Using MCMCF to Solve Multicommodity Flow Problems*

Jeffrey D. Oldham

1998 October 09

1 Introduction

The program MCMCF approximately solves the minimum-cost multicommodity flow problems on directed graphs. For information about this problem, read the technical reports on the author's home page http://theory.stanford.edu/~oldham or in *Network Flows* [AMO93].

The sections below show show how to run the program, the input and output formats, and known restrictions.

2 Running the Programs

To invoke MCMCF, use a command line like

`./MCMCF filename.`

The results are sent to the standard output.

2.1 Execution-Time Options

The program's execution can be altered using command-line flags. Some of these flags include:

`-E` *epsilon*	overrides the desired program accuracy value ϵ specified in the input.
`-L` *minimum-*λ	indicates a lower bound on λ^*, i.e., the multiple of the arc capacities needed to satisfy the the demands. The program terminates when λ is within a factor $1+\epsilon$ of the specified value and the costs are within the same factor of the minimum possible costs. For the minimum-cost multicommodity flow problem, the default is 1.0. That is, stop when the flow fits within a network using at most $1+\epsilon$ times arc capacity (and has small enough cost).
`-l` *frequency*	indicates the number of iterations (as a multiple of the number of commodities) that should occur between calculation of the lower bound. The default value is 3.
`-t` *multiple*	helps the program find an initial solution. By default, the program attempts to find an initial solution by individually routing each commodity through the network with capacities as specified in the input. Using the flag, causes the program to trying routing the commodities through a network with capacities multiplied by the specified floating point number. If the problem can be solved with $\lambda^* \leq 1$, this flag should not be necessary.

For example, to solve a problem with 4% accuracy when the input specifies an accuracy of 10%, use

`./MCMCF -E 0.04` *`filename`*.

3 Min-Cost Multicommodity Flow Problem Input and Output Formats

The multicommodity flow format is an extension of the DIMACS Algorithm Implementation Challenge network flow formats [DIM91]. Each line of a problem specification (input) or multicommodity flow (output) starts with a unique, lowercase identifying letter.

3.1 Problem Specification Format

Comments. A comment begins with a lowercase `c` and ends at the end of the line.

```
c This is an example of a comment line.
```

Problem descriptor. One problem descriptor line must appear exactly once before any arc or commodity lines.

`p mcmcf` *`nodes arcs commodities`*

The integer *`nodes`* specifies the number n of nodes in the graph. The integers *`arcs`* and *`commodities`* specify the numbers m and k of arcs and commodities, respectively.

Arc descriptors. For each arc, there is exactly one arc descriptor line.

`a` *`src dst cap cost`*

The arc (*`src`*, *`dst`*) has capacity *`cap`* and cost *`cost`*. Nodes *`src`* and *`dst`* must have integral values in the range $[1, n]$. Capacities and costs are nonnegative floating point numbers. The number of arc descriptors must match *`arcs`* in the problem descriptor.

Commodity descriptors. For each commodity, there is exactly one commodity descriptor line.

`k` *`src dst demand`*

`demand` units of flow must travel from node *`src`* to node *`dst`*. The nodes *`src`* and *`dst`* must have integral values in the range $[1, n]$, while the demand may be any nonnegative floating point number. The number of commodity descriptors must match *`commodities`* in the problem descriptor.

Precision descriptor. Exactly one line specifying the desired approximation must appear.

`e` *`epsilon`*

The desired approximation *`epsilon`* must be a positive floating point number.

3.2 The Output Format

The result of executing `MCMCF` on a problem is a file with the following format:

Comments. A comment begins with a lowercase `c` and ends at the end of the line.

```
c This is an example of a comment line.
```

Solution line. The approximate minimum-cost of the multicommodity flow appears in this line.

`s` *`solution`*

Flow descriptors. For each arc and commodity, one line

`f` *`src dst commodity flow`*

specifies the flow of commodity *`commodity`* from node *`src`* to node *`dst`*.

4 Known Restrictions

`MCMCF` is a research tool and is still in development. Thus, we provide no guarantee about its behavior. The program has the following known restrictions. If you experience other difficulties, please contact the author.

1. The program solves the single-source, single-sink directed multicommodity flow problem. Internally, `MCMCF` groups commodities with the same source nodes. The output is for these grouped nodes. The author can construct a Perl script to convert these flows to correspond to the original commodities.

2. The program solves directed multicommodity flow problems. The author can provide a script to convert undirected problems to directed ones. With sufficient motivation, the author can modify the program internals to support directly undirected problems.

3. For large accuracies, e.g., accuracy $\epsilon < 0.2\%$, the Newton-Raphson method may enter an infinite loop because of precision problems.

4. Be sure to specify a lower bound on the permissible λ^* using the `-L` option.

5. Parallel arcs with the same tail and head nodes are not supported.

6. The arc costs and capacities are the same for all commodities. Each commodity must have a distinct pair of source and destination vertices.

7. If too high a precision is desired, e.g., 0.1%, the program may attempt to exponentiate a very large number and fail.

References

[AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[DIM91] The first dimacs international algorithm implementation challenge: Problem definitions and specifications. `ftp://dimacs.rutgers.edu/pub/netflow/-general-info/specs.{tex,bbl}`, 1991.