National Seminar on Control and Optimization
سومین سمینارملی
کنترل وبهینه سازی

A NEURODYNAMIC MODEL TO SOLVE THE MAXIMUM FLOW PROBLEM

MOHAMMAD ESHAGHNEZHAD[1]*, SOHRAB EFFATI[1], AND FREYDOON RAHBARNIA[1],

[1] *Department of Applied Mathematics, Faculty of Mathematical Sciences,*
Ferdowsi University of Mashhad, Mashhad, Iran.
m_shaghnezhad@yahoo.com; s-effati@um.ac.ir; rahbarnia@um.ac.ir

Abstract. In the present brief, we are going to obtain the solution of the maximum flow problem (MFP). Here, by reformulating the original problem to a linear optimization problem, the Karush-Kuhn-Tucker (KKT) optimality conditions are suggested as a new strategy for the problem. The KKT conditions are used to propose the RNN model. An illustrative example is given to demonstrate the performance of this approach.

1. Introduction

In graph theory, a flow network is defined as a directed graph involving a source (S) and a sink (T) and several other nodes connected with edges. Each edge has an individual capacity which is the maximum limit of flow that edge could allow. The constrained maximum flow problem (MFP) is to send the maximum flow from a source to a sink in a directed capacitated network where each arc has a cost and the total cost of the flow cannot exceed a budget. The MFP is one of several well known basic problems for combinatorial optimization in weighted directed graphs. Because of its importance in many areas of applications such as computer science, engineering and operations research, the MFP has been extensively studied by many researchers using a variety of methods (see [1, 2]). Recently, neural networks to solve some programming problems have been extensively studied and many important results have also been obtained such as [3, 6, 4, 5].

2. Problem formulation and optimality conditions

In directed network $G = (V, E)$, each arc is denoted by an ordered pair (i, j), where $i, j \in V$. Note that, there is only one directed arc (i, j) from i to j. Also, node 1 is the source node and assume that s is the destination node. We associate with each arc (i, j), a lower bound on flow of $l_{ij} = 0$ and an upper bound on flow of u_{ij} which they are arc capacities and finite integers. The

Key words and phrases. Maximum flow problem, Recurrent Neural Networks, Lyapunov stability.
* Speaker.

MFP can be stated as the following program:

$$\begin{aligned} \max \quad & f, \\ s.t. \quad & \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} f, & i = 1, \\ 0, & i \neq 1 \text{ or } n, \\ -f, & i = n, \end{cases} \\ & 0 \leq x_{ij} \leq u_{ij}, \quad i, j = 1, 2, \dots, n. \end{aligned} \tag{2.1}$$

As a matter of fact, $f \geq 0$ and also, the upper bound of f is equal to the summation of all arc capacities; i. e., $0 \leq f \leq \sum_{i=1}^{m} u_i$. By defining $e_n = (0, 0, \dots, 0, 1)^T, e_1 = (1, 0, \dots, 0, 0)^T \in \mathbb{R}^m$ (i. e., $(e_n - e_1)$ is the activity vector for f) the problem is reduced to:

$$\begin{aligned} \max \quad & f \\ s.t. \quad & (e_n - e_1)f + Ax = 0 \\ & 0 \leq f \leq \sum_{i=1}^{m} u_i \\ & 0 \leq x \leq u. \end{aligned} \tag{2.2}$$

Here, assume that, $f = x_{m+1}$ and $u_{m+1} = \sum_{i=1}^{m} u_i$. Also, we define $c = (0, 0, \dots, 0, 1)^T \in \mathbb{R}^{(m+1)\times 1}$, $X = (x_1, x_2, \dots, x_{m+1})^T \in \mathbb{R}^{(m+1)\times 1}$, $U = (u_1, u_2, \dots, u_{m+1})^T \in \mathbb{R}^{(m+1)\times 1}$, $q = (-1, 0, \dots, 0, 1)^T \in \mathbb{R}^{n\times 1}$, and $B = [q|A] \in \mathbb{R}^{n\times(m+1)}$, so the problem is stated as follows:

$$\begin{aligned} \max \quad & c^T X \\ s.t. \quad & BX = 0 \\ & 0 \leq X \leq U. \end{aligned} \tag{2.3}$$

