

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

A Neural Network Model for Equality and Inequality Constrained Minimax Problems

Long-Hua Ma, Yu Zhang, Chun-Ning Yang and Hui Li
School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, 310027, China

Abstract: For signal processing and process control, the minimax problem is a crucial point in research subjects. But efficient solutions to equality and inequality constrained nonlinear general minimax problems are relatively scarce. A minimax neural network model was proposed to solve the general minimax problem based on penalty function. In this model, the unique requirement is that the objective function and constraint functions should be first-order differentiable. In addition to the global stability analysis based on the Lyapunov function, the proposed model was simulated and its validity was evaluated with numerical results. Experimental results demonstrated that the proposed minimax neural network model can solve the problem in seconds which is more efficient than the conventional genetic algorithm and simplex genetic algorithms.

Key words: Minimax problem, neural networks, penalty method, Lyapunov stability, optimization

INTRODUCTION

The Minimax Problem (MMP) is a kind of optimization problem to find the solution x with the minimum worst cost f , where some related parameter y is chosen by an adversary. At the same time, the design may have to satisfy a set of constraints. A general MMP with constraints can be formulated as follows:

$$\begin{aligned} \min_x \max_y f(x, y) \\ \text{st. } h(x, y) = 0 \\ g(x, y) \geq 0 \end{aligned} \quad (1)$$

where, $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, $y = (y_1, y_2, \dots, y_r)^T \in \mathbb{R}^r$, $f: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}$, $h: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^m$ and $g: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^{m-n}$. Minimax problems have been found in many fields such as scheduling (Herrmann, 1999), network design (Kalyanasundaram *et al.*, 1999), filter design (Chao *et al.*, 2000), controller design (Zheng *et al.*, 2001), function approximation (Demyanov and Malozemov, 1974) and so on. The main difficulty is that how to make the cost function decrease with x while increase with y simultaneously. It is not an easy task to simply minimizing the cost function with x while fixing y and then maximizing the cost function with y while fixing x , since the cost function value may oscillate between peaks and valleys.

Several authors have investigated the minimax problem and introduced some heuristic solutions to this problem. Herrmann adopted the Genetic Algorithm (GA) over a set of possible scenarios (Herrmann, 1999; Liu, 1998) utilized GA to solve the minimax problem with both

continuous variables. Longhua *et al.* (2001) proposed a Simplex-GA (SGA) based scheme through studying the convergence speed and solution accuracy of the GA approach (Longhua *et al.*, 2001; Ye *et al.*, 1997) presented a neural network model for unconstrained minimax optimization and proved its stability for some convex cases. Tao and Fang (2000) proposed a quadratic neural network model for quadratic minimax problems under some positive/negative matrix assumptions.

In virtue of penalty functions, this letter develops a kind of minimax neural network for the general minimax optimization problem with both equality-based and inequality-based constraints. The only requirement is that the objective function and constraint functions should be first-order differentiable. When the alternative searching point is forced into the feasible region by penalty terms, our neural network tries to solve the unconstrained minimax problem approximately similar to Ye *et al.* (1997) neural network. Thus, the neural network model of Ye *et al.* (1997) is used as a special case of ours safely. Although there are several types of penalty functions, for the sake of the global stability of neural network, the penalty function with an archetypical form is chosen which guarantees a large enough penalty parameter.

PROPOSED NEURAL NETWORKS MODEL AND STABILITY ANALYSIS

Equality constrained problem: Let us consider the following Equality-constrained Minimax Programming (EMMP) problem first:

$$\begin{aligned} & \min_x \max_y f(x, y) \\ & \text{s.t. } c(x, y) = 0 \end{aligned} \quad (2)$$

where, $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, $y = (y_1, y_2, \dots, y_r)^T \in \mathbb{R}^r$ and $f: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}$ and $c: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^m$ are given functions that are assumed to be first-order differentiable. Before detailed theoretic analysis, some definitions and evident propositions are given.

Definition 1: $\Omega: \mathbb{R}^n \times \mathbb{R}^r$ is the feasible region of EMMP, if $(x, y) \in \Omega \Rightarrow c(x, y) = 0$.

Definition 2: Ω' : $\mathbb{R}^n \times \mathbb{R}^r$ is the ϵ -feasible region of EMMP, if $(x, y) \in \Omega' \Rightarrow \|c(x, y)\| \leq \epsilon$, where $\epsilon > 0$ is a small enough positive real number.

Definition 3: (x^*, y^*) is the solution of EMMP, if $\forall (x, y) \in \Omega$, there is $f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*)$, that is (x^*, y^*) is a saddle point in Ω .

