CompAMa
www.compama.co.usb.ve

Bull. Comput. Appl. Math., Vol.1, No.1, 2013
ISSN 2244-8659

GNU Oflox: an academic software for the minimal cost network flow problem

Andrés M. Sajo–Castelli [*], *Bernardo Feijoo* [†]

CompAMa Vol.1, No.1, pp.73-77, 2013 - Accepted July 7, 2013

Abstract

We present an open-source software package written for GNU Octave. The software is an implementation of the Simplex algorithm for the minimal cost network flow problem oriented towards the academic environment. The implementation supports the use of Big-M and Phase I/Phase II methods and it can also start from a given feasible solution. Flexibility of the package's output configuration provides many attractive possibilities. The outputs are plain editable LaTeX files that can be modified and orchestrated to fit most academic needs. It can be used in examination materials, homework assignments or even form part of a project. The format used to describe the network is the DIMACS min file format to which a simple extension was added in order to support the description of feasible trees in the file.

Keywords: Educational software, minimal cost network problem, simplex algorithm, GNU Octave, Matlab.

[*]Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar, Venezuela, e-mail: asajo@usb.ve

[†]Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar, Venezuela, e-mail: bfeijoo@usb.ve

1 Introduction

Network flow is one of the most important branches of optimization not only because of the large variety of real-life problems that can be modeled but mainly because they can be solved very efficiently. The minimal cost network flow (MCF) problem has received a lot of attention partly because advances in processing capacity makes it possible to solve larger and larger problems. One of the dominant algorithms for solving the MCF problem is the Simplex [3] for MCF algorithm [1]. A solid understanding of it's internals is paramount in order for the next generation of professional to develop on the subject. This software aims at facilitating this purpose.

Educational software for the undergraduate student has been in the rise since the popularization of higher level languages such as Matlab® [5], GNU Octave [7], Mathematica®, Scilab, etc. Specially with the advent of tutorials and demos that can be played-back and manipulated inside these softwares. The possibility of extending the original software with libraries and packages has accelerated the adoption of such softwares.

We present an open-source software package written in Octave/Matlab® language for GNU Octave. The software is an implementation of the Simplex algorithm for the minimal cost network flow problem with an orientation towards academic uses. The main purpose is to show step by step the tasks undertaken by this Simplex algorithm while solving a problem.

The problem of minimal cost network flow problem belongs to the network flow domain and can be described briefly as follows. Given the network $G = (N, A)$ where N and A are the sets of nodes and arcs respectively. We seek to minimize

$$\sum_{(s,e)\in A} \text{flow}(s, e) \times \text{cost}(s, e), \tag{1}$$

considering the associated arc capacity constrains along with the assumption of node flow conservation. It is well known that this problem can be solved with a Simplex MCF algorithm. The maximization counterpart can be trivially solved by minimizing the negative of expression (1). Given the direct mapping between the minimization of the network flow problem and the assignation problem, this last can also be solved by GNU Oflox using a simple map.

2 About the package

The software package is an implementation of the Simplex MCF algorithm with a variety of selectable options which define how the algorithm will perform its tasks. Due to the nature of this software the configuration options are separated in two groups: parameters related to the way the problem is solved and parameters that control the graphical output. We begin by describing the options related to the algorithm, followed by the graphical output options.

In order to solve problems without a feasible starting point, GNU Oflox has implemented both the Big-M and Phase I/Phase II methods. But it can start from a feasible solution given in advance. It is also possible to stop the Simplex algorithm at a given iteration and retrieve all the information regarding that state. This is quite an attractive option for writing examinations or for experimenting with new variations of problems, among other alternatives. In the event of multiple choices for in-bound or out-bound arcs, Dantzig's rule is used, but implementing other rules is quite simple.

One of the main features of this package is the graphical output, when enabled it produces a page summarizing the problem to be solved: graph, arc bounds and costs, a table of offer/demands and a legend. Then for each Simplex iteration it displays the task to execute on the current graph tree, the resulting new graph tree after updating. It also shows updated reduced costs table, and a summary of the current state: objective, flows, etc. After solving the problem it outputs on a separate page the optimal solution, objective and shows the flows on the graph. If necessary it can display the *thread* and *depth* arrays. Colors and the size of graphs is fully customizable. The current version support the english and spanish languages but adding a new language is not complicated. In regard to the graphical output engine, this package relies on two components, the LaTeX[6] typesetting system and the *dot* program of the GraphViz software[4]. Figure 1 shows a tipical example of graphical output. The following paragraph presents some technical details.

