

Operations Research

Final Exam

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 (the Wyndor company problem from class): Maximize

$$Z = 3x_1 + 5x_2$$

subject to

$$\begin{aligned} x_1 &\leq 4, \\ 2x_2 &\leq 12, \\ 3x_1 + 2x_2 &\leq 18, \\ x_1, x_2 &\geq 0. \end{aligned}$$

- (a) Use the graphical method to find the binding constraints. Then, use the binding constraints to compute the optimal solution.
- (b) Now suppose one of the first three constraints is removed. For each of them, how does the optimal solution change? (Explicitly state the new optimal x_1, x_2 and the corresponding optimal value of Z , for each of the three cases.)
- (c) Write the Linear Programming problem above in standard form, i.e., as the problem to minimize

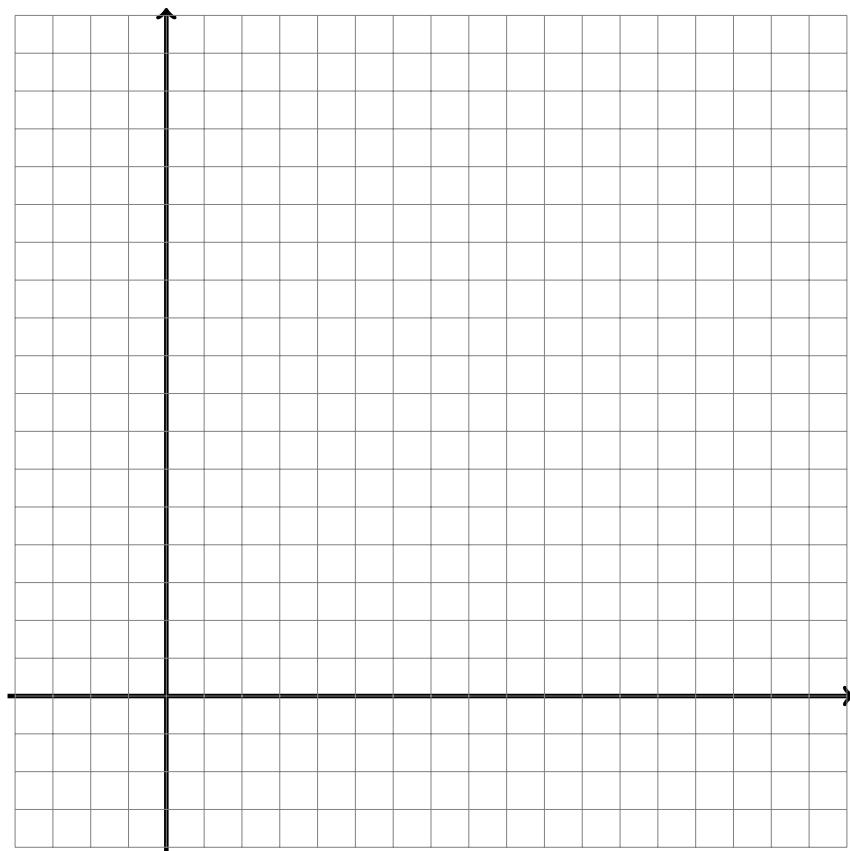
$$\tilde{Z} = c^T \tilde{x}$$

subject to

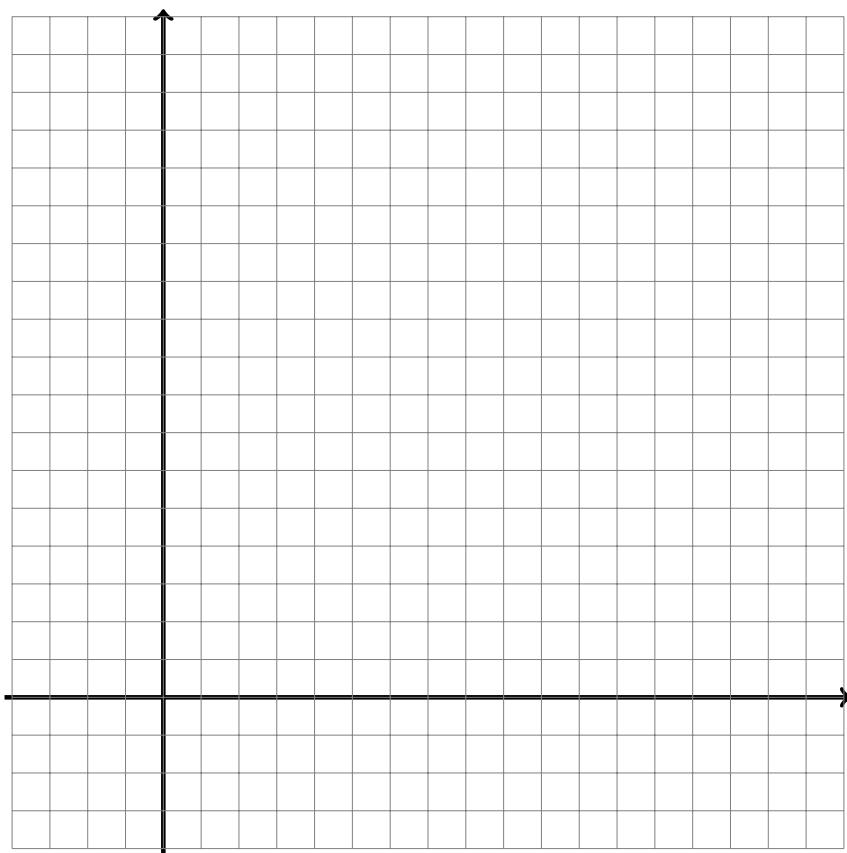
$$A\tilde{x} = b, \quad \tilde{x} \geq 0.$$

