Jacobs University
Fall 2021

January 26, 2022

Operations Research

Final Exam (Make-up)

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.
- Note that there are five problems, with a total of 125 points. The grade will be computed out of 100 points only, i.e., 25 of the possible exam points are counted as bonus.

Name: _____

Problem 1: Graphical Method [30 points]

Consider the following Linear Programming problem: Maximize

$$Z = x_1 + x_2$$

subject to

$$\begin{aligned} -x_1 + x_2 &\leq 4, \\ \frac{1}{2}x_1 + x_2 &\leq 7, \\ x_1 &\leq 6, \\ x_1, x_2 &\geq 0. \end{aligned}$$

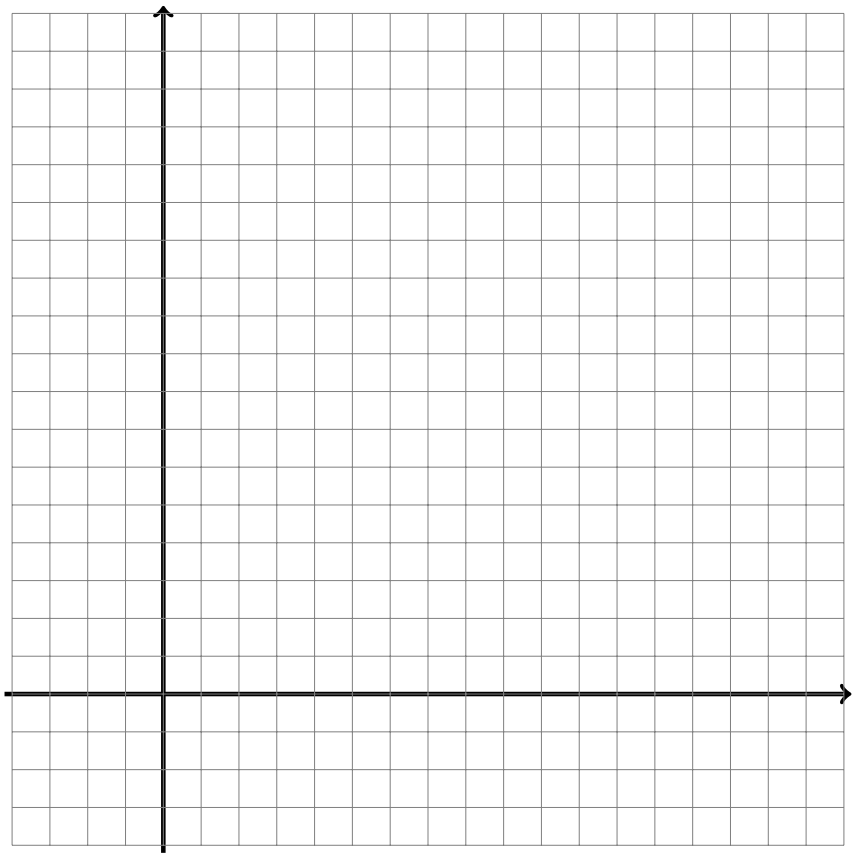
(a) Solve the problem with the graphical method (i.e., find the optimal solution, and the value of Z at the optimal solution graphically). Then confirm your answer by identifying the binding constraints and computing your answer from there.

(b) Now suppose the second constraint ($\frac{1}{2}x_1 + x_2 \leq 7$) is removed. How does the optimal solution change? (Explicitly state the new optimal x_1, x_2 and the corresponding optimal value of Z.)

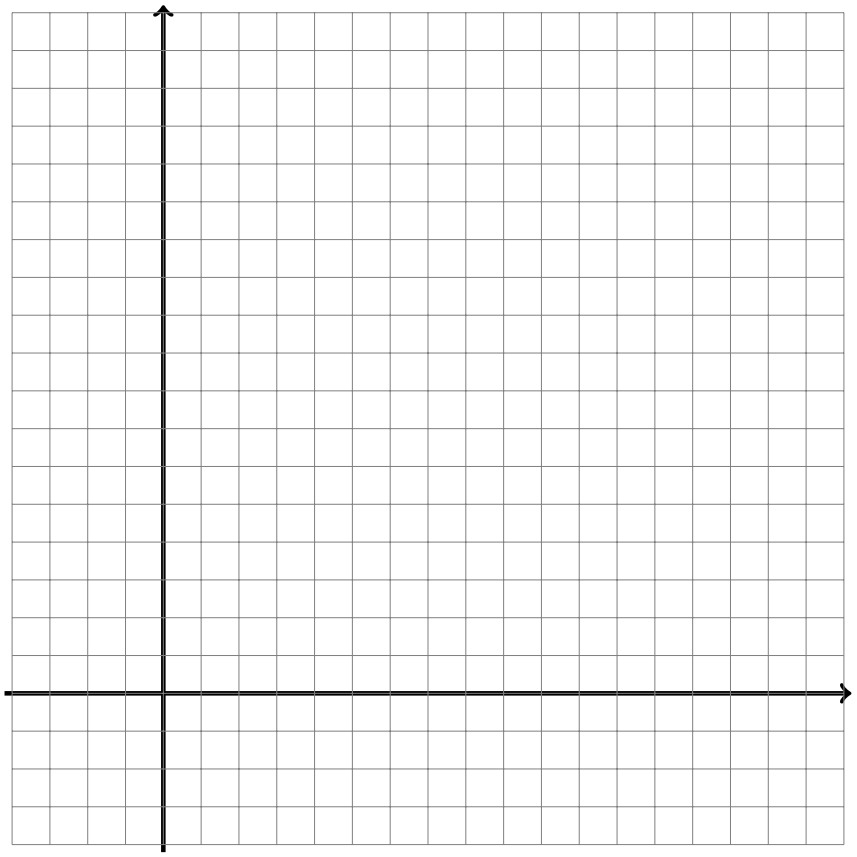
(c) Now suppose the objective function is changed to $Z = mx_1 + x_2$ for some parameter $m > 0$. Discuss what all the optimal solutions are depending on the possible values of $m > 0$.