The KKT optimality conditions for (2.3) can be obtained as follow:

$$\begin{aligned} (-c + B^T y + h)X = 0 \qquad & -c + B^T y + h \quad \geq 0 \\ -h^T(X - U) = 0 \qquad & -(X - U) \quad \geq 0 \\ BX = 0 \qquad & X, h \quad \geq 0, y \text{ is free}, \end{aligned} \tag{2.4}$$

where h and y are the Lagrange multipliers. Next theorem summarise this section.

Theorem 2.1. *The optimal solution of the MFP* (2.1) *is equal to the solution of the KKT conditions* (2.4).

3. Recurrent Neural Network Model

We define $F(w)$ and w as follow:

$$F(w) = \begin{bmatrix} -c + h + B^T y \\ -(X - U) \\ BX \end{bmatrix}, \qquad w = \begin{bmatrix} X \\ h \\ y \end{bmatrix}.$$

Also, it can be restated as follow:

$$\begin{cases} F_i = (-c + h + B^T y)_i, & w_i = X_i, \quad i = 1, 2, \ldots, m, \\ F_{i+m} = -(X - U)_i, & w_i = h_i, \quad i = 1, 2, \ldots, m, \\ F_{j+2m} = BX, & j = 2m + 1, \ldots, 2m + n. \end{cases} \tag{3.1}$$

Thus, based on (2.4) we can restated the KKT optimality conditions as follow:

$$\begin{cases} F_i w_i = 0, \quad F_i \geq 0, \quad w_i \geq 0, \quad i = 1, 2, \ldots, 2m, \\ F_j = 0, \qquad j = 2m + 1, \ldots, 2m + n. \end{cases} \tag{3.2}$$

The perturbed min function is formulated as:

$$\phi^{\varepsilon}_{\text{min}}(a, b) = \frac{a + b - |a - b + \varepsilon|}{2}, \qquad \varepsilon \to 0.$$

Here, by applying the definition of NCP function, based on (3.2) we can easily verify that the KKT optimality conditions (2.4) are equivalent to the following unconstrained minimization problem:

$$\min \quad \Psi(z) = \frac{1}{2}\|\Phi(w)\|^2. \tag{3.3}$$

where $w = (X, h, y)^T$ and

$$\Phi(w) = \begin{bmatrix} \phi^{\varepsilon}_{\text{min}}(X, -c + h + B^T y) \\ \phi^{\varepsilon}_{\text{min}}(h, -(X - U)) \\ BX \end{bmatrix}.$$

Theorem 3.1. *X^* is the KKT point of (2.4), if and only if, $w^* = (X^*, h^*, y^*)^T$ satisfies the equation $\Phi(w^*) = \mathbf{0}$ for every $\varepsilon \to 0$.*

The RNN model associated with (2.1) can be described as follow:

$$\frac{dw}{dt} = -\gamma \nabla \Psi(w), \qquad w(t_0) = w_0, \tag{3.4}$$

where $\gamma > 0$ is the convergence rate.

4. Convergence Analysis

Here, we study the stability analysis of the proposed model (3.4).

Theorem 4.1. *$w^* = (X^*, h^*, y^*)^T$ is the equilibrium point of RNN (3.4), if and only if, satisfies the KKT conditions (2.4).*

Lemma 4.2. *Right hand side of (3.4) is Lipschitzian.*

Theorem 4.3. *The proposed model (3.4) is globally stable in the sense of Lyapunov.*

5. An Example

Consider the maximum flow problem for the network flow in Fig. 1. All simulation results show that the output trajectory f of the proposed model converges to the optimal solution $f^* = 14$. For instance, the Fig. 2 depicts the convergence behaviour of the trajectories obtained from the recurrent neural network (3.4) for many random initial points.

