ISyE 6669, Deterministic Optimization
Homework #4, Due never (but you’ll need to know how to do these problems in order to do well on the first midterm)

1. In class, we described the Dual Revised Simplex algorithm for maximization problems: start with a dual feasible basis, and iteratively move to a better dual feasible basis until we find one that is also primal feasible – this will be optimal.

Primal: 
Maximize \( cx \) 
\( \text{st} \ Ax = b \) 
\( x \geq 0 \)

Dual: 
Minimize \( \text{ub} \) 
\( \text{st} \ uA \geq c \)

1) Start with a dual feasible basis \( B \)
2) If the basis is also primal-feasible (if \( x_B = B^{-1}b \geq 0 \)) then stop, we have the optimal solution. Otherwise,
   a) Find a primal-infeasible variable (a variable that is negative) to leave the basis. Call this row \( i \).
   b) Pick a variable to enter the basis, making sure that we retain dual feasibility. In this case, dual feasibility means \( uA \geq c \). Since \( u = c_B B^{-1} \), we need all \( c_k - c_B B^{-1}a_k \leq 0 \) for dual feasibility. The rate-of-change for any reduced cost (value of the dual excess variable) is \( -(B^{-1}a_k)_i \), so to find the entering variable that retains dual feasibility we want to choose from among all variables with negative values of \( (B^{-1}a_k)_i \) and pick the one with the smallest ratio \( (c_k - c_B B^{-1}a_k)/(B^{-1}a_k)_i \). Call this column \( j \).

So, the variable in column \( j \) will enter the basis, replacing the \( i \) th basic variable. To update \( B^{-1} \), create \( E \) as in primal revised simplex, and then \( B^{-1}_{\text{new}} = EB^{-1}_{\text{old}} \).

(a) That’s the dual revised simplex algorithm for a primal maximization problem. For this homework problem, I want you to describe the dual revised simplex algorithm for a primal minimization problem. Do not do what the book does (multiply the objective function by \(-1\) and call it a max problem) – instead, think about what we’re doing in each step, and how each step would (or would not) be different for a primal minimization problem. [You should be able to do this problem now.]

1) Start with a dual feasible basis \( B \).
2) If the basis is also primal feasible (if \( x_B = B^{-1}b \geq 0 \)) then stop, we have the optimal solution. Otherwise,
   a) Find a primal-infeasible variable (a variable that is negative) to leave the basis. Call this row \( i \).
b) Pick a variable to enter the basis, making sure that we retain dual feasibility. Until this step, nothing we did involved the primal objective function, so it didn’t matter whether we were maximizing or minimizing – it was all the same. But now, there’s a difference. Let’s look at our primal and dual problems:

\[
P: \text{minimize } cx \quad \text{subject to } Ax = b, \quad x \geq 0, \quad c \geq 0
\]

\[
D: \text{maximize } ub \quad \text{subject to } uA \leq c
\]

Notice that, because we now have a primal minimization problem, the dual constraints are ≤ constrains (instead of ≥ constrains for the dual of a primal maximization problem). So, for dual feasibility we need \( uA \leq c \). Since \( u = cB^{-1} \), dual feasibility is \( cB^{-1}A \leq c \), or, for each dual constraint \( j \), \( c_j - cB^{-1}a_j \geq 0 \) (in other worse, all primal reduced costs should be positive).

We will be limited by variables \( x_j \) for which \( (B^{-1}a)_i < 0 \); for positive values, the reduced cost will rise and will always remain positive. Therefore, we need to choose the variable which has the smallest positive value of \( (c_j - cB^{-1}a_j)/(-B^{-1}a)_i \). This will be the entering variable.

So, the variable in column \( j \) will enter the basis, replacing the \( i \)th basic variable. To update \( B^{-1} \), create \( E \) as in primal revised simplex, and then \( B^{-1}_{\text{new}} = EB^{-1}_{\text{old}} \).

(b) Suppose the primal problem is a minimization problem as in part (a), and all of the costs \( c \) are positive \( (c > 0) \). Then, \( u = 0 \) is a dual feasible solution to the dual, so if we knew a basis for which all \( u = 0 \), it would be a good starting basis for dual simplex. Can you find such a basis? [You probably won’t be able to do this problem until after class on Monday, 9/24.]

In fact, it may not be possible to find such a basis. Recall from class that, by complementary slackness, for every primal basic variable \( x_j \), the \( j \)th dual constraint must be satisfied at equality. If, for example, all \( c > 0 \), then none of the dual constraints \( uA \leq c \) will be satisfied at equality when \( u = 0 \). This means that \( u = 0 \), although a feasible solution to the dual, is not a corner point of the dual feasible region – instead, it is somewhere inside the feasible region.

2. From the last homework, find the optimal solution to problem 1 part (c) – the optimal basis. In part (d,ii) we found that when the right-hand side of the machine 2 constraint changed from 24 to 48, we had to start at the beginning and re-optimize the problem.
Now that we know dual simplex, we don’t need to start all over again. Instead, start at the optimal basis from part (c), and change the right-hand side of the machine 2 hours constraint to 48. Notice that we have a dual-feasible basis (all the reduced costs satisfy dual feasibility) but the problem is primal infeasible because one of the variables is negative.