Proposition 1: If (x^*, y^*) is a saddle point in Ω , then:

$$\nabla_x f(x, y) \Big|_{(x^*, y^*)} = 0 \quad \text{and} \quad \nabla_y f(x, y) \Big|_{(x^*, y^*)} = 0 \quad (3)$$

where:

$$\nabla_x f(x, y) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

Present goal is to design a neural network that can settle down to an equilibrium point satisfying (3), whereas, what should primarily be considered are the constraints. In general, a penalty approach is employed. And consider the following equivalent expression of Eq. 2:

$$\begin{cases} \min_x L_1(x, y) = \min_x [f(x, y) + 0.5\sigma \|c(x, y)\|^2] \\ \min_y L_2(x, y) = \min_y [-f(x, y) + 0.5\sigma \|c(x, y)\|^2] \end{cases} \quad (4)$$

where, $\sigma > 0$. Thus, if L_1 can be minimized with x , it is equivalent to minimizing f with x , unless constraints are not satisfied which makes L_1 going away from minima. Similarly, minimizing L_2 with y is equivalent to maximizing $-f$, that is, maximizing f , unless constraints are not satisfied. Wah *et al.* (2000) have showed that the convergence speed and/or the solution accuracy could be affected by an additional weight in the objective function (Wah *et al.*, 2000). Thus, take into account an additional balance factor ω in the objective function as follows:

$$\begin{cases} L_1(x, y) = \omega \cdot f(x, y) + 0.5\sigma \|c(x, y)\|^2 \\ L_2(x, y) = -\omega \cdot f(x, y) + 0.5\sigma \|c(x, y)\|^2 \end{cases} \quad (5)$$

where, $\sigma > 0$ and ω is a tunable weight, e.g., we can adopt:

$$\omega = \begin{cases} 0.1 & \|c(x, y)\| > \epsilon \\ 10 & \|c(x, y)\| \leq \epsilon \end{cases} \quad (6)$$

where, $\epsilon > 0$ is a small positive real number, e.g., $\epsilon = \sigma^{-0.75}$. Take above instantiated ω and ϵ values as default values for later simulation if there is no additional annotation. Thus, the transient behaviors of the proposed neural network can be defined as follows:

$$\begin{cases} \frac{dx}{dt} = -\nabla_x L_1 = -\omega \nabla_x f(x, y) - \sigma \nabla_x c(x, y)^T \cdot c(x, y) \\ \frac{dy}{dt} = -\nabla_y L_2 = \omega \nabla_y f(x, y) - \sigma \nabla_y c(x, y)^T \cdot c(x, y) \end{cases} \quad (7)$$

Theorem 1: When $\|c(x, y)\| \leq \epsilon$, $\epsilon = \sigma^{-0.75}$, Eq. 5 nearly degenerates to an unconstrained minimax problem $\min_x \max_y [f(x, y)]$ and there exists:

$$\begin{cases} \frac{dx}{dt} = -\nabla_x L_1(x, y) = -\omega \nabla_x f(x, y) \\ \frac{dy}{dt} = -\nabla_y L_2(x, y) = \omega \nabla_y f(x, y) \end{cases} \quad (8)$$

This theorem obviously exists, since if $\epsilon = \sigma^{-0.75}$ and $\|c(x, y)\| \leq \epsilon$, then $0.5\sigma \|c(x, y)\|^2 \leq 0.5\sigma^{-0.5}$, thus $0.5\sigma \|c(x, y)\|^2$ is neglectable in Eq. 5 when σ is large enough. Note that $\omega = 10$ when $\|c(x, y)\| \leq \epsilon$, merely because the gradient and accelerate the convergence speed are supposed to augment. If the network is physically stable, the equilibrium point denoted by (x^*, y^*) that satisfies Eq. 3 and $c(x^*, y^*) = 0$ obviously also satisfies Eq. 7.

Theorem 2: Let the energy function of the neural network be $E_1(x, y) = L_1(x, y)$, $E_2(x, y) = L_2(x, y)$. If $\|c(x, y)\| > \epsilon$ and σ is sufficiently large, the energy functions decrease with x and y , respectively.

