15.094J/1.124J/SMA5223 Systems Optimization: Models and Computation
Assignment 1 (115 points)
Due February 19, 2004

1 An Exercise using OPL Studio (Portfolio Optimization) (20 points)

The purpose of this question is to give you an introduction to using OPL Studio with a simple LP model. Consider the following problem.

Last year, Ellen Grunbaum purchased s_i shares of stock i at price q_i, $i = 1, \ldots, n$. The current price per share of stock i is p_i, $i = 1, \ldots, n$. Ellen Grunbaum expects that the price per share of stock i one year from now will be r_i, $i = 1, \ldots, n$. If she sells any shares, she must pay a transaction cost of 1% of the amount transacted. In addition, she must pay a capital-gains tax at the rate of 30% on any capital gains at the time of the sale. For example, suppose that Ellen Grunbaum sells 1,000 shares of a stock today at \$50 per share, that she had originally purchased at \$30 per share. She would receive \$50,000. However, she would have to pay capital gains taxes of $0.30\times(50{,}000-30{,}000) = \$6{,}000$, and she would have to pay $0.01\times(50{,}000) = \$500$ in transaction costs. Therefore, by selling 1,000 shares of this stock, she would have a net cashflow of $\$50,000 - \$6,000 - \$500 = \$43,500$. Now suppose that Ellen would like to sell enough shares of the stock in her portfolio today, to generate an amount of cash C today for use as a down payment on a home.

The problem of selecting how many shares of which stocks she needs to sell in order to generate the cash amount C, net of capital gains and transaction costs, while maximizing the expected value of her portfolio for next year can be formulated as follows. Let x_i be the number of shares of stock i that she needs to sell.

$$\begin{array}{rl} \textbf{PO}: \text{maximize} & \sum_{i=1}^{n} r_i(s_i - x_i) \\ \text{subject to} & \sum_{i=1}^{n} p_i x_i - 0.3 \sum_{i=1}^{n} (p_i - q_i)x_i - 0.01 \sum_{i=1}^{n} p_i x_i \geq C \\ & 0 \leq x_i \leq s_i, \quad i = 1, \cdots, n \end{array}$$

Suppose that Ellen needs at least \$100,000 for a down payment and that she has the following data about the stocks she is holding.

Stocks i	1	2	3	4	5	6	7	8	9	10
Shares s_i	3,000	200	200	1,500	600	500	1,200	400	200	2,100
Initial prices q_i	50	73	40	80	140	60	82	24	100	33
Current prices p_i	70	60	42	65	160	60	57	50	102	45
Future prices r_i	75	65	40	85	170	60	68	55	100	50

(a) **portfolio.mod** and **portfolio.dat** are an OPL model file and the data file for the above optimization problem, respectively. Download and unzip the zip file **Prob1.zip** from the assignment section and go over the codes and the data to understand how to write an optimization model using OPL and how to associate a data file with a model file. Now create a project file and run the model. What is the optimal objective value? What is the optimal solution?

For MIT students using MIT Server, the following two commands will open OPL Studio. (Please use SUN workstations. SGI machines do not support OPL Studio.)

```
% add oplstudio
% oplst &
```

If you need help in working with OPL Studio, contact the teaching assistant. Also, you might want to look at online reference guides for OPL Studio.

(b) The PO model has one drawback: Suppose $\sum_{i=1}^{n}(p_i - q_i)x_i$ is negative. This means that Ellen has negative capital gains if she sells x_i shares of stock i, $i = 1, \cdots, n$. In this case, Ellen should neither pay taxes nor receive tax savings. In order to reflect this, the PO model should be re-formulated as follows:

$$\begin{array}{lll} \textbf{MPO}: & \text{maximize} & \sum_{i=1}^{n} r_i(s_i - x_i) \\ & \text{subject to} & \sum_{i=1}^{n} p_i x_i - 0.3\ \max\{\sum_{i=1}^{n}(p_i - q_i)x_i, 0\} - 0.01 \sum_{i=1}^{n} p_i x_i \geq C \\ & & 0 \leq x_i \leq s_i, \quad i = 1, \cdots, n \end{array}$$

How would you modify MPO to convert it to a linear program? Modify **portfolio.mod**, solve it using OPL Studio, and report the optimal objective value and the optimal solution. Does the optimal objective value increase or decrease? Why?

