

Neural Networks Give a Warm Start to Linear Optimization Problems

MARTA I. VELAZCO , AURELIO R.L. OLIVEIRA AND CHRISTIANO LYRA , SENIOR MEMBER IEEE

School of Electrical Engineering and Computer Science
State University of Campinas (UNICAMP)
Av. Albert Einstein 400, Cidade Universitária, C P 6101
13083-970 Campinas, SP
BRAZIL

velazco@densis.fee.unicamp.br, chrllyra@densis.fee.unicamp.br

Institute of Mathematical and Computer Sciences
State University of São Paulo (USP)
Av. do Trabalhador São-Carlense 400
13560-970 São Carlos, SP
BRAZIL

aurelio@icmc.sc.usp.br

Abstract - Hopfield neural networks and interior point methods are used in an integrated way to solve linear optimization problems. The neural network unveils a warm starting point for the primal-dual interior point method. This approach was applied to a set of real world linear programming problems. Results from a pure primal-dual algorithm provide a yardstick. The integrated approach provided promising results, indicating that there might be a place for neural networks in the “real game” of optimization.

Index Terms - Hopfield networks, interior point methods, neural networks, linear programming.

I. INTRODUCTION

Tank and Hopfield [1] pioneered the use of neural networks to solve linear optimization problems, with Hopfield Networks [2]. Their results were extended by Kennedy and Chua [3] to non-linear optimization problems. Later, Xia and Wang [4], proposed a global convergent neural network for solving linear optimization problems. Recently Silva [5] coupled fuzzy logic with Hopfield networks to solve linear and non-linear optimization problems. All these approaches formulate an energy function to represent the original problem and use the gradient method [6] to minimize it.

Problems used to illustrate them are all very small; they are textbook examples that can be solved by hand calculations.

Indeed, it seems that the neural networks and optimization communities carry on their research in different habitats, with tiny links between each other. In one side, researchers in neural networks provided theoretical results for addressing optimization problems but did not go much beyond demonstrative examples. On the other side, the optimization community did not show to believe that neural networks could help to solve real-life optimization problems.

This paper unveils benefits from cross-fertilization between neural networks and optimization ideas. It proposes an approach to linear programming (i.e., linear optimization problems) where the Hopfield network developed by Silva [5] is improved to go half-way in the optimization process, where it passes the baton to a primal-dual interior point method [7,8]. Neural networks are used to build feasible points close to the optimal

solution. These points are used as warm start for primal-dual methods. Application to a set of real-life linear programming problems gives guidelines on the possibilities of the approach.

This paper is organized into six sections. Section II introduces the primal-dual interior point method for linear optimization. Section III reviews the Hopfield approach to solve optimization problems. Improvements in the Hopfield approach are presented in Section IV. The new approach is described in Section V. Case studies are described in Section VI. Conclusions follow.

II. PRIMAL-DUAL INTERIOR POINT METHOD

For a long time the simplex method [6] was considered the only practical approach for solving large linear optimization problems. By 1985 the interior point method of Karmarkar [7] stepped in the optimization scene, with polynomial time properties and claims of better practical results than simplex. Few years later new methods developed under interior point basis supported fecundity of these grounds (e.g., [9]); interior point implementations could behave better than simplex. Many variants of these methods have been appeared. Among them, one of the most successful variants is known as the primal-dual method [8].

Consider the primal linear optimization problem in the standard form:

$$\begin{aligned} \min \quad & c^T \cdot x \\ \text{subject to} \quad & A \cdot x = b \\ & x \geq 0 \end{aligned} \quad (1)$$

where $A \in \mathcal{R}^{m \times n}$.

Associated to problem (1), there is the dual problem represented by:

$$\begin{aligned} \max \quad & b^T \cdot y \\ \text{subject to} \quad & A^T \cdot y + z = c \\ & z \geq 0 \end{aligned} \quad (2)$$

The optimality conditions for primal and dual problems are given by primal and dual feasibility together with the *complementarity conditions*:

$$X \cdot Z \cdot e = 0 \quad (3)$$

where the notation $X = \text{diag}(x)$ for diagonal matrices formed from vectors is introduced and e is the vector of all ones of appropriated dimension.

