Programming Assignment 1

1 Questions

The aim of this project is to implement the simplex method. As we have seen, the simplex method has two stages: in the first stage, we find a basic feasible solution for our problem (or we discover that the feasible area of the problem is empty), and then in the second stage we implement the simplex method starting at this basic feasible solution. In this project we will do the two parts separately at first, and then combine them to provide a full implementation of the simplex method.

When you write the code you need to think of all the obstacles that might arise, and test your code with problems with all these different types of obstacles.

Some important things to keep in mind:

(i) A basis is always represented as a vector of indices.

(ii) You can always assume that the input makes sense. For example, if you are given a vector of indices (which is supposed to be a feasible basis) as input, you can assume that the column vectors corresponding to the indices in this vector form a basis, and that this basis gives a feasible solution.

(iii) Be careful about the cases when a problem has feasible set empty, or is unbounded. Be especially careful about how you handle this in the third question. Be careful about degenerate solutions.

(iv) When you make an auxiliary linear program, remember that the actual solutions you want are of the original program. This means you need to interpret the solution of the auxiliary program, maybe do a little extra work, and then get a solution for the original program.

(v) Since you will have to think about the basic and non-basic variables, be careful that the final bfs is in the right order. For example, if (s_1, s_2, s_3) is the final bfs, then s_2 should be the value of the second variable *in the problem as originally stated* etc.

(vi) Again, think of all the weird things that may happen, and design your code so that you take care of these.

(vii) Solve Question 2 first. Use it in the solution of Question 1.

(viii) With Question 1, you need to be careful that the output is actually a basis.

(ix) Given an $m \times k$ matrix A and a vector $\vec{b}$ in $\mathbb{R}^m$, the following is a useful way of checking in MATLAB whether the system $A\vec{x} = \vec{b}$ has any solutions or not: this system has solutions iff `norm(A*(A\b) - b, inf) = 0`.

(x) Some other useful functions: `ones`, `eye`, `zeros`.

Question 1. *Implement the first phase of the simplex method for a problem of the following type:*

$$\textit{maximise } \vec{c}^T\vec{x} \textit{ subject to } A\vec{x} = \vec{b},\ \vec{x} \geq \vec{0},\ \vec{b} \geq \vec{0}.$$

That is, you want to check find a basic feasible solution for a problem of this sort if one exists. Specifically, define a function

```
function[nvac, basis, bfs] = phaseOne(A, b, c)
% maximise c^T x
% subject to Ax = b, x >= 0, b >=0
%
% Input:
% A mxn matrix with m <= n and rank of A is m
% b column vector with m rows
% c column vector with n rows
%
% Output:
% nvac = 0 if the feasible set is empty
% nvac = 1 if the feasible set is non-empty
% basis = a vector of size m of indices of column vectors for a feasible basis for the problem if the feasible set is non-empty
% bfs = a vector of size n of the basic feasible solution corresponding to this basis (if the feasible set is non-empty)
...
end
```

Question 2. *Implement the second phase of the simplex algorithm for a problem of the following type:*

$$\textit{maximise } \vec{c}^T\vec{x} \textit{ subject to } A\vec{x} = \vec{b},\ \vec{x} \geq \vec{0},\ \vec{b} \geq \vec{0}.$$

That is, starting from a basic feasible solution, find the optimal solution (if it exists). Specifically, define a function

```
function[bound, obasis, obfs, oval] = phaseTwo(A, b, c, sbasis, sbfs)
% maximise c^T x
% subject to Ax = b, x >= 0, b >=0
%
% Input:
% A mxn matrix with m <= n and rank of A is m
% b column vector with m rows
% c column vector with n rows
% sbasis a vector of size m of indices of column vectors for a feasible basis for this problem from which to start the simplex method
% sbfs a vector of size n which is the basic feasible solution corresponding to this basis
%
% Output:
```

```
% bound = 0 if the problem is unbounded (there is no optimal solution)
% bound = 1 if the problem is bounded (there is an optimal solution)
% obasis = a vector of size m of indices of column vectors which gives an optimal feasible basis for the problem if the problem is bounded
% obfs = a vector of size n which is the optimal basic feasible solution corresponding to this optimal basis if the problem is bounded
% oval = the objective value of this optimal basic feasible solution (if the problem is bounded)
...
end
```

Question 3. *Implement the full simplex algorithm for a problem of the following type:*

$$\text{maximise } \vec{c}^T\vec{x} \text{ subject to } A\vec{x} = \vec{b},\ \vec{x} \geq \vec{0},\ \vec{b} \geq \vec{0}.$$

That is, starting from scratch, find the optimal solution (if it exists). Naturally, you should make use of the two functions you have defined for the two phases of the simplex method. Specifically, define a function

```
function[status, obasis, obfs, oval] = bothPhases(A, b, c)
% maximise c^T x
% subject to Ax = b, x >= 0, b >=0
%
% Input:
% A mxn matrix with m <= n and rank of A is m
% b column vector with m rows
% c column vector with n rows
%
% Output:
% status = -1 if the feasible set is empty
% status = 0 if the feasible set is non-empty but the problem is unbounded (there is no optimal solution)
% status = 1 if the problem is bounded (there is an optimal solution)
% obasis = a vector of size m of indices of an optimal feasible basis for the problem if the feasible set is non-empty and the problem is bounded (in terms of a set of indices of column vectors)
% obfs = a vector of size n which is the optimal basic feasible solution corresponding to this optimal basis if the feasible set is non-empty and the problem is bounded
% oval = the objective value of this optimal basic feasible solution (if the feasible set is non-empty and the problem is bounded)
...
end
```

2 Rules and comments

(i) You can do the project in a group of at most three people.

(ii) The deadline is 18:00, Tuesday, 30 April.

(iii) The file you send me should be called `firstassignment.m`.

(iv) DO NOT MAKE A SPELLING MISTAKE IN NAMING THE FILE OR THE FUNCTIONS. THAT WILL MEAN YOU DO NOT GET ANY POINTS FOR THE PROJECT. I WILL WRITE A PROGRAM TO TEST THE CODE WITH A SERIES OF TESTS TO CHECK SPECIFIC ASPECTS, SO THE FIRST AND MOST IMPORTANT TEST IT WILL RUN IS IF THE NAMES OF THE FUNCTIONS AND THE FILE ARE CORRECT.

(v) The name and clave unica of everyone on the group should be included in the email you send me with the file. Every person in the group should be cc-ed.

(vi) You should also state the name and clave unica of every person participating in the project in the beginning of the file. Also, your file should have a brief description of what you have done with some small explanations.

(vii) Please comment the code well so that if necessary I would be able to understand your code.

(viii) Please don't make any mistakes.