(d) Write down the dual LP problem to the LP problem stated above.

(e) Let us define $c^{\mathrm{T}} = (1, 1)$ and $b^{\mathrm{T}} = (4, 7, 6)$. Let x be a feasible solution to the LP problem above, and let y be a feasible solution to the dual LP problem. Show that then $c^T x \leq b^T y$.



Problem 1: Extra Space



Problem 1: Extra Space

Problem 2: Standard Form and Simplex Method [20 points]

Consider the following Linear Programming problem: Maximize

$$Z = 3x_1 + 5x_2$$

subject to

$$\begin{aligned} x_1 &\leq 4\,, \\ x_2 &\leq 6\,, \\ 3x_1 + 2x_2 &\leq 18\,, \\ x_1, x_2 &\geq 0\,. \end{aligned}$$

(a) Write the problem in standard form, i.e., as the problem to minimize

$$\tilde{Z} = c^{\mathrm{T}}\tilde{x}$$

subject to

$$A\tilde{x} = b, \quad \tilde{x} \geq 0.$$

Specify exactly the vectors b, c and the matrix A.

(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.)*

Problem 2: Extra Space

Problem 2: Extra Space

Problem 2: Extra Space

Problem 3: Dynamic Programming [25 points]

Consider the following *integer* nonlinear programming problem.

$$\text{Maximize } Z = 32\,x_1 - 2\,x_1^2 + 30\,x_2 + 20\,x_3$$

subject to

$$3\,x_1 + 7\,x_2 + 5\,x_3 \leq 20$$

and

$$x_1, x_2, x_3 \text{ are nonnegative integers.}$$

Use dynamic programming to solve this problem. *(Note: You will not get points for simply stating the correct solution; the exercise here is to solve the problem with the dynamic programming method.)*

Problem 3: Extra Space

Problem 3: Extra Space

Problem 3: Extra Space

Problem 4: Advanced Topics [25 points]

(a) Consider the decision analysis problem with the following payoff table:

Alternative	**State of Nature** S_1	S_2	S_3
A_1	-100	10	100
A_2	-10	20	50
A_3	10	10	60
Prior Probability	0.2	0.3	0.5

Which alternative should be chosen? What is the resulting expected payoff?

(b) Let us consider a basic inventory management problem with the following assumptions:

- The setup cost per order is called K; The cost per unit is called c;
- The holding (storage) cost is h per unit per time in inventory;
- There is a constant withdrawal rate of m units per time;
- We do not allow for shortages;
- The inventory level is continuously checked (continuous review).

Find the optimal order quantity Q^* that minimizes the cost per time.

(Hint: This exercise is to derive the EOQ formula. Proceed as we did in class, i.e.: Determine the cost per cycle (a picture might be helpful), then the total cost per time, and then find the minimum.)