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

- (d) Now write down the dual LP problem. Without solving the dual problem explicitly, what is the value of $b^T y$, where $b^T = (4, 12, 18)$, and y is the optimal solution of the dual problem?



Problem 1: Extra Space



Problem 1: Extra Space

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

Consider the following Linear Programming problem: Maximize

$$Z = x_1 + 2x_2$$

subject to

$$\begin{aligned}x_1 + 3x_2 &\leq 8, \\x_1 + x_2 &\leq 4, \\x_1, x_2 &\geq 0.\end{aligned}$$

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

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

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]

A college student has 7 days remaining before final examinations begin in her four courses, and she wants to allocate this study time as effectively as possible. She needs at least 1 day on each course, and she likes to concentrate on just one course each day, so she wants to allocate 1, 2, 3, or 4 days to each course. Having recently taken an OR course, she decides to use dynamic programming to make these allocations to maximize the total grade points to be obtained from the four courses. She estimates that the alternative allocations for each course would yield the number of grade points shown in the following table.

Study Days	Estimated Grade Points			
	Course 1	Course 2	Course 3	Course 4
1	1	5	4	4
2	3	6	6	4
3	6	8	7	5
4	8	8	9	8

Solve this problem by dynamic programming. (*Note: This problem has two optimal solutions, make sure you find both of them. Also note that 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: Decision Analysis [25 points]

Consider the decision analysis problem with the following payoff table.

Alternative	Payoff in State S_i	
	S_1	S_2
A_1	700	-100
A_2	90	90
Prior Probability	1/4	3/4

- (a) Compute the expected payoff (with the prior probabilities) for each alternative. Given the result, which alternative should be chosen?
- (b) Next, some experimentation is done in order to improve the decision making. The outcome of the experimentation can be positive (“pos”) or negative (“neg”). The probabilities that the outcome is positive/negative given state S_1/S_2 are given by

$$\begin{aligned} P(\text{pos} | S_1) &= 0.6, \\ P(\text{pos} | S_2) &= 0.2, \\ P(\text{neg} | S_1) &= 0.4, \\ P(\text{neg} | S_2) &= 0.8. \end{aligned}$$

- (i) Compute the total probabilities $P(\text{pos})$ and $P(\text{neg})$ for the outcome to be positive or negative. (*Note: The result for negative is $P(\text{neg}) = 0.7$; points for this exercise will be given for the correct computation.*)
- (ii) Use Bayes’ rule to compute the probabilities that state S_1 is realized, given a positive/negative outcome of the experimentation, i.e., compute $P(S_1 | \text{pos})$ and $P(S_1 | \text{neg})$. (*Note: The results are $P(S_1 | \text{pos}) = 0.5$ and $P(S_1 | \text{neg}) = 1/7$; points for this exercise will be given for the correct computation.*)
- (c) Given the results of part (b), set up the decision tree for this problem. (The expected payoffs for each step do *not* need to be computed!) The tree needs to include the possibilities “with/without experimentation”, “positive/negative outcome”, “alternative A_1/A_2 ”, “state S_1/S_2 ”. The tree should include a variable c for the cost of experimentation, the necessary probabilities, and the total payoffs at the end only (no intermediate payoffs need to be computed).

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 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 5: Extra Space

Problem 5: Extra Space

Problem 5: 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
```