DYNAMIC SOLVER FOR LINEAR OPTIMIZATION PROBLEMS

Yüksel Çakır Cüneyt Güzeliş
e-mail: yuksel@ehb.itu.edu.tr *e-mail: guzelis@deu.edu.tr*
IstanbulTechnical University, Faculty of Electrical and Electronics Engineering, Department of Electronics & Communication Engineering, 80626, Maslak, Istanbul, Turkey

Key words: Constrained optimization, gradient projection, dynamic solver

ABSTRACT

In this study, based on classical gradient projection method of optimisation theory, a dynamic solver for linearly constrained linear optimisation problems, called gradient projection network is introduced. To illustrate the performance of the network the result obtained for a special linear problem is compared to those obtained by Kennedy-Chua network.

I. INTRODUCTION

In the last two decades, beginning with that proposed by Hopfield, various neural network structures for optimization problems such as linear, quadratic and convex programming have been proposed [1, 2]. These structures, also called solvers, have the advantage of being implemented on analog hardware, thus producing solutions in real time. Compared with the traditional techniques which require iterative time consuming computations, analog solvers can be operated in parallel fasion.

In this work, first, methods for solving optimisation problems will be introduced. Thenanalog recurrent neural network structure, based on gradient projection method of optimisation will be introduced. In section IV simulation results and comparisons with other structures will be given.

II. NEURAL NETWORK STRUCTURES FOR SOLVİNG OPTİMİZATİON PROBLEMS

Following the work of Hopfield, various neural network structures [2] also called analog solvers has been proposed. Most of these depend on optimisation methods. In this section these methods in connection with analog solvers will be reviewed.

Analog solvers supply a continuous solutions while traditional iterative techniques give discrete point solutions. The dynamic solvers for optimization problem, in general, use energy descent dynamics, where the dynamics is constructed in a way that the objective function of optimisation problem named the energy decreases along trajectories and at the reached minimum point the system states give the solution to the problem. At such a state, the gradient of the energy function is zero, which is also a necessary condition for the state to be an extremum of the energy function. Systems of this type, named gradient systems, are constructed utilizing the concepts developed for stability analysis of a dynamical system introduced in 1892 by the Russian mathematician and engineer A. M. Liapunov and have the property of convergence if solutions are bounded [3]. Because of gradient nature of solvers, obtained solutions are local. In many optimization problems, especially combinatorial problems, time required for obtaining global solution grows exponentially with the size of problem [4, 5]. An approximate solution for such a problem is also meaningful. With gradient based techniques, a satisfactory but mostly not global solutions can be obtained in a reasonable time.

Constrained optimisation problems can also be handled with gradient based dynamics by reorganizing energy function to have an extra terms originating from constraints. For equality constrained problems, Lagrange multiplier technique is used. For inequality constrained problems, one way of transforming it to an unconstrained one is by adding a function of the constraints as a penalty term to the cost function. The method is named as a penalty function method and is classified as an exterior point method because the minimum is approached from the exterior of the feasible region. But, this method may give infeasible solutions. This is due to the difficulty in penalty parameter setting ensuring that obtained solutions satisfy the constraints in an exact way. Another way of converting a constrained problem to an unconstrained one is barrier function method where a term taking an infinitely large value at borders of the feasible region is added to cost. This is a kind of interior point methods meaning that points obtained by the method are always in the feasible region, and always give a legal solution. But the drawback of this method is that it is useless for a problem whose solutions are close to the border of feasible region. An alternative way to handle with constrained problems is the method of gradient projection. In this method, search for a minimum is done recursively by moving step by step towards direction defined by negative gradient of the cost function. In the case of a constraint violation, i.e., the gradient direction points to the infeasible region, so the calculated new point is not feasible, the projection of the gradient onto the violated constraint surface is performed and the search for a minimum is continued towards this projected gradient. A minimum is obtained when the projected gradient is zero, that can happen when the gradient of cost function is zero, i.e., the first order condition for a point to be a minimum is satisfied or when projection of the gradient is zero, meaning that there is not feasible direction to move. In the following section, gradient projection network based on this idea will be introduced.

