

Operations Research

Midterm Exam

(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: _____

Problem 1: Graphical Method [20 points]

Consider the following Linear Programming problem: Maximize

$$Z = x_1 + x_2$$

subject to

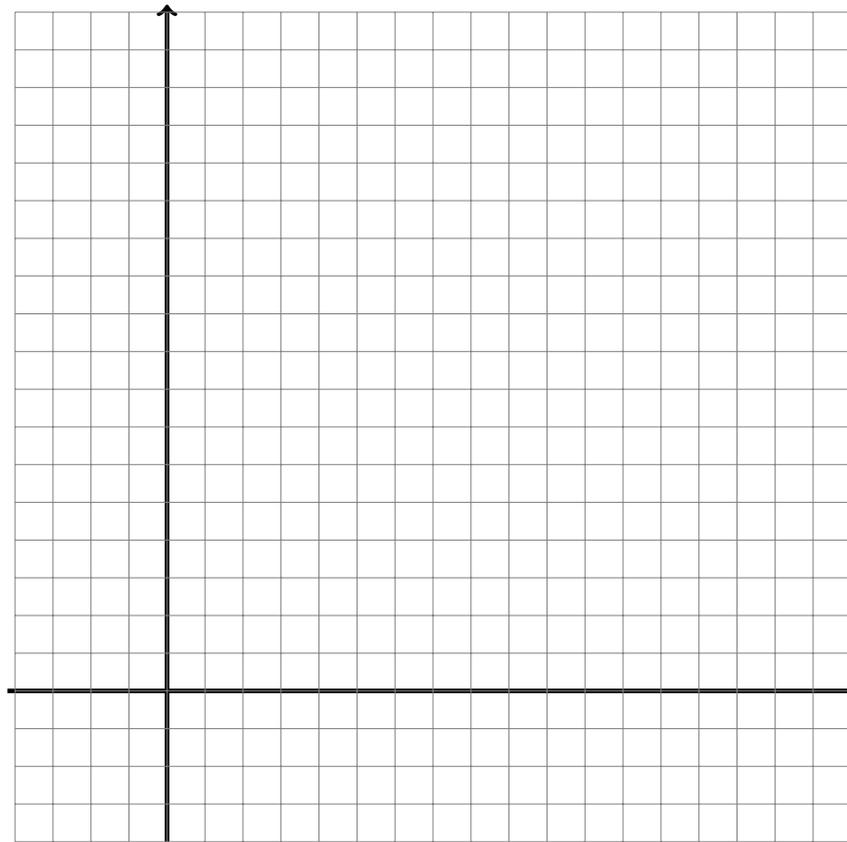
$$-x_1 + x_2 \leq 4,$$

$$\frac{1}{2}x_1 + x_2 \leq 7,$$

$$x_1 \leq 6,$$

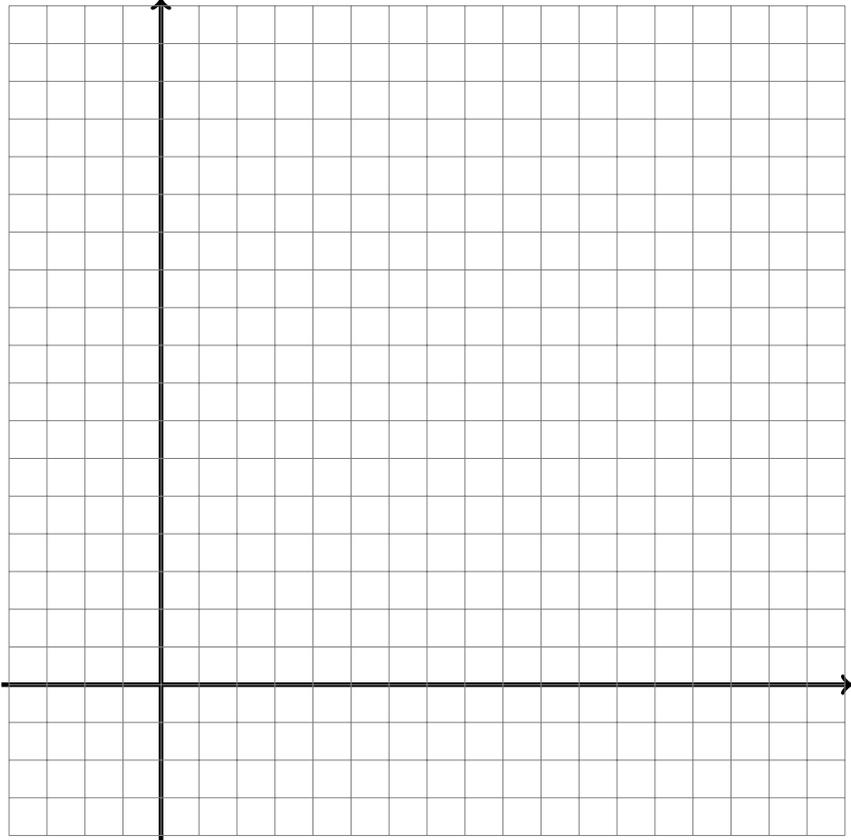
$$x_1, x_2 \geq 0.$$

- (a) Solve the problem with the graphical method (i.e., find the optimal solution, and the value of Z at the optimal solution graphically).
- (b) Is the feasible region bounded or unbounded? Does the answer change if one of the first three constraints is removed?
- (c) Now suppose the objective function is changed to $Z = cx_1 + x_2$ for some parameter $c > 0$. Discuss what all the optimal solutions are depending on the possible values of $c > 0$.

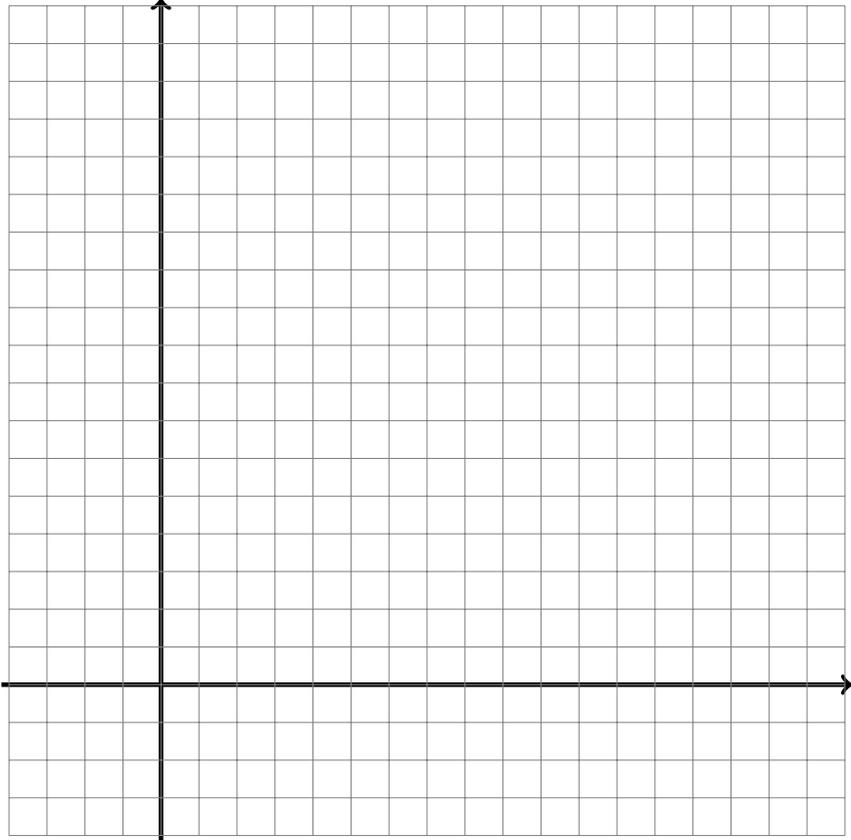


Problem 1: Extra Space

Problem 1: Extra Space



Problem 1: Extra Space



Problem 2: Standard Form and Simplex Method [35 points]

(a) Consider the following Linear Programming problem: Maximize

$$Z = 2x_1 + x_2 - 3x_3$$

subject to

$$\begin{aligned}x_1 + x_2 &\leq 40, \\4x_1 + x_2 &\leq 100, \\x_1 + x_2 + x_3 &\geq 20, \\x_1, x_2 &\geq 0,\end{aligned}$$

and no further restriction on the sign of x_3 . Write this problem in standard form, i.e., as the problem to minimize

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

subject to

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

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

(b) Consider now the following simpler Linear Programming problem: Maximize

$$Z = 2x_1 + x_2$$

subject to

$$\begin{aligned}x_1 + x_2 &\leq 40, \\4x_1 + x_2 &\leq 100, \\x_1, x_2 &\geq 0.\end{aligned}$$

Solve the problem using the simplex method. In the end, specify the optimal solution and the value of the objective function at the optimal solution.

