A Neural Network Model for Quadratic Programming with Simple Upper and Lower Bounds and its Application to Linear Programming *

Xiang-sun ZHANG Hui-can ZHU

Institute of Applied Mathematics, Academia Sinica
Beijing 100080, China

Abstract. In this paper we put forward a neural network model for quadratic programming problems with simple upper and lower bounds and analyze the properties of solutions obtained by the model. It is shown that linear programming problems can be transferred into such quadratic programming problems and be solved by the model.

Key words: neural network, linear programming, quadratic programming, limit set

1 Introduction

The work carried out by Hopfield and Tank to solve travelling salesman problem(TSP)[1] initiated application of neural network(NN) in the field of optimization that includes both combinatorial optimization problems and optimization problems with continuous variables. Since then many NN models have been proposed on these subjects, in which linear programming(LP) problems and quadratic programming(QP) problems were dealt with the most for their fundamental importance. First we give an outline of some existing models on LP and QP [2-4].

Let the LP problem be

\[ \minimize \quad f(x) = a^T x \]

subject to \[ g(x) = Dx - b \leq 0 \]

where \( D \) is an \( m \times n \) matrix, \( b \in R^m \), and \( a, x \in R^n \).

The first neural network model for solving LP was proposed by Tank and Hopfield[2]. The dynamical system of the model can be written as

\[ \frac{dx}{dt} = s(- \nabla f(x) - \nabla g(x)g^+(x) - \frac{1}{s}x) \]  

* This work is supported partly by the National Natural Science Foundation of China(No. 101931209) and by the Hong Kong Baptist College(Proj. No. FRG /93-94/II-05).
where $g_i^+(x) = \max\{g_i(x), 0\}$, $g^+(x) = (g_1^+(x), \ldots, g_m^+(x))^T$, $s > 0$.

The importance of Tank and Hopfield model lies in that it began application of continuous neural network models in the optimization problems with continuous variable domain. The mechanism of the system is to trace out the equilibrium of (2). But as pointed out by Maa and Shanblatt in their paper [5], either the equilibrium equation

$$- \nabla f(x) - \nabla g(x)g^+(x) - \frac{1}{s}x = 0$$

has no equilibrium or if it does, the equilibrium may not be a solution to the problem (1). When the parameter $s$ is sufficiently large as suggested by Tank and Hopfield, (3) may be viewed as K-T condition for (1) in some sense [for K-T condition, see [11]]. So suppose there exists equilibriums for such sufficiently large $s$, the equilibrium will approximate the solution of (1) from outside of the feasible region of (1). But since no theory guarantees that the equilibrium will converge to the solution of (1) as $s$ becomes infinite, the model is not practicable even though it offers a significant advantage in hardware implementation.

Kennedy and Chua [3] modified the Tank and Hopfield model by using integrator in the hardware implementation. Their model can be described by

$$\frac{dx}{dt} = - \nabla f(x) - s \nabla g(x)g^+(x)$$

which has been used for solving both linear and quadratic programming problems. The dynamics of the model is aimed to minimize

$$E(x) = f(x) + \frac{s}{2} \sum_{i=1}^m (g_i^+(x))^2.$$  (5)

So it corresponds to the penalty function method in optimization theory. The intrinsic drawbacks of penalty function method on such aspects as how to choose the parameter $s$ to make proper convergence, and how to transfer the equilibrium point, which is infeasible, to the original solution, etc., also make the model not effective.

Consider the quadratic programming problem with simple upper and lower bounds such as

$$\begin{align*}
\text{minimize} & \quad E(x) \triangleq \frac{1}{2}x^TQx + q^Tx \\
\text{subject to} & \quad 0 \leq x_i \leq 1, \quad i = 1, \ldots, n
\end{align*}$$

where $Q = (q_{ij})_{n \times n}$ is an $n \times n$ symmetric matrix, $q = (q_1, \ldots, q_n)^T$, $x = (x_1, \ldots, x_n)^T \in R^n$. For simplicity, we call such a problem as simple quadratic programming (SQP) problem.

In [4], Bouzerdoum and Pattison proposed a network model for SQP which can be written as

$$\frac{dx}{dt} = q - CH(x) - Ax$$