Jacobs University
Fall 2022

November 10, 2022

Operations Research

Midterm Exam (make-up)

(For bonus points only.)

Instructions:

- Do all the work on this exam paper.
- Show your work, i.e., carefully write down the steps of your solution. You will receive points not just based on your final answer, but on the correct steps in your solution.
- No tools or other resources are allowed for this exam. In particular, no notes and no calculators.

Name: Solutions

Matric. No.: ____________________

Problem 1: Graphical Method [25 points]
Consider the following Linear Programming problem: Maximize

$$Z = 3x_1 + 4x_2$$

subject to

$$\begin{aligned} (1) \quad -x_1 + 2x_2 &\leq 12, \\ (2) \quad 4x_1 + x_2 &\leq 24, \\ (3) \quad x_1 + 4x_2 &\geq 16, \quad \text{(note the } \geq \text{ sign)} \\ x_1, x_2 &\geq 0. \end{aligned}$$

(8) (a) Solve the problem with the graphical method, i.e., draw the feasible region, and find the optimal solution and the corresponding value of Z.

(6) (b) From part (a), identify the two binding constraints. Using this information, compute the optimal solution and the corresponding value of Z.

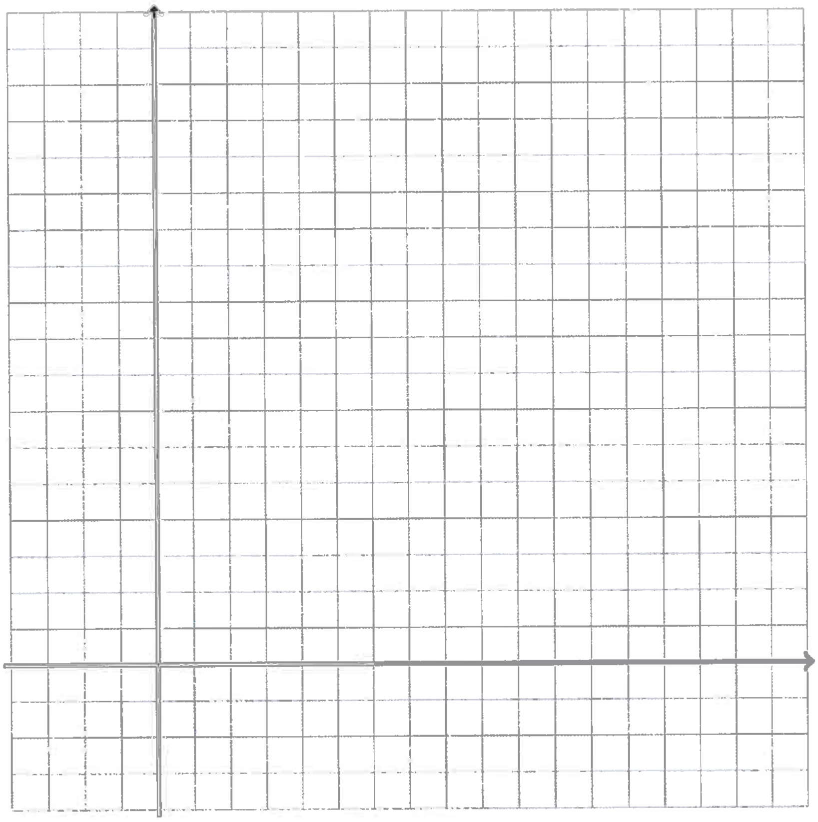
(4) (c) Using part (b), determine which one of the following **cannot** be the shadow prices for this problem? Briefly explain your answer.

- $(y_1, y_2, y_3) = (\frac{13}{9}, \frac{10}{9}, 0)$,
- $(y_1, y_2, y_3) = (0, 0, \frac{11}{4})$,
- $(y_1, y_2, y_3) = (\frac{13}{9}, 0, \frac{10}{9})$,
- $(y_1, y_2, y_3) = (0, \frac{13}{9}, \frac{10}{9})$.