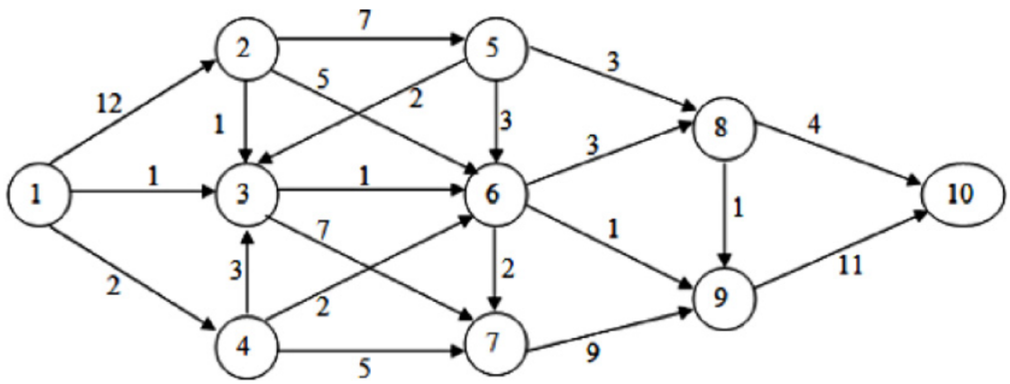

FIGURE 1. The network in Example.

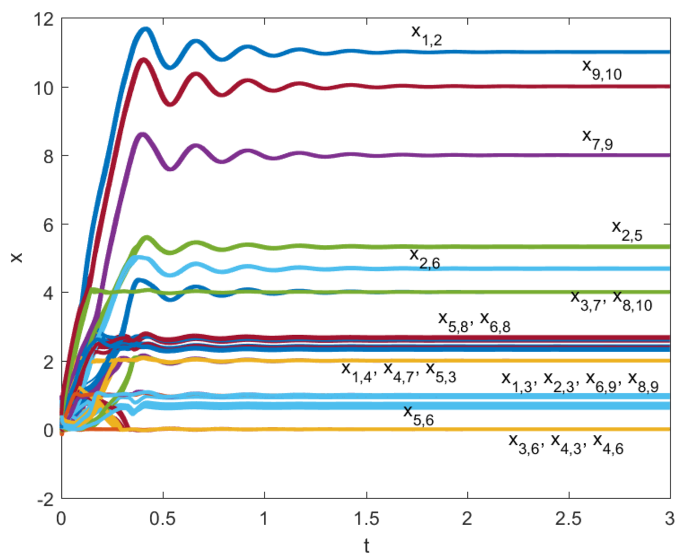

FIGURE 2. Transient behaviour of trajectories obtained from RNN (3.4).

REFERENCES

1. A. K. Ahuja, J. B. Orlin, A capacity scaling algorithm for the constrained maximum flow problem, Networks 25 (1995) 89-98.
2. C. Caliskan, A double scaling algorithm for the constrained maximum flow problem, Computers and Operations Research 35 (2008) 1138-1150.
3. S. Effati, A. Mansoori, M. Eshaghnezhad, An efficient projection neural network for solving bilinear programming problems, Neurocomputing 168 (2015) 1188-1197.
4. M. Eshaghnezhad, S. Effati, A. Mansoori, A Neurodynamic Model to Solve Nonlinear Pseudo-Monotone Projection Equation and Its Applications, IEEE Transactions on Cybernetics 47 (2017) 3050-3062.
5. A. Mansoori, M. Erfanian, A dynamic model to solve the absolute value equations, Journal of Computational and Applied Mathematics 333 (2018) 28-35.
6. A. Mansoori, M. Eshaghnezhad, S. Effati, Recurrent neural network model: a new strategy to solve fuzzy matrix games, IEEE Transactions on Neural Networks and Learning Systems, 30 (8) (2019) 2538-2547.