Instead of re-optimizing from the beginning, start at this basis and use dual revised simplex to re-optimize the problem. [You should be able to do this problem now.]

Recall that the linear program we are solving is the following:

**Minimize** \( 4000 x_1 + 1000 x_2 \)

**Subject to**

\[
\begin{align*}
300 x_1 + 100 x_2 - e_1 &= 3000 \\
100 x_1 + 100 x_2 - e_2 &= 500 \\
100 x_1 + 200 x_2 - e_3 &= 2000 \\
x_1 + s_4 &= 24 \\
x_2 + s_5 &= 24 \\
x_1, x_2, e_1, e_2, e_3, s_4, s_5 &\geq 0
\end{align*}
\]

At the optimal solution, the basic variables are \( \{s_4, e_2, e_3, x_1, x_2\} \), and \( B^{-1} \) is the following:

\[
B^{-1} = 
\begin{bmatrix}
\frac{1}{300} & 0 & 0 & 1 & \frac{1}{3} \\
\frac{1}{3} & -1 & 0 & 0 & \frac{662}{3} \\
\frac{1}{3} & 0 & -1 & 0 & \frac{1662}{3} \\
\frac{1}{300} & 0 & 0 & 0 & \frac{1}{3} \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
x_B = B^{-1}b = \begin{bmatrix}
22 \\
2100 \\
3000 \\
2 \\
24
\end{bmatrix}
\]

However, when we change the right-hand side of the last constraint from 24 to 48, we get the following values for the basic variables:

\[
x_B = B^{-1}b = \begin{bmatrix}
30 \\
3700 \\
7000 \\
-6 \\
48
\end{bmatrix}
\]
We have a negative value for a basic variable, so the solution is no longer optimal. However, it is still dual feasible, so we can use dual revised simplex to solve the problem.

ITERATION 1

There is only one negative basic variable, so \( x_1 \) will leave the basis. \( x_1 \) is the fourth basic variable.

Now, let’s check the reduced costs and the values of \((B^{-1}a)_4\) (because the fourth basic variable is leaving the basis).

Reduced costs for nonbasic variables:

It will be easier to calculate \( c_BB^{-1} \) once, instead of repeating the calculation for every nonbasic variable.

\[
\begin{bmatrix}
\frac{-1}{300} & 0 & 0 & \frac{1}{3} \\
\frac{1}{3} & -1 & 0 & 0 & 66^{2/3}
\end{bmatrix}
\]

\( c_BB^{-1} = \begin{bmatrix} 0 & 0 & 4000 & 1000 \end{bmatrix} \begin{bmatrix}
\frac{1}{3} & 0 & -1 & 0 & 166^{2/3} \\
\frac{1}{300} & 0 & 0 & 0 & -1^{1/3} \\
0 & 0 & 0 & 0 & 1
\end{bmatrix} = \begin{bmatrix} 13^{1/3} & 0 & 0 & -333^{1/3} \end{bmatrix} \]

\( e_1 : c_{e1} - c_BB^{-1}a_{e1} = 0 - -13^{1/3} = 13^{1/3} \)

\( s_5 : c_{s5} - c_BB^{-1}a_{s5} = 0 - -333^{1/3} = 333^{1/3} \)

(Actually, we knew these from our final iteration of primal revised simplex in the previous homework.)

Rates of change:

\( e_1 : (B^{-1}a_{e1})_4 = \frac{-1}{300} \)

\( s_5 : (B^{-1}a_{s5})_4 = \frac{-1}{3} \)

Ratios:

\( e_1 : \left(\frac{40^{1/3}}{1^{300}}\right) / \left(\frac{1}{300}\right) = 4000 \)

\( s_5 : \left(\frac{1000^{1/3}}{1^{3}}\right) / \left(\frac{1}{3}\right) = 1000 \)

So \( s_5 \) will enter the basis, because it has the smallest ratio.
So to get the special column, we divide each entry but the fourth by $\frac{1}{3}$ (because the leaving variable was the fourth one) and the fourth entry is just $1/(\frac{1}{3})$.

$$E = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 200 & 0 \\ 0 & 0 & 1 & 500 & 0 \\ 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 3 & 1 \end{bmatrix}$$

Now we calculate the new value of $B^{-1}$, as in primal revised simplex. As we had before, $B^{-1}_{\text{new}} = EB^{-1}_{\text{old}}$.

$$B^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 200 & 0 \\ 0 & 0 & 1 & 500 & 0 \\ 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{300} & 0 & 0 & 1 & \frac{1}{3} \\ \frac{1}{3} & -1 & 0 & 0 & 66\frac{2}{3} \\ \frac{1}{3} & 0 & -1 & 0 & 166\frac{2}{3} \\ \frac{1}{300} & 0 & 0 & 0 & -\frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, let’s check the new values of the basic variables $\{s_4, e_2, e_3, s_5, x_2\}$:

$$x_B = B^{-1}b = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 2 & 0 & -1 & 0 & 0 \\ -\frac{1}{100} & 0 & 0 & 0 & 1 \\ \frac{1}{100} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3000 \\ 500 \\ 2000 \\ 24 \\ 48 \end{bmatrix} = \begin{bmatrix} 24 \\ 2500 \\ 4000 \\ 18 \\ 30 \end{bmatrix}$$

All of the basic variables are nonnegative, so the solution is optimal.