III. GRADIENT PROJECTION NETWORK

The proposed gradient projection network utilizes the gradient projection concept and can be viewed as continuous time version of the iterative gradient projection method of optimisation explained in previous section. It is obtained for linear cost constrained with linear inequalities given as

$$\begin{aligned} &\min \Phi(x) = c^T x \\ &\textit{subject to } g(x) = Ax - b \le 0 \end{aligned} \qquad (1)$$

Where $A \in R^{m \times n}$, $b \in R^m$ and $g_i(x) = (Ax - b)_i$, $i \in M := \{1, 2, \dots, m\}$. The proposed dynamics is

$$\dot{x} = -P_{I_a}(x).\nabla E(x) \qquad (2)$$

Here $E(x)$ is a cost function $\Phi(x)$ considered as an energy function, and $P_{I_a}(x)$ is the projection matrix obtained as $P_{I_a}(x) = \left\lfloor I - G_{I_a}^T (G_{I_a} G_{I_a}^T) G_{I_a} \right\rfloor$ where matrix G_{I_a} is constructed depending on I_a which is an index set of constraints tending to be violated, defined as

$$I_a = \{ i \in M \,|\, g_i(x) = 0 \textit{ and } -\nabla g_i(x)^T.\nabla\Phi(x) \ge 0 \}$$

Now G_{I_a} is an $|I_a| \times n$ dimensional matrix whose $j(i)$'th row $(G_{I_a})_{j(i)}$ is defined as $(G_{I_a})_{j(i)} = (A)_i$. Here $j(i) = \{1, 2, \dots, |I_a|\}$ is an index set for renumbering the active constraints indexed by i.

Even though the right-hand side of proposed solver is discontinuous in x, as in [7] and [31] it is known that for any initial condition chosen in the feasible set there exists a unique solution which is continuous, nondifferentiable but right differentiable with respect to time and also kept in the feasible region.

Performance of proposed network is compared with penalty method based Kennedy-Chua solver defined in the form of

$$\dot{x} = -\nabla\Phi(x) - s\nabla g(x) g^+(x) \qquad (3)$$

where $g^+(x) := \{g_1^+(x) \dots g_m^+\}$ and $g_i^+(x) = \max\{g_i(x), 0\}$.

VI. PERFORMANCE COMPARISON

In this section, comparison of proposed neural network structure with that of Kennedy-Chua network will be given. Simulations are done in MATLAB as M-files . NNToolbox of MATLAB cannot be used for these applications since structures are novel for optimisation problems. The linear optimization problem considered here is a gasoline blending problem [1] formulated as

$$\min -8x_1 - 8x_2 - 5x_3 - 5x_4$$

$$\textit{subject to } g_1(x) = -5x_1 + 5x_2 \le 0$$

$$g_2(x) = 2x_1 - 3x_2 \le 0$$

$$g_3(x) = -x_1 - x_3 + 40 \le 0$$

$$g_4(x) = x_1 + x_3 - 40 \le 0$$

$$g_5(x) = -x_2 - x_4 + 60 \le 0$$

$$g_6(x) = x_2 + x_4 - 60 \le 0$$

$$g_7(x) = -x_1 \le 0$$

$$g_8(x) = -x_2 \le 0$$

$$g_9(x) = -x_3 \le 0$$

$$g_{10}(x) = -x_4 \le 0$$

By using Euler algorithm and step size of 0.01 the differential equation in (3) is transformed to difference equation

$$x_{k+1} = x_k + 0.01\left[-\nabla\Phi(x) - s\nabla g(x) g^{+}(x)\right]$$

Results given in Figure 1 and Figure 2 are obtained for $s = 1$, $s = 10$ and initial condition $x(0) = [0\ 0\ 0\ 0]^T$. It can be seen that as s increase, results converges to optimal solution $x^{*} = [40\ 40\ 0\ 20]^T$.

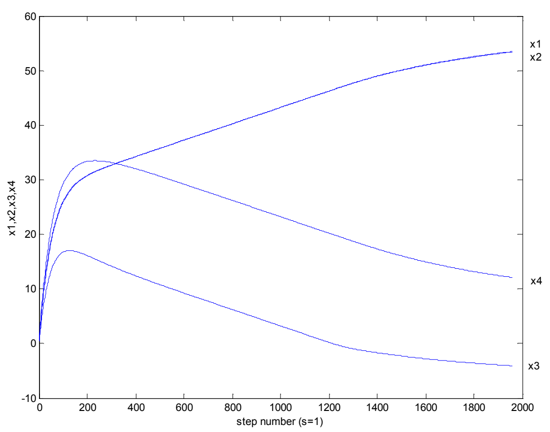

Fig1. Kennedy-Chua result for s=1.

