Step 3: Initialize the pivot: Instructions

Help

Step 3: Initialize and Complete

Bundle of unit tests and assignment parts for this step are available online at ZIP or alternatively as a tar.gz file.

The goal of this step is to perform initialization phase simplex and put the two phases together to build a complete solver. Thus far, the dictionaries we provided were all pre-initialized for you. Now, we will have to drop that assumption.

Dictionary

A typical dictionary looks like this:

$$\begin{array}{c|ccccc}
x_{B1} & b_1 & +a_{11}x_{N1} & +\cdots+ & a_{1n}x_{Nn} \\
\vdots & \vdots & & \ddots & \\
x_{Bj} & b_j & +a_{j1}x_{N1} & +\cdots+ & a_{jn}x_{Nn} \\
\vdots & & & & \\
x_{Bm} & b_m & +a_{m1}x_{N1} & +\cdots+ & a_{mn}x_{Nn} \\
\hline
z & z_0 & +c_1x_{N1} & +\cdots+ & c_nx_{Nn}
\end{array}$$

Steps 1 and 2 guaranteed that $b_1, \ldots, b_m \geq 0$ to start with, so that you did not bother with initialization. This time, we will drop that assumption.

Dictionary Format

Dictionary format will remain the same as it has been in steps 1 and 2. For your convenience it is recalled below. We will store a dictionary as a text file. The text file will have the following format:

```
[Line 1] m n
[Line 2] B1 B2 ... Bm [the list of basic indices m integers]
[Line 3] N1 N2 ... Nn [the list of non-basic indices n integers]
[Line 4] b1 .. bm (m floating point numbers)
[Line 5] a11 ... a1n (first row coefficients excluding the constant coefficient)
....
[Line m+4] am1 ... amn (mth row coefficients excluding the constant coefficient)
[Line m+5] z0 c1 .. cn (objective coefficients (n+1 floating point numbers))
```

Note: Text shown in black and line numbers are **not** part of the input file.

For example, the dictionary

$$\begin{array}{c|cccccc} x_1 & 4 & +2x_3 & -3x_4 & +x_2 & +x_7 \\ x_5 & 5 & -x_3 & +3x_4 & -x_2 & -2x_7 \\ x_6 & -1 & & -x_4 & +x_2 & +3x_7 \\ \hline z & 10 & -x_3 & +x_4 & -x_2 & +0x_7 \end{array}$$

is represented by the file

```
3 4
1 5 6
3 4 2 7
4.0 5.0 -1.0
2.0 -3.0 1.0 1.0
-1.0 3.0 -1.0 -2.0
0.0 -1.0 1.0 3.0
10.0 -1.0 1.0 -1.0 0.0
```

Here is an explanation of the numbers above:

- The first line says " 3 4 " and indicates $m = 3$ and $n = 4$.
- The second line says "1 5 6" and indicates that the basic variables are x_1, x_5, x_6
- The third line says "3 4 2 7" and indicates that the non-basic variables are x_3, x_4, x_2, x_7
- The fourth line says "4.0 5.0 -1.0" and indicates that the $\vec{b}$ coefficients (constant coefficients) are $\vec{b} = \begin{pmatrix} 4 \\ 5 \\ -1 \end{pmatrix}$.
- Lines 5-7 represent the 3×4 matrix A.
- Line 8 (the last line) has two parts: the first entry is $z = 10$ the current objective value and the rest are objective row coefficients.

Assignment

The files for the assignment are provided under the directory `assignmentParts/` For this step, you are asked to perform the following steps:

- First, perform initialization phase Simplex. You can use one of two methods: initialization using the dual problem (preferred) or initialization using the auxiliary problem (you can do this if you wish, but we advise initialization using the dual).
- After initialization phase Simplex, you will obtain a feasible dictionary for the primal LP problem. You should reuse the code from step 2 as much as possible.
- Reuse the code from step 2 to perform optimization phase Simplex.
- Some students from Eastern Europe use `,` (comma) instead of the standard decimal point. We remind them that our grader expects decimal point.

Output format

If the problem has an optimal solution, the output file should have a number that corresponds to this optimal objective. Eg., if the LP has an optimal value of -3.01, your file should simply be a single line: `-3.01` Nothing else should be in that file. More significantly, **DO NOT** report the number of steps. On the other hand, if the problem is unbounded, your file should have a single line with `UNBOUNDED` Finally, if the problem is infeasible, your file should have a single line with `INFEASIBLE`

Zip Bundle (tar ball)

Bundle of unit tests and assignment parts for this step are available online at ZIP or alternatively as a tar.gz file.
The unit tests are available in `unitTests` directory, also have sample output files that you can use to check your answers.

The `assignmentParts` directory consist of five parts. Once you solve a given part, you are welcome to upload the answer here and get graded.

Under `unitTests` there are three subdirectories `unitTests/10`, `unitTests/20`, `unitTests/50`, These represent problem sets with increasing sizes. We recommend trying to match answers with the unit tests in the directory 10.

Each unit test subdirectory has 100 unit tests in files labelled `test0.dict` to `test99.dict` The output for each are given under `test0.output` to `test99.output` Finally, the files `test0.tex` to `test99.tex` provide the LaTex sources that can be compiled using a Latex compiler to provide you a step by step output from our sample implementation. The file with `.tex` extension is a latex file that can be compiled using a suitable latex compiler to yield a PDF that shows the steps precisely.

- The sample code initializes using the dual problem.
- The objective is always changed to max -x1 -x2 -x3 ... -xn , with -1 for all objective coefficients.
- Our sample implementation follows Bland's rule for both initialization and optimization phases.

Following this convention, you will be able to reproduce our exact answers step by step. But you are not required to do so by any means. Finally, `test0.ampl` to `test99.ampl` provide AMPL format problems for the dictionary that you can use inside a solver like GLPK to test your, and our answers, as well. :-) All the best