Operations Research Exam

2h45m

This exam consists of 6 exercises, from which 100 points can be obtained in total. The division of the points over the various parts is as follows:

Question:	1	2	3	4	5	6	Total
Points:	25	15	20	20	10	10	100
Score:							

Exam grade: $\frac{\text{total number of points}}{10}$

Final course grade: $\frac{1}{4}$Weekly pretest grade + $\frac{3}{4}$Exam grade

---

1. Consider the following LP, which we will refer to as the "primal LP".

$$\begin{aligned} \max \quad & z = 4x_1 - x_2 + 3x_3 \\ \text{s.t.} \quad & 2x_1 + 2x_2 + x_3 + x_4 = 5 \\ & x_1 + 3x_2 + x_3 - x_4 \leq 15 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

(a) (10 points) Determine the dual of this LP.

Solution:

$$\begin{aligned} \min \quad & w = 5y_1 + 15y_2 \\ \text{s.t.} \quad & 2y_1 + y_2 \geq 4 \\ & 2y_1 + 3y_2 \geq -1 \\ & y_1 + y_2 \geq 3 \\ & y_1 - y_2 \geq 0 \\ & y_1 \text{ unrestricted} \\ & y_2 \geq 0 \end{aligned}$$

(b) (5 points) The primal LP has been solved using the simplex method, yielding the final tableau below.

Basic	z	x_1	x_2	x_3	x_4	s_2	value
z	1	2	7	0	3	0	15
x_3	0	2	2	1	1	0	5
s_2	0	-1	1	0	-2	1	10

Write down the primal optimal solution and objective value. Also state the objective value of an optimal dual solution (you don't need to give the actual optimal solution to the dual). Explain why there is a difference in the variables listed in in the tableau compared to the original LP.

Solution: The optimal solution is $x_3 = 5$, $x_1 = x_2 = 0$, and the objective value is 15 .

By strong LP duality, the objective value of the dual is the same as the objective value of the primal.

There is a difference between the variables in the tableau and the original LP because slack variables were introduced.

(c) (10 points) Now introduce the new constraint

$$x_1 + 2x_3 \leq 6,$$

in addition to the existing constraints in the primal LP. Determine an optimal solution to the new LP (along with the corresponding new objective value) bu using the dual simplex method.

(Hint: if you do the calculations correctly, all fractions you see will be multiples of $\frac{1}{3}$, and only one pivot step will be required.)

Solution:

		x_1	x_2	x_3	x_4	s_2	s_3	
Z	1	2	7	0	3	0	0	15
x_3	0	2	2	1	1	0	0	5
s_2	0	-1	1	0	-2	1	0	10
s_3	0	1	0	2	0	0	1	6

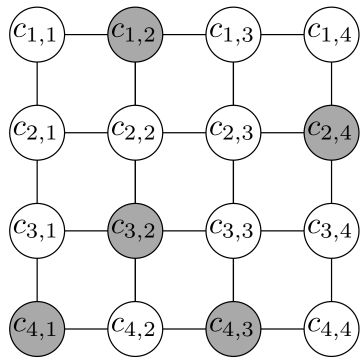
Z	1	2	7	0	3	0	0	15
x_3	0	2	2	1	1	0	0	5
s_2	0	-1	1	0	-2	1	0	10
→ s_3	0	(-3)	-4	0	-2	0	1	-4

Z	1	2	7	0	3	0	0	15
x_3	0	2	2	1	1	0	0	5
s_2	0	-1	1	0	-2	1	0	10
→ s_3	0	1	4/3	0	2/3	0	-1/3	4/3

Z	1	0	13/3	0	2/3	0	2/3	37/3 = 12 1/3.
x_3	0	0	-2/3	1	-1/3	0	2/3	7/3
s_2	0	0	7/3	0	-4/3	1	-1/3	34/3.
x_1	0	1	4/3	0	2/3	0	-1/3	4/3

$Z^* = 12\frac{1}{3}, x_1^* = \frac{4}{3}, x_3^* = \frac{7}{3}, x_2^* = 0$

2. (15 points) You want to build some windmills for generating power out in the North Sea. The possible windmill locations form a grid, as shown. Different locations will incur different costs, because of variations in the sea floor; we denote the cost of site at row i and column j by $c_{i,j}$, as shown.



You must build exactly 5 windmills. However, it is not allowed to place two windmills on two sites that are directly adjacent, either horizontally or vertically, because this will cause wind interference. The grayed circles show one possible solution that is valid. (Notice that you can place two windmills on squares that are diagonal from each other.)

Formulate the problem of building all the required windmills as an integer linear program. Explain the meaning of all your variables and constraints.

Solution: Let $x_{i,j}$ be a binary variable indicating whether we build a windmill at the site at row i and column j.

$$\min \quad \sum_{i=1}^{4} \sum_{j=1}^{4} c_{i,j} x_{i,j}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{i=1}^{4} \sum_{j=1}^{4} x_{i,j} = 5 \\
& x_{i,j} + x_{i,j+1} \leq 1 \qquad i = 1, \ldots, 4, j = 1, \ldots, 3 \\
& x_{i,j} + x_{i+1,j} \leq 1 \qquad i = 1, \ldots, 3, j = 1, \ldots, 4 \\
& x_{i,j} \in \{0, 1\} \qquad i = 1, \ldots, 4, j = 1, \ldots, 4.
\end{aligned}$$

The first constraint says that there are 5 windmills. The second says that in any horizontally adjacent pair of sites, at most 1 windmill is built. The third says the same for vertically adjacent sites.