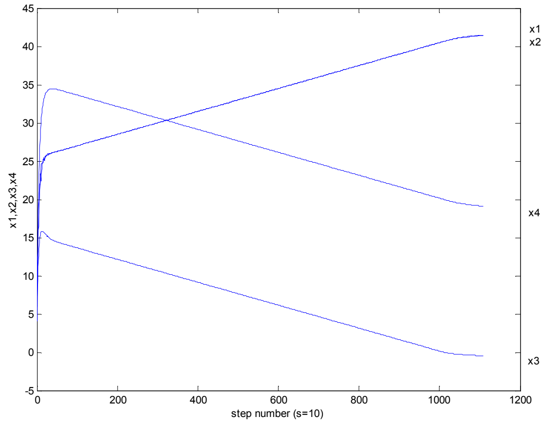

Fig2. Kennedy-Chua result for s=10.

Result obtained for gradient projection network is given in Figure 3. The difference equation corresponding to differential equation given in (2) is as

$$x_{k+1} = x_k + 0.01\left[P_{I_a}(x)\nabla\Phi(x)\right]$$

The solver requires initial condition to be feasible so initial condition taken as $x(0) = [0\ 0\ 0\ 0]^T$ is first projected to feasible region and new initial value is obtained as $x = [25\ 25\ 15\ 35]^T$. As can be seen, result obtained is an optimal solution.

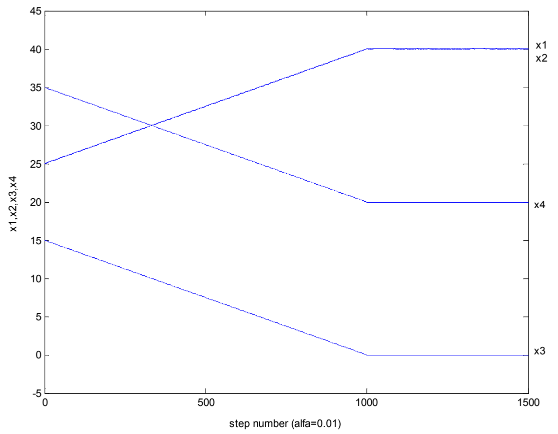

Fig3. Gradient projection network result.

V. CONCLUSION

In an iterative gradient projection method, on the boundaries, where at each step a new constraint becomes active, a jamming effect can occur. This is a result of step by step movement where the step size can not be chosen arbitrary small. In the proposed gradient projection network due to continuous nature of system, no such a problem arise.

Continuous solvers based on penalty method have the drawback to give unfeasible solutions in the case of improperly chosen penalty parameters. The proposed network does not require any parameter settings and for every initial condition, the trajectory converges to a stable equilibrium point satisfying necessary conditions for a point to be minimum.

The proposed network is general and discussions for linear case also hold on for linearly constrained quadratic problems.

ACKNOWLEDGMENT

We would like to thank N.Serap Şengör for her valuable contributions.

REFERENCES

1. Zak, S.H., Upatising, V. and Hui, S., Solving Linear Programming Problems with Neural Networks: A Comparative Study, IEEE Trans. Neural Networks, Vol.6, No.1, pp.94-103, January,1995.
2. Smith, K.A., Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research, INFORMS Journal on Computing, Vol.11, No.1, pp.15-34, Winter, 1999.
3. Hirsch, M.W. ve Smale, S., Differential Equations, Dynamical Systems, and Linear Algebra, Akademic Press, 1974.
4. Cichocki, A, ve Unbehauben, R., Neural Networks for Optimization and Signal Processing, John Wiley and Sons and B.G.Teubner, Stuttgart, 1993.
5. Vavasis, S.A., Nonlinear Optimization, Oxford University Press, New York, 1991.
6. Pekergin, F., Morğül, Ö. ve Güzeliş, C. A Saturated Linear Dynamical Network for Approximating Maximum Clique, IEEE Trans. on CAS Part-I, Vol.46, No.6, pp.677-685, June, 1999.
7. Hou, L. and Michel, A.N., Asymptotic stability of systems with saturation constraints, IEEE Trans. Automatic Control, Vol. 43, No.8.pp.1148-1154. August, 1998.