2 Radiation Therapy Model (40 points)

The purpose of this question is to give you more experience in using OPL Studio by solving a relatively large-scale linear optimization model, using a variety of different solution methods, and making some changes to the model in the process.

On the assignment section, you will find the zip file **Prob2.zip** which contains the following files:

- **regular5.mod** This is the OPL model file for the "base case" radiation therapy model discussed in the first lecture of the course.
- **small5.mod** This is the "small" version of the radiation therapy model discussed in the first lecture of the course.
- **opl505.dat** This is the data file for the radiation therapy problem.
- **display5.c** This is a C-program that is used to automatically allow the output of the radiation therapy model to be turned into a "picture" that can be displayed in MATLAB.
- **BOUNDARY5.hlp** This is a supplementary file that is used to produce visual images of the model output.

Part I. Run the "base case" model and produce a picture of the optimal solution. In order to do so, proceed as follows:

- First, download all files from the assignment section. Then open OPL Studio.
- Next, create a new project using the model file **regular5.mod** and the data file **opl505.dat**.
- Prior to running the model, click the "Options" menu and the "Customize Active Project Options" submenu. Change the project options in the following four ways:
  - (i) Under "MP General", change the "LP Method" to "Primal"
  - (ii) Under "Simplex", change "Pricing candidate list size" to 200
  - (iii) Under "Simplex", turn "Perturbation" to the on position.
  - (iv) Under "Preprocessing", change "Simplification with pre-solve" to the off position.
- Now you are ready to solve your optimization model. Do so by clicking the green Run button.
- After the model is solved, you will need to save the solution. First, click the green "proceed" button. Then choose the "File" menu, and choose the "Dump Active Model and Result ..." submenu. Give your solution file a name such as "**coolmodel.out**".
- In order to create a picture of the model output, compile the program **display5.c** using the MIT Server commands **add gnu** and **gcc display5.c**. This will create the executable file **a.out**. Execute this file by typing the command **a.out** . You will next be prompted to supply an input file, for which you should type your model solution output file **coolmodel.out** (or whatever name you called your output file). You will need to specify the name of the new output file, such as "**hotmodel**" without an extension.

- Now you can open MATLAB by typing the commands **add matlab** and **matlab**. Once MATLAB is open, type the commands **load hotmodel** and **pcolor(hotmodel)**. MATLAB will produce and display a picture of the optimal solution of the radiation therapy model.

Part II. Experiment with the "base case" and "small" versions of the radiation therapy model. Run the models **regular5.mod** and **small5.mod** using the two different LP methods "Primal" (which is the primal simplex algorithm) and "Barrier" (which is the interior-point logarithmic barrier algorithm described in the second lecture of the course). These solution method options can be found in OPL Studio under "Options" then "Customize Active Project Options" then "MP General". Fill in the following table:

MODEL	Algorithm	Number of Constraints	Number of Variables	Running time (seconds)	Number of Iterations
Base Case	Primal Simplex				
	Barrier				
Small Model	Primal Simplex				
	Barrier				

Part III. Experiment with different versions of the model to see if you can produce solutions that look "better" to you. Here are two ideas for different versions of the model:

- Idea #1. You might want to change the weights on the different pixel regions ($\mathcal{T}$, $\mathcal{C}$ and $\mathcal{N}$) in the objective function, to see if you can produce better solutions. Try to weight more on the tumor part and the critical part, respectively, and see what happens.

- Idea #2. In the first lecture, you were shown several different linear optimization models for this problem. In one of those models, the objective function was to minimize the maximum difference between the delivered dose and target dose at each pixel. This model was:

$$\begin{array}{lll}
\underset{w,D,\mu}{\text{minimize}} & \mu & \\
\text{s.t.} & -\mu \leq D_{ij} - (\text{Target})_{ij} \leq \mu & (i,j) \in \mathcal{S} \\
& D_{ij} = \sum_{p=1}^{n} D_{ij}^{p}\, w_p & (i,j) \in \mathcal{S} \\
& w \geq 0 &
\end{array}$$

With slight changes, you can modify the given model files to become a version of this formulation. How does your solution change? Is it better than any other of your solutions?

If you have any questions about the model, please send an email to the MIT Teaching Assistant.

