^2 + 4\mu^2} - (a+b)$$

The above function $\phi_\mu(a,b)$ has the property that $\phi_\mu(a,b) = 0$ if and only if $a \geqslant 0$, $b \geqslant 0$, $ab = \mu^2$, and for every $\mu \neq 0$, $\phi_\mu(a,b)$ is smooth for every a, b. Further, for every (a,b), $\lim_{\mu\to 0}\phi_\mu(a,b) = -2\min(a,b)$. Then, the function is therefore a smooth perturbation of the min function (Facchinei et al., 1999).

Based on the definition of $\phi_\mu(a,b)$, it is obvious that Problem (5) can be approximated by

$$\begin{aligned} &\min\ f(x,y)\\ &s.t.\quad h(x) \leqslant 0\\ &\qquad F(x,y) - \nabla_y g(x,y)\lambda = 0\\ &\qquad g(x,y) - z = 0\\ &\qquad \phi_\mu(z,\lambda) = (\phi_\mu(z_1,\lambda_1),\ldots,\phi_\mu(z_l,\lambda_l)) = 0 \end{aligned} \tag{6}$$

Following Problem (6), we overcome the difficulty that Problem (4) dose not satisfy any regularity assumptions, which are needed for successfully handling smooth optimization problems, and pave the way for using neural network approach to solve Problem (4). To make the discussion simpler, we introduce the following notations

$$G(x,y,z,\lambda) = h(x), \quad H(x,y,z,\lambda) = \begin{pmatrix} F(x,y) - \nabla_y g(x,y)\lambda \\ g(x,y) - z \\ \phi_\mu(z,\lambda) \end{pmatrix}$$

Let $w = (x,y,z,\lambda)$, then we can rewrite the above Problem (6) more compactly as

$$\begin{aligned} &\min\ f(w)\\ &s.t.\quad G_\tau(w) \leqslant 0,\ \tau = 1,\ldots,p\\ &\qquad H_k(w) = 0,\ k = 1,\ldots,m+l+l \end{aligned} \tag{7}$$

Definition 1. Let w be a feasible point of Problem (7) and $L = \{l : G_\tau(x') = 0, \tau = 1,\ldots,p\}$. We say that w is a regular point if the gradients $\nabla H_1(x'),\ldots,\nabla H_{m+l+l}(w)$ and $\nabla G_\tau(w)$, $\tau \in L$ are linearly independent.

The following result gives the relationship between the solution of Problem (7) and that of the MPEC problem (1).

Theorem 1 (Facchinei et al. (1999)). *Let w^k and $\mu^k \to 0$ be two sequence such that, for every k, w^k is a stationary point for Problem (7). Suppose that $\{w^k\} \to w^*$. Then $w^* = (x^*, y^*, z^*, \lambda^*)$ is a strong C-stationary point of the MPEC problem (1).*

4. Neural network for MPEC problem

4.1. Definition of the neural network

We can define the following Lagrange function of Problem (7)

$$L(w,Y,\gamma,r) = f(w) + \sum_{k=1}^{m+l+l} \gamma_k H_k(w) + \sum_{\tau=1}^{p} r_\tau \left[G_\tau(w) + Y_\tau^2\right]$$

where $Y \in R^p$ is slack variable and $\gamma \in R^{m+l+l}$, $\tau \in R^p$ are referred as the Lagrange multiplier.

Now, our aim is to design a neural network that will settle down to an equilibrium, which is also a stationary point of the Lagrange function $L(w,Y,\gamma,r)$. We can use the gradient system to construct the following neural network for solving the MPEC problem (1):

$$(MPECNN) \quad \begin{cases} \frac{dw}{dt} = -\nabla_w L(w,Y,\gamma,r) \\ \frac{dY}{dt} = -\nabla_Y L(w,Y,\gamma,r) \\ \frac{d\gamma}{dt} = \nabla_\gamma L(w,Y,\gamma,r) \\ \frac{dr}{dt} = \nabla_r L(w,Y,\gamma,r) \end{cases} \tag{8}$$

or, in component form,

$$\begin{aligned} &\frac{dw_j}{dt} = -\frac{\partial f}{\partial w_j} - \sum_{k=1}^{m+l+l} \gamma_k \frac{\partial H_k}{\partial w_j} - \sum_{\tau=1}^{p} \mu_\tau \frac{\partial G_\tau}{\partial w_j}, \quad j = 1,\ldots,m+n+l+l\\ &\frac{dY_\tau}{dt} = -2r_\tau Y_\tau, \quad \tau = 1,\ldots,p\\ &\frac{d\gamma_k}{dt} = H_k(w), \quad k = 1,\ldots,m+l+l\\ &\frac{dr_\tau}{dt} = G_\tau(w) + Y_\tau^2, \quad \tau = 1,\ldots,p \end{aligned}$$