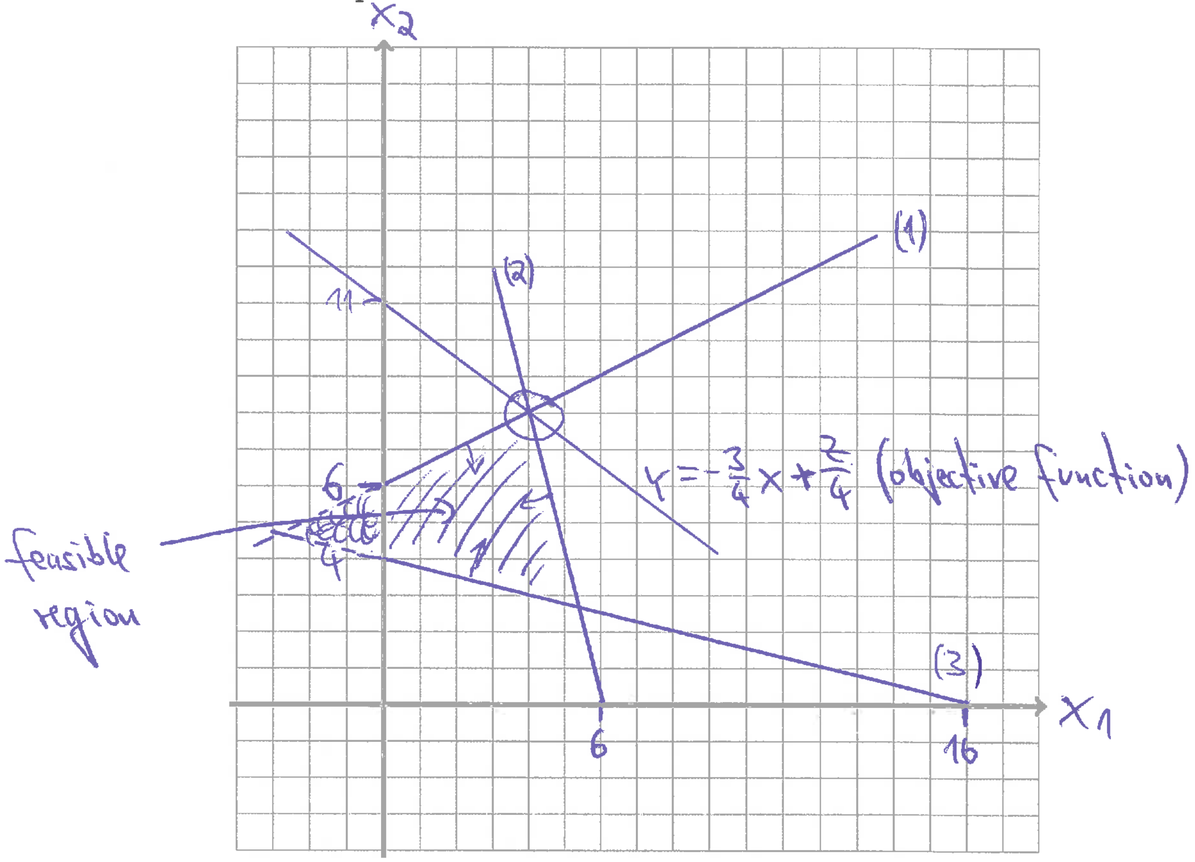
(3) (d) Now suppose the constraint $x_1 \geq 0$ is removed, i.e., x_1 can attain any real value. Does this change the optimal solution? If yes, explain what the new optimal solution is. If no, explain why not.

(4) (e) Now suppose the constraint $4x_1 + x_2 \leq 24$ is removed. Does this change the optimal solution? Explain your answer.

Problem 1: Extra Space



Problem 1: Extra Space



a) The optimal solution is $x_1 = 4$, $x_2 = 8$, with $z = 44$.

b) (1) and (2) are the binding constraints.

