Matrikelnr.:

Name:

Discrete Optimization

WS 2013/14

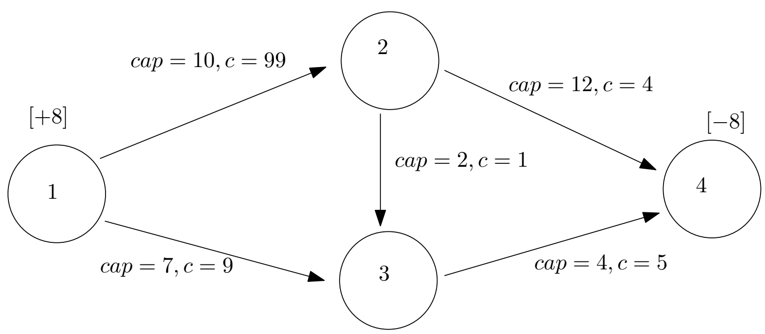
Institut für Formale
Methoden der
Informatik
Abt. Algorithmik
Universität Stuttgart

Exercise Sheet 2

Problem 1

Install the software package *mcf* which solves minCostFlow problems from the following URL: `http://typo.zib.de/opt-long_projects/Software/Mcf/`.

Both binaries (executables) as well as the source code is available there. When compiling from the source packages on Linux you might have to remove the `-mcpu=...` and `-march=...` options from `Makefile.arch.Linux`. You can also run the Windows executables under Linux using wine!



The following file describes this problem instance in a format readable for mcf:

```
p min 4 5
n 1 8
n 2 0
n 3 0
n 4 -8
a 1 2 0 10 99
a 1 3 0 7 9
a 2 3 0 2 1
a 2 4 0 12 4
a 3 4 0 4 5
```

Read the input format specification in the mcf documentation to understand how this describes the above minCostFlow problem instance. Create a text file (e.g. `mcf-ex1.mcf`) with the above content and run `mcflight` with that filename as an argument (e.g. `mcflight.Linux mcf-Ex1.mcf`). The solution should be written out to `mcf-ex1.mcf.sol`. Read and understand the solution file and draw the optimal flow in the above graph.

Problem 2

You are running a coach service (like meinfernbus.de) each day making a roundtrip Freiburg → Karlsruhe→ Stuttgart → Ulm → Augsburg → München → Friedrichshafen → Freiburg. Apart from transporting passengers your vehicle has also space to carry 10 moving cartons between the cities.

There is considerable demand for transportation of such moving cartons and people are willing to pay the following prices to have a carton transported between two cities:

trip	# cartons	price/box
Freiburg→Stuttgart	9	9 €
Freiburg→Karlsruhe	5	6 €
Karlsruhe→Augsburg	7	10 €
Stuttgart→München	9	11 €
Stuttgart→Freiburg	1	13 €
Ulm→München	6	5 €
Augsburg→ Friedrichshafen	10	12 €

How many boxes of each source-destination pair should be taken onboard if the revenue is to be maximized and your bus capacity not to be exceeded?

(a) Model this problem as a minCostFlow instance (draw the respective network).

(b) Draw the feasible flow corresponding to zero revenue into the network. Perform two rounds of cycle cancelling.

(c) Start with the zero flow and perform two rounds of the successive shortest path algorithm.

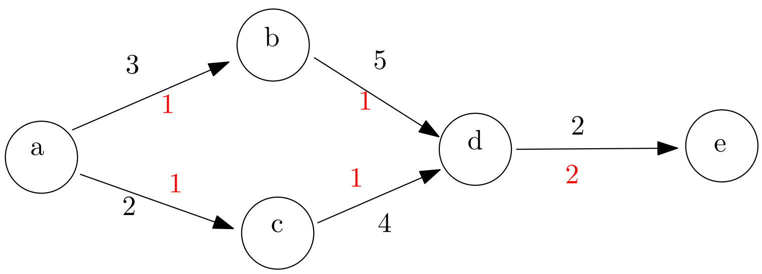
(d) Solve this minCostFlow problem instance using the mcf software package.

Draw the optimal flow and the respective residual network.

Problem 3

On the last exercise sheet, the following question was posed:

> Is there a always a sequence of augmenting paths that does not use any virtual back edges and still yields a maximum flow? Prove or disprove (the latter you can do via an example).



While there is always a sequence of augmenting paths not using any virtual back edges which leads to any maximum flow, the flow to send along each of these augmenting paths is sometimes not equal to the respective bottleneck value. For the above graph consider the maximum flow in red; the only two augmenting paths which yield this maximum flow are *abde* and *acde*. Whichever is chosen first, one has to send less flow along that path than the respective bottleneck value.

On the other hand, for this example there is some other maximum flow (e.g. a flow of 2 along *abde* only) which can be realized by a sequence of augmenting paths with flow value equal to the respective bottleneck value (in this case, it is only one augmenting path).

Now the new question is whether there is always some maximum flow which can be constructed by a sequence of augmenting paths not using back edges and always used up to their bottleneck values.