The majority of primal-dual interior point methods can be seen to be variants of the Newton method applied to the optimality conditions. The following outlines a framework for such methods.

Given y^0 and $(x^0, z^0) > 0$ (an interior point)

For $k = 0, 1, 2, \dots$, Do

1) Choose $\mathbf{s}^k \in [0, 1)$ and set $\mathbf{m}^k = \mathbf{s}^k \cdot \left(\frac{\mathbf{g}^k}{n} \right)$ where,

$$\mathbf{g}^k = (x^k)' \cdot z^k \text{ and } n \text{ is the size of } x.$$

2) Compute the Newton search directions, $(\mathbf{D}_x^k, \mathbf{D}_y^k, \mathbf{D}_z^k)$.

3) Choose an appropriate step length to remain interior, $\mathbf{a}^k = \min(t^k \mathbf{r}_p^k, t^k \mathbf{r}_d^k)$, where $t^k \in (0, 1)$,

$$\mathbf{r}_p^k = \frac{-1}{\min_j \left(\frac{\mathbf{D}_{x_i}^k}{x_j^k} \right)} \text{ and } \mathbf{r}_d^k = \frac{-1}{\min_j \left(\frac{\mathbf{D}_{z_i}^k}{z_j^k} \right)}.$$

4) Compute the new interior point, $(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + \mathbf{a}^k \cdot (\mathbf{D}_x^k, \mathbf{D}_y^k, \mathbf{D}_z^k)$.

The point (x, y, z) is interior if there are not binding variables (i.e., $(x, z) > 0$). The step length (\mathbf{a}^k) computed in step 3 keeps such property.

Next section presents a Hopfield neural network approach to solve optimization problems.

III. A REVIEW OF HOPFIELD APPROACH TO LINEAR OPTIMIZATION

The Modified Hopfield Network (MHN) adopted by Silva [5] is illustrated in Fig.1, where each step is represented by a box.

The MHN is a cycle of two phases: a feasibility process and an updating one via search direction. The process stops when it reaches a minimum. In the feasibility process the vector x is updated to compute a feasible point. The updating process computes a new vector x

from the search direction and the step length. Let us discuss it with some more details.

I. Feasibility. It is made iteratively in a cycle of two steps:

1) Projection in valid subspace [10]. The new feasible point, x^p , is computed using

$$x^p = T \cdot x + s \quad (4)$$

where T is a projection matrix that represents the connection among the neurons and is computed from the constraints of the original optimization problem; the vector s is orthogonal to the matrix and is computed, in a similar way, from the constraints of the problem [8,10].

2) Application of the activation function ($g(x)$); the point is constrained over a feasible hypercube,

$$g(x) = \begin{cases} 0 & x \leq 0 \\ x & 0 < x < u \\ u & x > u \end{cases} \quad (5)$$

where u is the upper bound for variable x .

II. Updating by search direction. The feasible variable x is updated as follows:

1) A fuzzy controller computes the step \mathbf{a}_k ;

2) The search direction d^l is computed;

3) The vector x is updated,

$$x^{l+1} = x^l + \mathbf{a}^l \cdot d^l \quad (6)$$

The search direction is computed from an energy function associated to the original problem. For linear problems the search direction at each iteration is fixed ($d^l = c$).

The convergence of the MHN is very slow, even for small instances that do not represent real-life linear optimization problems. Three drawbacks can be identified in this methodology:

- 1) The search direction, $d^l = c$, is fixed - it should be updated with information from each new point.
- 2) The step size is obtained by a cut-and-try method in the fuzzy controller, what leads to a high computational cost.
- 3) The floating-point operations to carry on the projection process get too expensive as the size of the problem grows.