=> Need to solve $-x_1 + 2x_2 = 12$

$4x_1 + x_2 = 24$

=> $9x_2 = 72 \Rightarrow x_2 = 8 \Rightarrow 4x_1 = 16 \Rightarrow x_1 = 4$

=> $(x_1, x_2) = (4, 8)$ with $z = 3 \cdot 4 + 4 \cdot 8 = 44$

c) The 2nd, 3rd, 4th possibilities cannot be the shadow prices, because constraint (3) is non-binding, i.e., y_3 has to be equal 0.

d) This would turn the feasible region into a triangle as in the picture above. Therefore the optimal solution would not change.

Problem 1: Extra Space

e) This would make the feasible region unbounded. And z increases in the unbounded direction, so there would be ∞ many feasible, but no optimal solution.

Problem 2: Standard Form and Simplex Method [25 points]

For some given LP problem the following simplex tableau with decision variables x_1, x_2, and slack variables s_1, s_2 was set up (in the standard form discussed in class):

x_1	x_2	s_1	s_2	
4	1	1	0	20
1	4	0	1	20
−1	−2	0	0	0

(5) (a) Write down the original LP problem in mathematical notation.

(20) (b) Solve the problem with the simplex method as shown in class. *(Note: You will not receive points for simply stating the solution, but only for using the simplex method step by step.)*

a) Original LP problem:

Maximize $z = x + 2y$,

with constraints $4x + y \leq 20$,

$x + 4y \leq 20$, and $x, y \geq 0$.

b)

$$\begin{array}{cccc|c} 4 & 1 & 1 & 0 & 20 \\ 1 & 4 & 0 & 1 & 20 \\ \hline -1 & -2 & 0 & 0 & \end{array} \Longrightarrow \begin{array}{r|cccc|c} -\frac{1}{4}R_2 + R_1 & \frac{15}{4} & 0 & 1 & -\frac{1}{4} & 15 \\ \frac{R_2}{4} & \frac{1}{4} & 1 & 0 & \frac{1}{4} & 5 \\ \hline \frac{R_2}{2} + R_3 & -\frac{1}{2} & 0 & 0 & \frac{1}{2} & 10 \end{array}$$

$$\Longrightarrow \begin{array}{r|cccc|c} \frac{4}{15} \cdot R_1 & 1 & 0 & \frac{4}{15} & -\frac{1}{15} & 4 \\ -\frac{1}{15}R_1 + R_2 & 0 & 1 & -\frac{1}{15} & \frac{4}{15} & 4 \\ \hline \frac{2}{15}R_1 + R_3 & 0 & 0 & \frac{2}{15} & \frac{7}{15} & 12 \end{array}$$

$\Longrightarrow$ optimal solution is $x_1 = 4$, $x_2 = 4$, with $z = 12$.

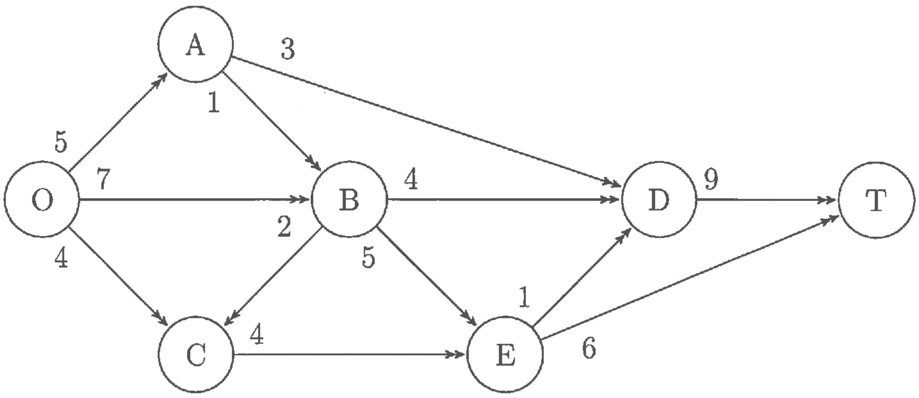
Problem 2: Extra Space

Problem 2: Extra Space

Problem 2: Extra Space

Problem 3: Network optimization [25 points]

Consider the following network (the Seervada Park problem that we discussed in class). The numbers at the arcs are capacity limits for the (visitor) flow.