3 Variable Upper and Lower Bounds (20 points)

Consider the general linear programming problem with upper and lower bounds as follows:

$$\begin{aligned} LP: \quad \text{minimize} \quad & z = c^T x \\ \\ \text{s.t.} \quad & Ax = b \\ & l \leq x \leq u \ . \end{aligned}$$

Suppose that the simplex tableau for such an instance of this problem is as shown in Table 1. Answer the following questions:

UB	8	10	8	3*	6	10
LB	1	2	1	0	2*	1*
RHS	x_1	x_2	x_3	x_4	x_5	x_6
-7	0	0	0	-1	5	4
11	1	0	0	3	-3	6
25	0	1	0	3	4	5
20	0	0	1	2	5	2

Table 1: A simplex tableau.

1. What is the basic solution corresponding to the tableau in Table 1? Why is this solution feasible?

2. Why is this basic solution also optimal?

3. Suppose that the tableau objective coefficient on x_4 is changed from -1 to 1. Then the tableau would look like that shown in Table 2. What would be the next change in the basis for this tableau that would improve the objective function value? Compute the next tableau.

4. Now suppose that the lower bound on x_4 is changed from 0 to 2. Then the tableau would like that shown in Table 3. What would be the next change in the basis for this tableau that would improve the objective function value? Compute the next tableau.

UB	8	10	8	3*	6	10
LB	1	2	1	0	2*	1*
RHS	x_1	x_2	x_3	x_4	x_5	x_6
-7	0	0	0	1	5	4
11	1	0	0	3	-3	6
25	0	1	0	3	4	5
20	0	0	1	2	5	2

Table 2: Another simplex tableau.

UB	8	10	8	3*	6	10
LB	1	2	1	2	2*	1*
RHS	x_1	x_2	x_3	x_4	x_5	x_6
-7	0	0	0	1	5	4
11	1	0	0	3	-3	6
25	0	1	0	3	4	5
20	0	0	1	2	5	2

Table 3: A third simplex tableau.

4 Lagrange Duality Problems (20 points)

A. Construct a Lagrange dual of the following problem:

$$\begin{array}{lll} \mathrm{BP43}: & \mathrm{minimum}_{x,s} & c^T x - 43 \sum_{i=1}^{m} \ln s_j \\ & & \\ & \mathrm{s.t.} & Ax + s = b \\ & & s > 0 \end{array}$$

B. Consider the following problem:

$$\begin{array}{llll} P: & \mathrm{minimize}_{x,y} & c^T x & \\ & \mathrm{s.t.} & Gx & \leq f \\ & & Nx + My & \leq d \\ & & y \ \text{ binary integer} & \end{array}$$

It is helpful in this problem to think of the variables x as production decisions and the binary integer variables y as set-up decisions. The constraints "$Gx \leq f$" are ordinary resource and/or capacity constraints on the production levels x, and the constraints "$Nx + My \leq d$" can be thought of as arising from the effects of set-up decisions y on the production activities x.

x^0	d	α^u
$(5, 7)$	$(1, -2)$	2.3
$(-30, -50)$	$(1, 1)$	1.5
$(20, -10)$	$(-1, 0)$	5.6

Table 4: Starting points, directions, and step-size upper bound.

(a) Construct a "good" dual of this problem. What makes your choice of dual problem "good"?

(b) For a given value of your dual variables u, how would you solve for the value of $L^*(u)$?

(c) What type of algorithm might you use to solve the dual problem?

5 Bisection Line-Search Algorithm (15 points)

Consider the function:

$$\begin{aligned} f(x) &= 25 - 4x_1 - 6x_2 + (3x_1 - 6x_2 - 10)^2 + (x_1 + x_2 + 15)^2 \\ &+ (x_1 - x_2 + 4)^4 + \tfrac{1}{10}(5x_1 + x_2 - 6)^6 \ . \end{aligned}$$

Given a current point $x^0 = (x_1^0, x_2^0)$ and a direction $d = (d_1, d_2)$, we are interested in solving the 1-dimensional minimization problem:

$$P : \qquad \min_\alpha \quad h(\alpha) := f(x^0 + \alpha d)$$

- Construct a bisection algorithm, coded either in OPLScript or Matlab, to solve this problem. The algorithm should be designed to take as input a starting point x^0, a direction d, and an initial value for the upper bound α_u. Be certain to specify a stopping criteria for your algorithm. Your code should check whether or not d is a descent direction (and terminate if it is not), and should check that α_u is a valid upper bound on the optimal step-size $\bar{\alpha}$.
- Solve for the minimizing value of α using your algorithm for the three problem instances shown in Table 4.
- Hand in a hard copy of your code.