This theorem can be proved by checking if $\frac{\Delta E_1}{\Delta t} \leq 0$ and $\frac{\Delta E_2}{\Delta t} \leq 0$ or not. Since:

$$\frac{\Delta E_1}{\Delta t} = \nabla_x L_1(x, y)^T \cdot \frac{dx}{dt} + \nabla_y L_1(x, y)^T \cdot \frac{dy}{dt} \quad (9)$$

On the one hand, we have:

$$\nabla_x L_1(x, y)^T \cdot \frac{dx}{dt} = - \left\| \frac{dx}{dt} \right\|^2 \leq 0 \quad (10)$$

On the other hand, we have:

$$c(x, y) = \begin{bmatrix} h(x, y) \\ \min(g(x, y), 0) \end{bmatrix} \quad (17)$$

where, $c: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^m$, $h: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^m$ and $g: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^{m-m}$. By such disposition, all constrained conditions can be transferred into equality constraints.

SPECTRAL RADIUS ANALYSIS AND COMPARISON

The multilevel programming example given (Liu, 1998) can be modified as a good minimax programming example where $n = 2$ which can be described as follows:

$$\begin{aligned} \min_x \max_y f(x, y) &= -\frac{(x_1 + y_2)(x_2 + y_2)}{1 + x_1 y_1 + x_2 y_2} \\ \text{s.t.} \quad &\begin{cases} x_1^2 + x_2^2 \leq 100 \\ x_1 \geq 0, x_2 \geq 0 \\ 0 \leq y_1 < x_1 \\ 0 \leq y_2 \leq x_2 \end{cases} \end{aligned} \quad (18)$$

The solution given by GA (Liu, 1998) is $x^* = (7.0854, 7.0291)$, $y^* = (7.0854, 0.0000)$, with $f = -1.9760$. We have studied the transient behavior of the network from various initial states by simulating the dynamic equations via the classical fourth-order Runge-Kutta method (Atkinson *et al.*, 1994; Li *et al.*, 2010) on a Pentium III 500 Hz digital computer. We found that the total computation time varies with different start points. As a representation we choose $x = (80, 80)$ and $y = (80, 80)$ as our initial points that have a mean performance. The results are listed in Table 1 in comparison with the GA (Liu, 1998) (number of chromosomes 30, number of generations 300) and the Simplex-genetic Algorithm (SGA) (number of chromosomes 8, number of generations 100) (Longhua *et al.*, 2001; Jia-Liang *et al.*, 2010).

Performance of the proposed minimax neural network outperforms the evolution-based algorithms in shortening the computation time, because it is a gradient guided algorithm while GA and SGA are "blind search" algorithms to some degree. But the solution accuracy of minimax neural network is a bit inferior to that of the other two algorithms. Three reasons for this phenomenon may be: Firstly, there is truncated error in Runge-Kutta method

which is ineluctable; Secondly, to avoid becoming infinite, it is obliged to cease increasing at 10^6 which brings a lot of error; Thirdly, the gradient around the saddle point is very small and our algorithm stops somewhere near the equilibrium. The geometric shape of f when $x = (7.0711, 7.0711)$ with $y = (y_1, y_2)$ between 0 and 8 is shown in Fig. 1 and the surface is so flat that our algorithm could not proceed any more within the error limit which confirms our supposal.

The speed advantage of neural circuits over evolution strategies is more salient in large dimension optimization. GA has to spread initial seeds in the whole search space uniformly, so with the increase of variable dimension, the number of seeds increases exponentially. For example, if the 2-D plane are divided in grid with 9 equal parts per dimension and make every cross point be an initial chromosome, consequently we have about $(9+1)^2$ initial seeds. If the space is not a 2-D plane but a 3-D cube, the number is 10^3 . Thus when the variable dimension increases to N , the number of initial chromosomes needed is about 10^N . The more the chromosomes, the slower the algorithm converges. In contrast, neural networks start at an arbitrary point then follow gradient information searching every dimension, respectively, thus the computation assignment does not increase much.

However, the proposed neural network model is a purely gradient guided algorithm which could not escape

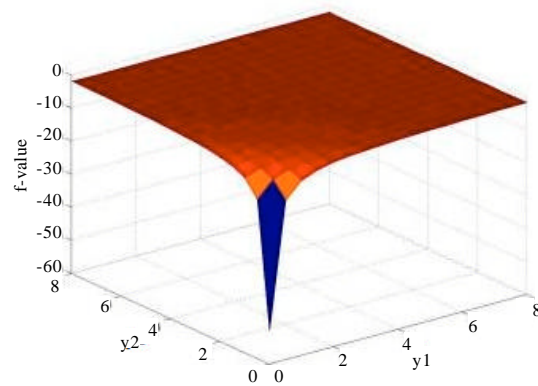


Fig. 1: Geometric information of f when $x = (7.0711, 7.0711)$ ($y = (y_1, y_2)$ are the vector when $n = 2$ in Eq. 2)