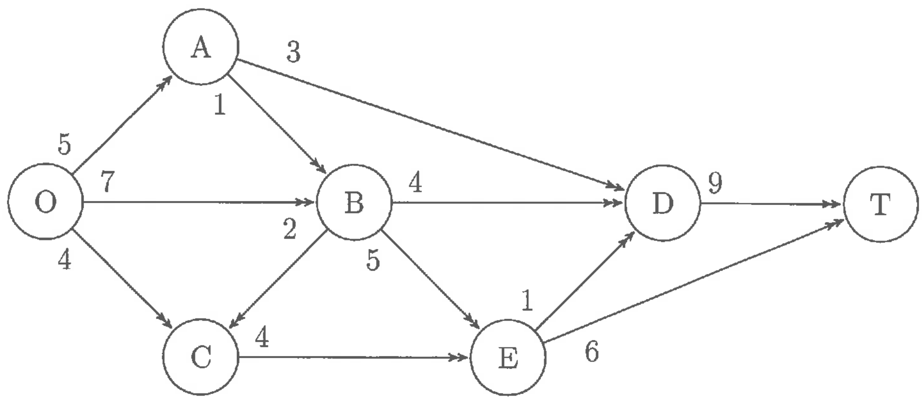


Find the maximum flow from the origin "O" to the target "T" using the "Augmenting Path Algorithm" discussed in class. *(Note: You will not receive points for simply stating the solution, but only for using the "Augmenting Path" method step by step.)*

See Session 16 lecture notes.

Problem 3: Extra Space

Problem 3: Extra Space



Problem 3: Extra Space

Problem 4: Pyomo [25 points]

The pyomo program on the last exam page shows an implementation of a minimum cost flow problem (which we discussed in class).

(6) (a) Draw the corresponding network representation of the problem. (Make sure to include the given values of the parameters from the pyomo program.)

(5) (b) Write down the Linear Programming problem that is solved in mathematical notation (i.e., write down the objective function and the constraints).

(5) (c) Briefly explain the meaning of the parameters b, C, U in the context of minimum cost flow problems. In particular, explain what positive, negative, or zero values for entries in b mean.

(3) (d) Suppose one of the entries in U can be slightly increased. Which one should we choose in order to lower the costs? (You will receive points only if you justify your answer correctly, and not for guessing.)

(3) (e) Suppose one of the entries F1 or F2 in b is increased by one unit, and simultaneously one of the entries W1 or W2 in b is decreased by one unit. Which entries should be increased/decreased in order to be most cost effective? What will be the extra cost for the increase/decrease?

(3) (f) Management notices that there is some flexibility between F1 and F2 concerning the units of b, i.e., F1 could be increased by a few units while simultaneously decreasing F2 by the same number of units, or the other way around. Which of the following three solutions should we recommend in order to be most cost effective:

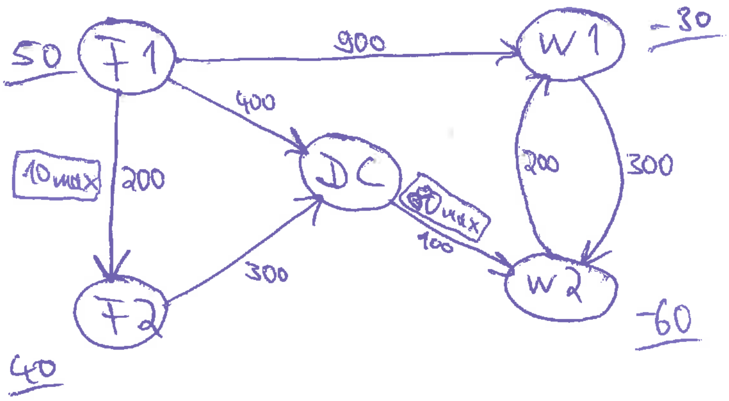
1. Increase F1 by a few units and decrease F2.
2. Decrease F1 by a few units and increase F2.
3. Do not change the values of F1 and F2.