The GNU Oflox package consists of the `oflox` function and the auxiliary configuration routine `ofloxconfig`. This main function requires two parameters, the network description file and a configuration structure that indicates what to do and how. The network description file is a text file that specifies the network characteristics: number of nodes and node requirements, arcs and their bounds. The format used to describe the network is the well known DIMACS min format [2, Section 7]. In its most purest form the `oflox`

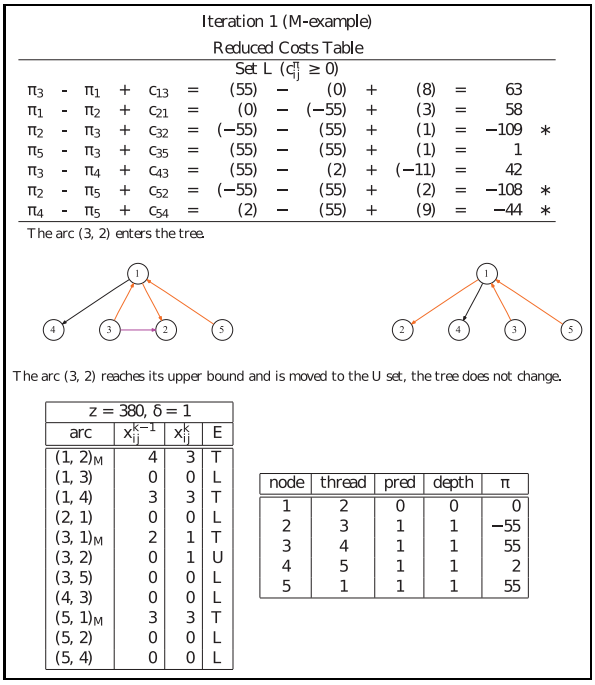

Fig. 1: Example showing a tipical iteration graphical output.

function returns the objective cost, the arc flow to attain the given objective cost, the number of iteration required and, if applies, the number of Phase I iterations. In order to use the full extent of the package, it is necessary to set-up the configuration structure using the `ofloxconfig` routine. To be able to retrieve a feasible solution from the network file a simple extension was added to the DIMACS min file format that supports the description of feasible trees in the file. To output the graphical steps of the algorithm, two type of files are generated per Simplex iteration: LaTeX files containing the textual information and DOT format files containing the information necessary to reconstruct the graph trees of the current state. Once the problem is solved, the DOT files are compiled to PostScript pictures and included in the master LaTeX document. This document can be compiled to DVI or PDF document formats. The current implementation has a minimal working set of features compatible with Matlab®, but there are plans to extend the list of available features for this system.

3 Final remarks

We presented a GNU Octave add-on package that solves the MCF problem. The software is oriented to educational purposes. Flexibility of the output configuration of the software package provides many attractive possibilities. It can be used to produce not only examination materials but also generate homework assignments or even be used as part of a project task. The final outputs are plain LaTeX files that can be modified and orchestrated into each individual needs. The package is being successfully used in the undergraduate course of Network flows (CO5422) of the Universidad Simón Bolívar since 2010. This add-on package can be freely downloaded from the project's page: `http://code.google.com/p/gnuoflox`.

References

[1] M. S. Bazaraa, J. J. Jarvis, and H. F. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, New York, NY, USA, second edition, 1990.

[2] Center for Discrete Mathematics and Theoretical Computer Science. The First DIMACS International Algorithm Implementation Challenge: Problem Definitions and Specifications, 1990.

[3] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[4] E. R. Gansner and S. C. North. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience*, 30:1203–1233, 2000.

[5] The MathWorks Inc. Matlab, 2010.

[6] Leslie Lamport. *LaTeX: a document preparation system*. Addison-Wesley Publishing Company, 1986.

[7] Octave community. GNU/Octave, 2012.