Matrikelnr.:

Name:

Discrete Optimization

WS 2013/14

Institut für Formale Methoden der Informatik
Abt. Algorithmik
Universität Stuttgart

Exercise Sheet 3

Problem 1

Install the software package *glpk* (GNU Linear Programming Kit) which solves (integer) linear programs. On Debian-based Linux (like Ubuntu) distributions this can typically be done by:

```
sudo apt-get install glpk glpk-doc
```

But you are free to compile the software yourself by getting the source code from `http://www.gnu.org/software/glpk/`. Windows/MacOS users can also compile from the sources or get precompiled executables at `http://sourceforge.net/projects/winglpk/` (Windows only).

The following is a simple linear program expressed in the *CPLEX LP format*, a description language for LPs which glpk can read:

```
Maximize
  value: 2x +  y
Subject To
  first: 0.5x + y<=5
  second: -x + y>=0
  third: x >= 0
  fourth: y >= 0
End
```

(a) Draw the respective feasible region and determine the optimal solution by looking at your drawing.

(b) Write the above LP in a file (e.g. `glpk-Ex1.lp`) and solve it by running glpk as follows:

```
glpsol --lp glpk-Ex1.lp -o glpk-Ex1.sol
```

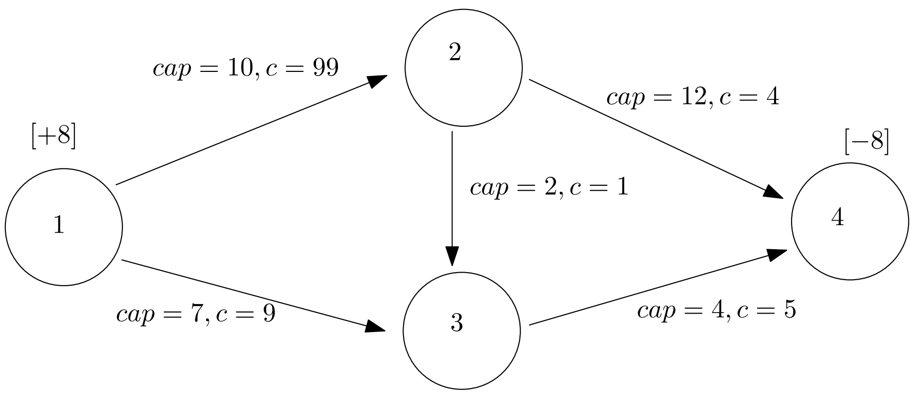
Inspect the solution written to `glpk-Ex1.sol`, what is the optimum objective function value for which values for x and y?

Remarks:

- apart from solving linear programs, glpk also allows for the solution of integer linear programs; read the included documentation to get an overview of glpk's functionality
- glpk not only provides a standalone solver but also an API such that you can easily integrate the (I)LP solver

Problem 2

Consider the following well-known min-cost flow problem instance.



Formulate it as an LP and solve it using glpk. Write the resulting flow into the above drawing.

Problem 3 One might restrict the linear program from problem 1. to *integral solutions*, that is, values for x, y which are integral.

(a) Mark all feasible *integra*l points for the linear program from problem 1.

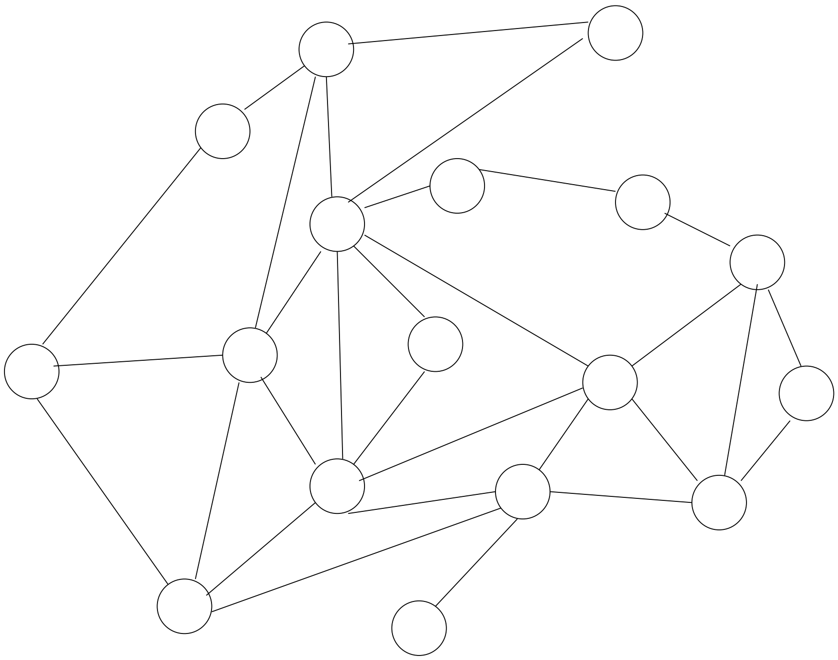
(b) Read in the documentation for gtklp how to modify the problem specification file to restrict the values of a variable to be integral (Section on the CPLEX LP format).

(c) Compute the optimum solutions for the cases where

- both x and y are to take integral values
- x is to take integral values only whereas y might take fractional values
- y is to take integral values only whereas x might take fractional values

Mark them in your drawing.

Problem 4 Consider the following graph:



(a) Look at the graph and construct – after a brief visual inspection – a *vertex cover* of minimal cardinality according to your intuition.

(b) Formulate the problem of finding a minimum cardinality vertex cover for the above graph as an integer linear program.

(c) Compute both the optimum solution of the lp relaxation (allowing fractional values for the variables) as well as the optimum integral solution using glpk. Visualize the latter in the above graph.

(d) The *integrality gap* of a minimization integer linear program I is the maximum possible value of the ratio $\frac{OPT_{int}(I)}{OPT_{frac}(I)}$ (where $OPT_{int}(I)$ denotes the objective function value of the optimum integral solution, and $OPT_{frac}(I)$ the optimum fractional solution when the integrality constraint is dropped). Construct an instance of the vertex cover problem which exhibits a large integrality gap of the respective ILP formulation.

(e) Derive the dual of the primal LP; compute the respective optimum integral solution to this dual LP using glpk. Visualize the result in the above graph.

Problem 5

Assume that we have an algorithm that computes an optimal solution to a linear program in standard form (as we defined it above). However, suppose that we are given a linear program in the following form:

$$\begin{aligned} \min \quad & c^T x \\ & A_1 x \leq b_1 \\ & A_2 x = b_2 \\ & A_3 x \geq b_3 \\ & x \geq 0 \end{aligned}$$

In other words, we have a *minimization* problem, and the constraint set contains equalities and $\geq$-type constraints. Explain how our algorithm can be used to solve this kind of problem, too.

Problem 6

Consider a linear program (P) in our standard form and its dual (D). Show that the dual of (D) is again (P).

Problem 7

Show how to find an initial feasible corner for the primal Simplex algorithm to start with.

Problem 8

Show how to find an initial V-Shape for the dual Simplex algorithm to start with.

Problem 9

Show for both the primal as well as the dual Simplex algorithm that a constraint which has left the basis during the course of the algorithm might reenter the basis again. (Hint: can be shown by an example in R^2.)