Explain your answer. (You will not receive points for a random choice.)

Problem 4: Extra Space

Problem 4: Extra Space

a)



b) Minimize $Z = \sum_{(i,j)\in A} c_{ij} x_{ij}$,

with constraints $\sum_j x_{ij} - \sum_j x_{ji} = b_i$,

and $0 \le x_{ij} \le u_{ij} \quad \forall (i,j) \in A$.

c) b_i = supply/demand at node i

↳ $b_i > 0$ ⟹ supply/source node
↳ $b_i < 0$ ⟹ demand/sink node
↳ $b_i = 0$ ⟹ transshipment node

c_{ij} = cost of transportation from i to j

u_{ij} = maximum capacity on arc (i,j)

d) We look at the shadow prices for the capacity constraints after In[10]. Since the price for DC→W2 is negative, we should increase $u_{DC,W2}$ in order to decrease the cost.

Problem 4: Extra Space

e) We choose the combination with the smallest shadow price. Here, F2 should be increased (price 600), while W2 should be decreased (price 0). The extra cost will be 600. (Note: Decreasing W1 instead would add (−1)(−200) = +200 to the cost.)

f) The shadow price of F1 is 700, while the shadow price of F2 is 600. Thus, decreasing b at F1 and increasing at F2 will cost −700 + 600 = −100, i.e., it will save 100 cost per unit.

In [1]:
```
from pyomo.environ import *
from pyomo.opt import *
opt = solvers.SolverFactory("glpk")
```

In [2]:
```
b = {'F1':50,
     'F2':40,
     'DC':0,
     'W1':-30,
     'W2':-60}

C = {('F1','F2'):200,
     ('F1','DC'):400,
     ('F1','W1'):900,
     ('F2','DC'):300,
     ('DC','W2'):100,
     ('W1','W2'):300,
     ('W2','W1'):200}

U = {('F1','F2'):10,
     ('DC','W2'):80}

N = list(b.keys())
A = list(C.keys())
V = list(U.keys())
```

In [3]:
```
model = ConcreteModel()
model.x = Var(A, within=NonNegativeReals)
```

In [4]:
```
def flow_rule(model, n):
    InFlow  = sum(model.x[i,j] for (i,j) in A if j==n)
    OutFlow = sum(model.x[i,j] for (i,j) in A if i==n)
    return OutFlow - InFlow == b[n]

model.flow = Constraint(N, rule=flow_rule)
```

In [5]:
```
def capacity_rule(model, i, j):
    return model.x[i,j] <= U[i,j]

model.capacity = Constraint(V, rule=capacity_rule)
```

In [6]:
```
model.cost = Objective(expr = sum(model.x[a]*C[a] for a in A), sense=minimize)
```

In [7]:
```
model.dual = Suffix(direction=Suffix.IMPORT)
results = opt.solve(model)
model.x.get_values()
```

Out[7]:
```
{('F1', 'F2'): 0.0,
 ('F1', 'DC'): 40.0,
 ('F1', 'W1'): 10.0,
 ('F2', 'DC'): 40.0,
 ('DC', 'W2'): 80.0,
 ('W1', 'W2'): 0.0,
 ('W2', 'W1'): 20.0}
```

In [8]:
```
model.cost.expr()
```

Out[8]:
```
49000.0
```

In [9]:
```
for n in N:
    print(n, model.dual[model.flow[n]])
```

```
F1 700.0
F2 600.0
DC 300.0
W1 -200.0
W2 0.0
```

In [10]:
```
for j in V:
    print(j, model.dual[model.capacity[j]])
```

```
('F1', 'F2') 0.0
('DC', 'W2') -200.0
```