3. Long John Jarvis the pirate discovered a vast treasure hidden on a deserted island. There were limitless amounts of gold and silver coins, jewel-encrusted plates, and ivory statues, all of which were completely unguarded and could be taken at no risk. Unfortunately, Long John’s ship had been sunk, and he had only a small rowboat in which to put his treasure. The rowboat could hold up to 1000 pounds (plus the pirate himself) and a volume of 200 cubic feet.

The table below gives the value, size, and weight of each type of treasure:

<table>
<thead>
<tr>
<th>Value (dollars)</th>
<th>Size (cubic feet)</th>
<th>Weight (pounds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Being a very worldly and intelligent pirate, Long John was familiar with the basic principles of linear programming, and set up the following LP to help him decide how much of each type of treasure he should take with him in order to maximize his wealth.

Variables

- $x_1$: Number of gold coins taken
- $x_2$: Number of silver coins taken
- $x_3$: Number of jewel-encrusted plates taken
- $x_4$: Number of ivory statues taken

Maximize

$$200x_1 + 100x_2 + 4000x_3 + 10000x_4$$

Subject to

$$
0.000145x_1 + 0.000145x_2 + 0.04x_3 + 0.75x_4 \leq 200 \\
0.177x_1 + 0.096x_2 + x_3 + 20x_4 \leq 1000 \\
x_1, x_2, x_3, x_4 \geq 0
$$

Because the boat’s limits were merely approximations, Long John was not concerned with restricting his variables to be integers; he could just round his solution off to the nearest integer if necessary.

A more important problem for Long John was that his copy of LINDO was destroyed when his ship was sunk, so he could only solve linear programs graphically by drawing in the sand with a stick. Since this linear program has four variables, it would require a 4-dimensional graph to solve, which was obviously impossible for Long John to draw using a stick in the sand.

Not knowing how to solve his linear program, Long John just grabbed as many gold coins as he could (5650 coins) and returned home with a treasure worth $1,130,000. However, as a more advanced student of linear programming than Long John was, you can do better!

Use duality to solve Long John’s problem using only graphical solution methods. (You may also need to draw upon your shadow prices, including both the definition and how to calculate them.) Show all of your work. [You should be able to do this problem now.]

The original problem has 4 variables, but only 2 constraints. Therefore, if we take the dual of the problem, it will have 4 constraints but only 2 variables, and we can solve it graphically.
Let $u_{size}$ and $u_{weight}$ be the dual variables associated with the primal size and weight constraints. The dual is then

\[
\text{Minimize} \quad 200 \, u_{size} + 1000 \, u_{weight}
\]

Subject to

\[
\begin{align*}
0.000145 \, u_{size} + 0.177 \, u_{weight} & \geq 200 \quad \text{(gold)} \\
0.000145 \, u_{size} + 0.096 \, u_{weight} & \geq 100 \quad \text{(silver)} \\
0.04 \, u_{size} + u_{weight} & \geq 4000 \quad \text{(plates)} \\
0.75 \, u_{size} + 20 \, u_{weight} & \geq 10000 \quad \text{(statues)} \\
u_{size}, u_{weight} & \geq 0
\end{align*}
\]

We can solve this problem graphically (see attached sheet).

Recall that the primal shadow prices are the optimal dual variables, and the dual shadow prices are the optimal primal variables. So, if we can find the dual shadow prices from our graphical solution, we’ll know the optimal solution to Long John’s original problem.

When we look at our graphical solution, we see that three of the constraints, the gold, silver, and statue constraints, are not binding at the optimal solution. Therefore, their shadow prices are zero; Long John should take zero gold coins, zero silver coins, and zero ivory statues.

The optimal dual solution is at the intersection of the plate constraint and the non-negativity constraint for the variable $u_{size}$. Therefore, to calculate the shadow price of the plate constraint, we can add one to the plate constraint’s right-hand side and solve for the intersection of those two constraints:

\[
\begin{align*}
u_{size} & = 0 \quad \text{(non-negativity for } u_{size}) \\
0.04 \, u_{size} + u_{weight} & = 4001 \quad \text{(plates)}
\end{align*}
\]

This intersection is at $u_{size} = 0$, $u_{weight} = 4001$. Thus $u_{size}$ is the same as in the optimal solution and $u_{weight}$ has increased by 1, so the objective has increased by 0 times the objective coefficient of $u_{size}$, plus 1 times the objective coefficient of $u_{weight}$, or 1000. Therefore, the shadow price of the plate constraint is 1000, and Long John should take nothing but 1000 jewel-encrusted plates, for a total worth of $4,000,000.$
The feasible region is to the upper right of all the constraints, and the optimal solution is (0,4000).