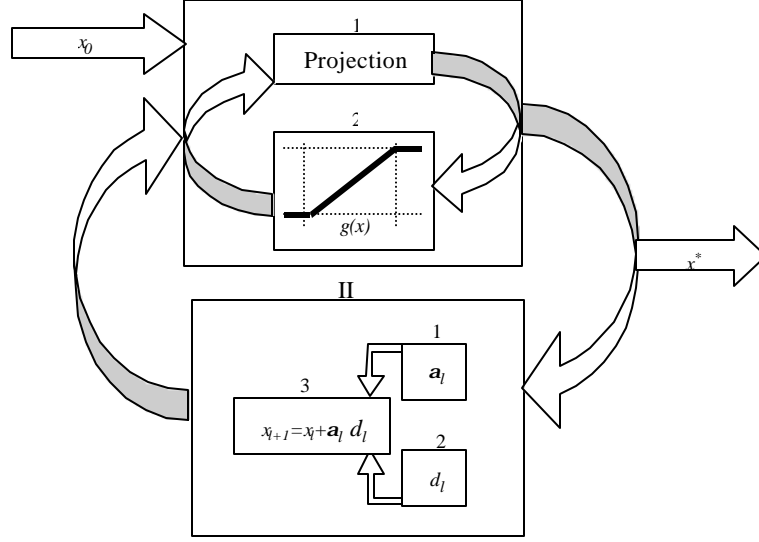


Figure 1. Modified Hopfield Network

IV. IMPROVEMENTS IN THE MODIFIED HOPFIELD APPROACH

The MHN and the primal-dual interior point method are used in an integrated way to solve linear optimization problems. The MHN is used to obtain an “intelligent” initial point for the primal-dual interior point method.

A few modifications are introduced in the MHN to enhance the warm start process. Namely,

- 1) the feasibility process is modified to produce a feasible interior point;
- 2) the projection process is enhanced;
- 3) the MHN is tailored to provide a feasible dual interior point;
- 4) a fixed step is used and, thus the fuzzy controller is eliminated;
- 5) search directions are computed by an interior point method.

The first three points will be further discussed.

1) The Feasibility Process

The feasibility cycle is modified to obtain points inside the feasible hypercube ($0 \leq x \leq u$) and a tolerance is introduced.

In this work, the upper bound u is ∞ . Therefore we have only a tolerance for the lower bound, named *lower interior tolerance* (It). It is used by the activation function ($g_{It}(x)$) of box I (Fig.1) as follows.

$$g_{It}(x) = \begin{cases} It & x < It \\ x & x \geq It \end{cases} \quad (7)$$

This new function ($g_{It}(x)$) modifies step I of the MHN. Its choice leads to a point that is both feasible and interior.

2) Improving Efficiency in the Projection Process

The matrix T and vector s for primal linear optimization problems are computed by the following equations.

$$T = I - A^T \cdot (A \cdot A^T)^{-1} \cdot A \quad (8)$$

$$s = A^T \cdot (A \cdot A^T)^{-1} \cdot b \quad (9)$$

where I is the identity matrix and A and b are given by (1).

Matrix T is never computed - it has few elements different of zero and large number of floating point operation are made when it is computed.

The projected vector x^p is computed using Equations (4), (8) and (9),

$$x^p = (I - A^T \cdot (A \cdot A^T)^{-1} \cdot A) \cdot x + A^T \cdot (A \cdot A^T)^{-1} \cdot b \quad (10)$$

which can be written as,

$$x^p = x + A^T \cdot (P_R^{-1} \cdot (P_R^{T-1} \cdot (b - A \cdot x))) \quad (11)$$

where, the Cholesky factorization of $A \cdot A^T$, P_R is computed only once, before starting the iterating process.

$$P_R \cdot P_R^T = A \cdot A^T \quad (12)$$

3) Computing a Feasible Dual Interior Point

For the dual problem, the matrix T and the vector s are computed by the following equations:

$$T = I - A \cdot (A^T \cdot A)^{-1} \cdot A^T, \quad (13)$$

$$s = A \cdot (A^T \cdot A)^{-1} \cdot c \quad (14)$$

Variables y and z need to be projected. New projection equations are computed using equations (4), (13) and (14).