(c) Now write down the dual LP problem to the problem from part (b). Without solving the dual problem explicitly, what is the value of $b^T y$, where $b^T = (40, 100)$, and y is the optimal solution of the dual problem?

Problem 2: Extra Space

Problem 2: Extra Space

Problem 2: Extra Space

Problem 3: Network optimization [15 points]

The Wirehouse Lumber Company will soon begin logging eight groves of trees in the same general area. Therefore, it must develop a system of dirt roads that makes each grove accessible from every other grove. The distance (in miles) between every pair of groves is as follows:

	Distance between Pairs of Groves							
	1	2	3	4	5	6	7	8
Grove 1	—	1.3	2.1	0.9	0.7	1.8	2.0	1.5
Grove 2	1.3	—	0.9	1.8	1.2	2.6	2.3	1.1
Grove 3	2.1	0.9	—	2.6	1.7	2.5	1.9	1.0
Grove 4	0.9	1.8	2.6	—	0.7	1.6	1.5	0.9
Grove 5	0.7	1.2	1.7	0.7	—	0.9	1.1	0.8
Grove 6	1.8	2.6	2.5	1.6	0.9	—	0.6	1.0
Grove 7	2.0	2.3	1.9	1.5	1.1	0.6	—	0.5
Grove 8	1.5	1.1	1.0	0.9	0.8	1.0	0.5	—

Management now wishes to determine between which pairs of groves the roads should be constructed to connect all groves with a minimum total length of road. Solve this problem.

Problem 3: Extra Space

Problem 3: Extra Space

Problem 3: Extra Space

Problem 4: Pyomo [30 points]

The pyomo program on the next page shows an implementation of a transportation problem.

- (a) Draw a network representation of the problem.
- (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 C, W, d, s, c in the context of transportation problems.
- (d) Would the value of the objective function (in line “Out [30]”) change if we replace \leq and \geq in the two functions “supply_rule” and “demand_rule” by $=$?
- (e) Suppose the supply in one of the entries of C can be increased by 1 unit. Which one should we choose for the supply increase to be most cost effective? To which entry in W should we ideally deliver the extra supply (to be most cost effective)?
- (f) In the standard form of the dual problem we discussed in class the dual variables were constrained to be positive. Explain how it is possible then that the dual variables at the bottom (under “In [32]”) are negative in this example. How should we fix our code in order to make all dual variables positive?
- (g) Suppose the shipping from the entries in C to the entries in W 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 d, s, c ?

Problem 4: Extra Space

Problem 4: Extra Space

Problem 4: Extra Space

```
In [24]: from pyomo.environ import *
from pyomo.opt import *
opt = solvers.SolverFactory("glpk")
```

```
In [25]: C = [1, 2, 3]
W = [1, 2, 3, 4]
d = {1:80, 2:65, 3:70, 4:85}
s = {1:75, 2:125, 3:100}
c = {(1,1):464, (1,2):513, (1,3):654, (1,4):867,
      (2,1):352, (2,2):416, (2,3):690, (2,4):791,
      (3,1):995, (3,2):682, (3,3):388, (3,4):685}

model = ConcreteModel()
```

```
In [26]: model.x = Var(C, W, within=NonNegativeReals)
```

```
In [27]: model.z = Objective(expr = sum(c[i,j]*model.x[i,j] for i in C for j in W), sense=minimize)
```

```
In [28]: def supply_rule (model, i):
return sum(model.x[i,j] for j in W) <= s[i]

model.supply = Constraint(C, rule=supply_rule)

def demand_rule (model, j):
return sum(model.x[i,j] for i in C) >= d[j]

model.demand = Constraint(W, rule=demand_rule)
```

```
In [29]: model.dual = Suffix(direction=Suffix.IMPORT)
results = opt.solve(model)
model.x.get_values()
```

```
Out[29]: {(1, 1): 0.0,
(1, 2): 20.0,
(1, 3): 0.0,
(1, 4): 55.0,
(2, 1): 80.0,
(2, 2): 45.0,
(2, 3): 0.0,
(2, 4): 0.0,
(3, 1): 0.0,
(3, 2): 0.0,
(3, 3): 70.0,
(3, 4): 30.0}
```

```
In [30]: model.z.expr()
```

```
Out[30]: 152535.0
```

```
In [31]: for j in W:
print(j, model.dual[model.demand[j]])
```

```
1 449.0
2 513.0
3 570.0
4 867.0
```

```
In [32]: for i in C:
print(i, model.dual[model.supply[i]])
```

```
1 0.0
2 -97.0
3 -182.0
```