Table 1: Comparisons among genetic algorithm, simplex-genetic algorithm and the minimax neural network model scheme

Algorithm	Final solution	Objective function value	Execution time
GA (Liu, 1998)	$x = (7.0854, 7.0291)$ $y = (7.0854, 0.0000)$	-1.9760	About 28 m
SGA (Longhua <i>et al.</i> , 2001)	$x = (7.0711, 7.0711)$ $y = (7.0711, 0.0000)$	-1.9608	About 7 m
Minimax neural network model scheme	$x = (7.0711, 7.0711)$ $y = (7.0711, 7.0711)$	-1.9807	Less than 4 sec

$x = (x_1, x_2)$ and $y = (y_1, y_2)$ are vectors where $n = 2$

from the attraction of local minima of x or local maxima of y spontaneously. This algorithm is global stable but does not guarantee that global minima or maxima are bound to reach. Since some algorithms, such as simulated annealing, evolutionary algorithm, gradient descent with multi-starts (Wah *et al.*, 2000), perform well in global optimization, combination with neural network may be a way out. Nevertheless, how to combine them with only little sacrifice of speed is still an open topic.

CONCLUSIONS

In this letter, a novel kind of neural network for general minimax problems with both equality and inequality constraints has been constructed based on the penalty function approach. There is no explicit restriction imposed on the functional form. A Lyapunov function is established for the global stability analysis. Finally, simulation results show that the proposed model exhibits remarkable capability in solving general nonlinear minimax programming. In a word, the proposed minimax neural network is a powerful computing tool in minimax problem research.

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the financial support from the National Nature Science Foundation of China under grant No. 61070003 and 61071128; Zhejiang Provincial Natural Science Foundation of China under grant No. R1090052 and Y1101184; Project of Zhejiang Department of Education under the grant No. N20100690.

REFERENCES

- Atkinson, L.V., P.J. Harley and J.D. Hudson, 1994. Numerical Methods with Fortran 77: A Practical Introduction. Addison-Wesley, London, Pages: 395.
- Chao, H.C., C.S. Lin and B.C. Chieu, 2000. Minimax design of FIR digital all-pass filters. *IEEE Trans. Circuits Syst.*, 47: 576-580.
- Demyanov, V.F. and V.N. Malozemov, 1974. Introduction to Minimax. John Wiley and Sons, New York, USA., ISBN: 13-9780470208502, Pages: 307.
- Herrmann, J.W., 1999. A genetic algorithm for minimax optimization problems. Proceedings of the Congress on Evolutionary Computation, July 6-9, 1999, Washington, DC USA., pp: 1099-1103.
- Jia-Liang, G., Z. Hong-Xia and Z. Jin, 2010. Fuzzy logic based current control schemes for vector-controlled asynchronous motor drives. *Inform. Technol. J.*, 9: 1495-1499.
- Kalyanasundaram, S., J. Li, E.K.P. Chong and N.B. Shroff, 1999. Channel sharing scheme for packet-switched cellular networks. Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, March 21-25, 1999, New York, USA., pp: 609-616.
- Li, H., F.X. Yu, X.I. Zhou and H. Luo, 2010. Every connection routing under modified random waypoint models in delay tolerant mobile networks. *Inform. Technol. J.*, 9: 1686-1695.
- Liu, B.D., 1998. Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Comput. Math. Appl.*, 36: 79-89.
- Longhua, M., Z. Yongling and Q. Jixin, 2001. A new hybrid genetic algorithm for global minimax optimization. *Int. Conf. Info-tech Info-Net*, 4: 316-323.
- Tao, Q. and T.J. Fang, 2000. The neural network model for solving minimax problems with constraints. *Control Theory Appl.*, 17: 82-84.
- Wah, B.W., T. Wang, Y. Shang and Z. Wu, 2000. Improving the performance of weighted Lagrange-multiplier methods for nonlinear constrained optimization. *Inform. Sci.*, 124: 241-272.
- Ye, Z.Q., B.L. Zhang and C.X. Cao, 1997. Convergence analysis on minimax neural networks. *Inform. Control*, 26: 1-6.
- Yuan, Y.X., 1993. Numerical Methods for Nonlinear Programming. Shanghai Scientific and Technical Publishers, Shanghai, China.
- Zheng, Y.L., L.H. Ma and J.X. Qian, 2001. Parameter tuning method of robust PID controller. *Control Instrum. Chem. Ind.*, 28: 14-17.