$$y^p = y - A^T \cdot A \cdot y + A^T \cdot G - A^T \cdot z + A^T \cdot c - A^T \cdot w, \quad (15)$$

$$z^p = G - A \cdot y + c - w, \quad (16)$$

where G and w are computed by the following Equations.

$$G = A \cdot \left(D_R^{-1} \cdot \left(D_R^{T-1} \cdot \left(A^T \cdot (A \cdot y + z) \right) \right) \right), \quad (17)$$

$$D_R \cdot D_R^T = A^T \cdot A + I_m, \quad (18)$$

$$w = A \cdot \left(D_R^{-1} \cdot \left(D_R^{T-1} \cdot A^T \cdot c \right) \right). \quad (19)$$

Equations (18) and (19) are also computed only once, before starting the iterative process.

V. WARM START WITH NEURAL NETWORKS

The primal-dual interior point methods are used in an integrated way with the MHN. We will call it the Hopfield-primal-dual method.

The methodology can be outlined as follows.

- 1) The feasibility process described in last section gives an initial interior point x_0 .
- 2) A “warm point” x_w is computed with a fixed number of the following cycle of steps.
 - I. Compute a primal-dual interior point search direction, d^l .
 - II. Compute a new point in the search direction d^l , using a fixed step size $\mathbf{a}^l = I$.
 - III. Use again the feasibility process to convert the new point into a feasible interior point with the feasibility process - the point computed in the previous step is not an interior point.
- 3) A primal-dual interior point method takes over the optimization process, from the warm point x_w .

It can be argued that the full step size $\mathbf{a}^l = I$ is the best step size. Three points may be invoked: (1) it leads to a feasible point with respect to the equality constraints; (2) there is no reason for smaller step size, which would lead to both smaller improvement and to unfeasibility with

respect to the equality constraints; (3) a larger step size would also lead to unfeasibility (also, it would be to trust too much on “luck”, since the quadratic convergence properties of Newton method are proved for the full step size).

It should be remarked that, for the primal-dual interior point method, the initial point x_0 and the point x^l computed at each iteration do not need to be feasible. Nevertheless when a feasible point is computed the number of iterations is reduced (compared to cases when only an interior point is computed).

VI. CASE STUDIES

This section presents a set of comparative test between the approach defined in the previous section and the corresponding interior point method. The implementation of both methods was made in MATLAB.

The set of problems comes from Netlib [11] (<http://www.netlib.org>). This set of real-life for linear programming problems is widely adopted for comparison purposes.

Three parameters are used to computed the warm initial point:

- 1) k - the number of iterations of the Hopfield-interior-point-method to obtain the warm initial point. Its value was set to 3.
- 2) T - tolerance for convergence of feasibility process. Its value was set to 10^{-4} .
- 3) It - lower interior tolerance used by the activation function ($g_{it}(x)$). A good value was found for each case.

The classic primal-dual method and the Hopfield primal-dual approach (which includes the warn initialization) were applied to a subset of the Netlib collection of problems. Table I presents a summary of results.

We can observe in Table I that the Hopfield primal-dual approach always reached a solution with a smaller number of iterations than the classic primal-dual method. This feature shows that the new methodology is able to uncover a better path to optimality.

On the other side, the improvement in the path to optimality did not lead to better overall times for most of the problems. However, there is a general trend of better processing times as the size of the problems increase.