(c) We consider the following nonlinear optimization problem: Maximize

$$Z = x_1 + x_2$$

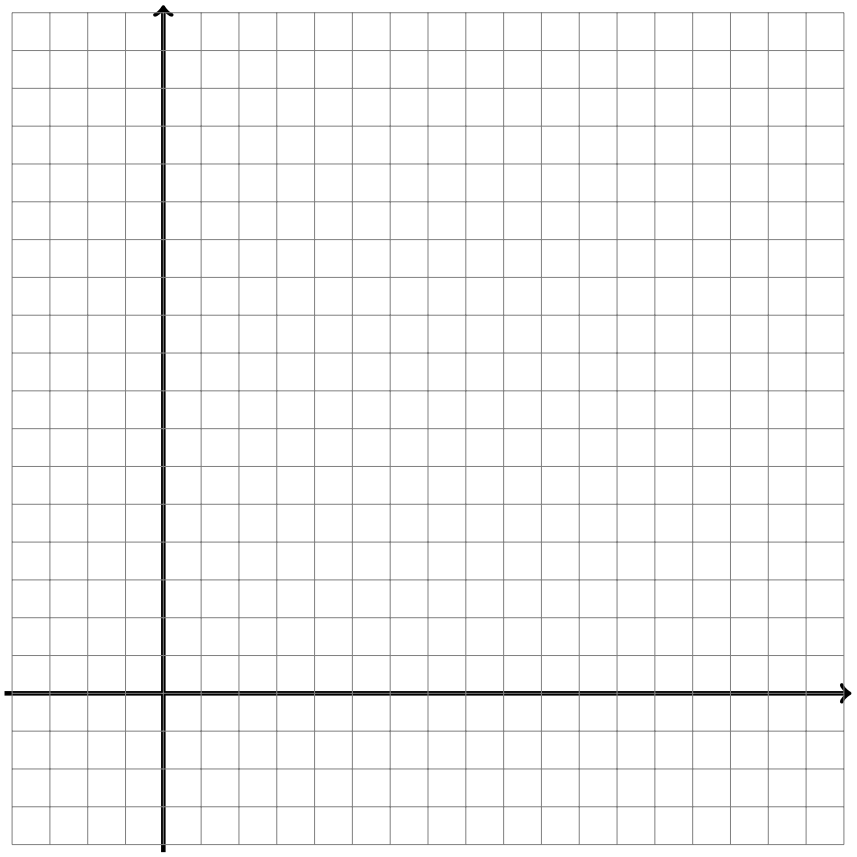
subject to

$$x_1^2 + x_2^2 \leq 1\,,$$
$$x_1, x_2 \geq 0\,.$$

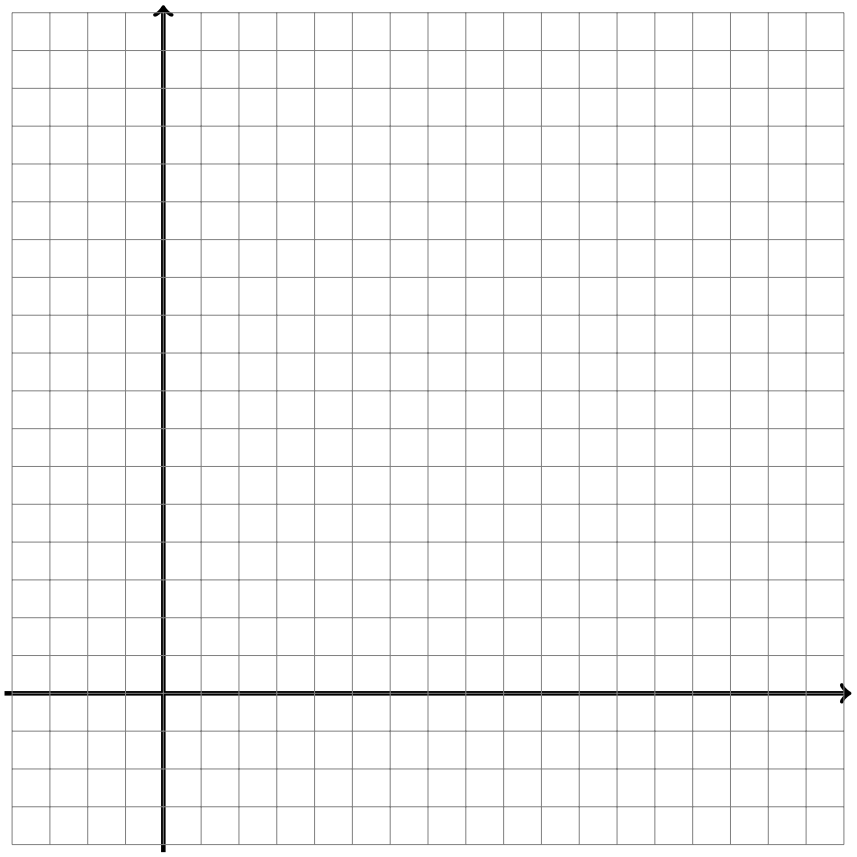
(i) Use the graphical method to solve this problem.

(ii) Is this a convex optimization problem? (Briefly justify your answer.) What would convexity or non-convexity imply for solvers such as ipopt (that we discussed in class)?

Problem 4: Extra Space



Problem 4: Extra Space



Problem 4: Extra Space

Problem 5: 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).

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

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

(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.

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

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

(f) Suppose the optimal flows in this problem can only be done with n units at the same time. What are the possible numbers n here? How and why can we find the biggest such number n in general, given the parameters b, C, U?

Problem 5: Extra Space

Problem 5: Extra Space

Problem 5: Extra Space

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.f = Var(A, within=NonNegativeReals)
```

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

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

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

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

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

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