Generate network files

Table of contents

1. Where are these files ?

The classical network files we use are stored in the directory "files". You can find here 'R1m.mgl", "EU.mgl", 'nsfnet.mgl'. This files use the MGL format. It means that networks are represented by graphs because only graphs can be stored in a MGL file. Actually, those networks uses two graphs: the request graph and the cable graph. These two graphs share their vertex set. As these files contain two graphs, it is not easy to use the Editor to create some sample files. The goal of this page is to explain how to create some networks file using two possibilities: a Net file or an GOd file.

2. GOd format

The GOd format was developped by David Coudert and is used in by other programs not included in Mascopt.

The GOd format uses two files:

- e .graph file, which describes the graph
- The .od fil, which describes the requests

2.1. The .graph files

The .graph lines can contain comments. The lines of comments begins with "c". These lines are ignored when parsing. For example:

```
c
c NSFET, 14 noeuds et 21 aretes, symetrique
c
c donnees extraites de:
c Practical routing and wavelength assignment algorithms for
c    all optical networks with limited wavelength conversion
c M.D. Swaminathan,  K.N. Sivarajan
c IEEE ICC 2002
c
c
```

The first used line of the .graph file contains "n" and the number of nodes:

```
n 14
```

Then, the file must contains 14 lines with the flag "e" which means that this is an edge, then the number of the source node and the number of destination node. You can leave some lines empty. For example:

```
e 0 1
e 0 2
```

```
e 0 7
e 1 2
e 1 3
e 2 5
e 3 4
e 3 10
e 4 5
e 4 6
e 5 9
e 5 13
e 6 7
e 7 8
e 8 9
e 8 11
e 8 12
e 10 11
e 10 12
e 11 13
e 12 13
```

Note that the nodes of the network are numbered from 0 to n-1, if we consider n nodes. You can find the complete file in mascoptDev/files/nsfnet.graph.

2.2. .od file

The .od file contains the request on the graph. The use of two files allow to change the requests using the same network. As in the .graph file, you can put some comments with the flag "c":

```
c
c Instance pour
c NSFET, 14 noeuds et 21 aretes, symetrique
c
c 268 requetes au total
c
c donnees extraites de:
c Practical routing and wavelength assignment algorithms for
c    all optical networks with limited wavelength conversion
c M.D. Swaminathan,  K.N. Sivarajan
c IEEE ICC 2002
c
c
```

Then, you have just to declare the requests with the flag r: r source_node dest_node request_size. For example:

```
r 3 11 2
r 3 12 1
r 3 13 3
r 4 0 1
r 4 1 3
```

```
r 4 3 2
r 4 5 1
r 4 7 2
...
```

You can find the complete file in mascoptDev/files/nsfnet.od.

3. Net format

The Net format is dedicated to WDM networks. It has been extended to describe any network with hierarchical structure.

You can define some comments at the beginning of the file with "//". For example:

```
// ----------------------------------------------------------------------//
// Rosalie
//
// Simple nework in "R" design.
//
// All edges and requests exprimed in this file. Do not use the -mirrorEdge
//  nor -mirrorRequest options to translate this net in xml file.
// ----------------------------------------------------------------------//
//
//
```

You can also, after a line containing some data, add some comments with "//".

The first line containing data must give the number of levels (in a WDM network, 4 levels). Then, the next lines contain the names of each level (two times) and the number of sub-levels in this level. Here an example:

```
4 // 4 levels
CABLE   CABLE   0   // Main tag
FIBER   FIBER   4   // 4 bands in a fiber
BAND    BAND  8    // 8 lambdas in a band
LAMBDA LAMBDA  0 // Last tag
```

Then the data starts. The next line gives the number of nodes and the network and the following lines give the name of each nodes and its coordinates:

```
6
Zero    200    100
Un      400    100
Deux    200    300
Trois   400    300
Quatre 200    500
Cinq   400    500
```

Then the description of the links starts. Again, the first lines gives the number of requests, then the following lines give the description of each ling with "node1 node2 cable_size

cable_lenght". For example:

```
11
0 1 1 100
2 0 2 100
0 2 2 100
3 0 1 141
1 3 2 100
3 1 2 100
3 2 1 100
2 4 2 100
4 2 2 100
2 5 1 141
5 2 1 141
```

4. How to read these files in Mascopt ?

We implemented the classes wich read these two formats. You can find these classes in the package mascoptLib.io.graph. These classes works as the MGLReader works (it's quite the same).

5. How to convert the GOd format to an MGL file ?

From the reader classes, we implemented some small converters (that's easy to program).

5.1. od2mgl

You can find this program in mascoptDev/bin. You can also call it using:

```
java applications.Od2Mgl
usage: od2mgl fileIn.graph fileIn.od fileOut.mgl [-h] [--help]
Try `od2mgl --help' for more options.
```

This converter takes the .graph and .od files in input, and then writes the mgl file.

5.2. net2mgl

You can find this program in mascoptDev/bin. You can also call it using:

```
java applications.Net2Mgl
usage: net2mgl fileIn.net fileOut.mgl [-nma] [--noMirrorArc] [-nmr]
[--noMirrorRequest] [-h] [--help]
Try `net2mgl --help' for more options.
```

This converter takes the .net file and writes the mgl file. You have some optional flags: noMirrorArc does not duplicates the arcs of the networks (which is done by default). The noMirrorRequest does not duplicates the requests (which is done by default).