TABLE I. RESULTS FOR NETLIB LINEAR PROGRAMMING PROBLEMS

PROBLEMS (mxn)	PRIMAL -DUAL ITERATIONS	HOPFIELD PRIMAL - DUAL ITERATIONS	PRIMAL -DUAL TIME (seconds)	HOPFIELD PRIMAL - DUAL TIME (seconds)
STOCFOR1 (128x184)	21	14	4.26	4379.0
BLEND (74x114)	19	14	2.21	184.49
SHARE2B (96x162)	21	12	2.82	104.4
SCI05 (104x162)	15	9	1.78	32.14
ADLITTLE (55x137)	20	10	1.56	205.71
SCAGR7 (128x184)	19	16	2.26	176.2
SHIP12s (466x2293)	48	26	52.99	76.53
25FV47 (798x1854)	46	44	206.07	243.20
SCSD8 (397x2750)	18	11	4.281	22.80
SCFXM2 (644x1184)	40	38	12.254	25.134
FFFFF800 (501x1005)	127	54	373.81	135.01

VII. CONCLUSIONS

This paper uncovered benefits from cross-fertilization between neural networks and optimization ideas to solve linear programming problems. A Hopfield neural network approach is used to go part of the way in the optimization process, where it passes the baton to a primal-dual interior point method. In other words, Hopfield networks are used to build feasible points close to the optimal solution; these points are used as warm start for primal-dual methods.

Application of these new ideas to a set of real-life linear programming problems from Netlib indicates that neural networks can have a place beyond demonstrative examples in the optimization arena.

In all case studies, the Hopfield primal-dual interior point approach reduced the number of iterations to obtain an optimal solution.

The overall processing time improved as the size of the problems increased - in one of the large problems solved, the Hopfield primal-dual approach was almost three time faster than the classic primal-dual method.

We are not yet ready to state that the ideas presented here will provide a better way to solve large optimization problems. However, we can affirm that mutual benefits will be achieved if the neural network and optimization communities strengthen their links.

ACKNOWLEDGMENTS

The authors thank the supports of FAPESP (Research Supporting Foundation of the state of São Paulo) and CNPq (Brazilian Council for the Development of Science and Technology).

REFERENCES

- [1] D. W. Tank and J. J. Hopfield, "Simple 'Neural' Optimization Network: an A/D Converter, Signal Decision Circuit and a Linear Programming Circuit", *IEEE Trans. Circuits and Systems*, CAS-33, pp. 533-541, 1986.
- [2] S. U. Haykin, "Neural Network: A comprehensive Foundation", Second Edition, Prentice Hall Inc., Upper Saddle River, New Jersey, USA, 1999.
- [3] M. P. Kennedy and L. O. Chua, "Neural Networks for Nonlinear Programming", *IEEE Trans. On Circuits and Systems*, Vol. 35:5, pp. 554-562, 1988.
- [4] Y-S. Xia and J. Wang, "A General Methodology for Designing Globally Convergent Optimization Neural Networks", *IEEE Trans. Neural Networks*, Vol. 9:6, pp. 1331-1343, 1998.
- [5] I. N. da Silva, "A Neuro-Fuzzy Approach to Systems Optimization and Robust Estimation", Ph.D. thesis, School of Electrical Engineering and Computer Science, State University of Campinas, Brazil, December, 1997 (in Portuguese)
- [6] D. Luenberger, "Linear and Nonlinear Programming", Addison-Wesley, Reading, Massachusetts, 1984.
- [7] N. Karmarkar, "A New Polynomial-Time Algorithm for Linear Programming", *Combinatorica*, Vol. 4, pp.373-395, 1984.
- [8] S. J. Wright, "Primal-Dual Interior-Point Methods", SIAM Publications, SIAM, Philadelphia, PA, USA, 1996.
- [9] I. Adler, M. G. C. Resende, G. Veiga and N. Karmarkar, "An Implementation of Karmarkar's Algorithm for Linear

- Programming”, *Mathematical Programming*, Vol. 44, pp. 297—335, 1989.
- [10] S. V. B. Aiyer, M. Niranja and F. Fallside, “A Theoretical Investigation into the Performance of the Hopfield Model”, *IEEE Trans. Neural Networks*, Vol. 1:2, pp. 204-215, 1990.
- [11] D. M. Gay. “Electronic Mail Distribution of Linear Programming Test Problems”. *Mathematical Programming Society COAL Newsletter*, No. 13, pp.10 -12, Dec., 1985.