3. Consider the following problem. You are packing your suitcase for an overseas trip, and need to decide what items to bring. You have 4 items (just one of each); each has a certain value to you, and a certain weight:

weight (kg)	3	5	4	1
value	7	12	10	2

You can only carry at most 10kg of weight. The problem is to decide which items you should put into your knapsack to maximize the overall value.

(a) (5 points) Write down an ILP for this problem.

Solution:

$$\begin{aligned} \max \quad & 7x_1 + 12x_2 + 10x_3 + 2x_4 \\ \text{s.t.} \quad & 3x_1 + 5x_2 + 4x_3 + 1x_4 \leq 10 \\ & x_1, x_2, x_3, x_4 \in \{0, 1\} \end{aligned}$$

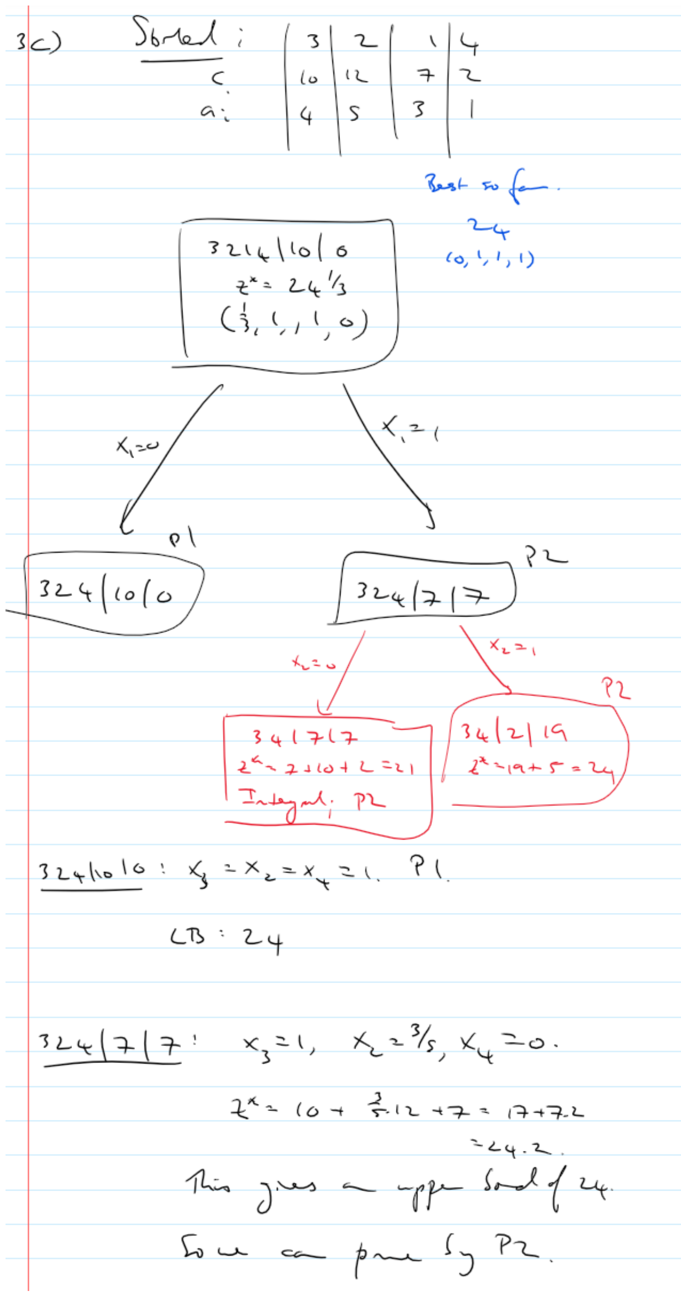
(b) (5 points) Determine the unique optimal solution of the LP relaxation of this problem. Also give the corresponding optimal objective value.

Solution: Order the items by decreasing profit per unit weight: 3,2,1,4.

The greedy algorithm yields then $x_3 = 1$, $x_2 = 1$, $x_1 = \frac{1}{3}$, $x_4 = 0$ with value $22 + 7/3 = 24\frac{1}{3}$.

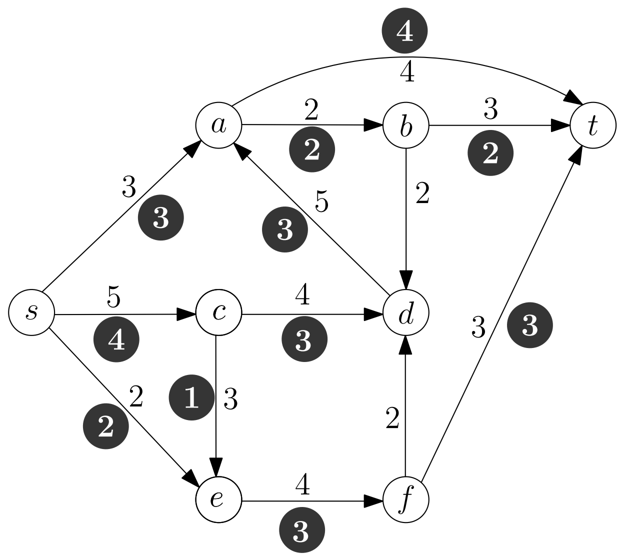
(c) (10 points) Solve the ILP from (a) using Branch and Bound. Clearly indicate in which order you compute the nodes of the search tree and where you prune the search tree, and based on which pruning criterion.

Solution:



So the final solution is $x_1 = 0, x_2 = x_3 = x_4 = 1$, with an objective value of 24. (The above didn't rule out the possibility of another solution of the same objective value, but there cannot be a solution with a strictly larger objective value.

4. Consider the following directed graph; arc capacities are shown. In addition, an s-t-flow f has been given (the circled numbers).



(a) (2 points) Write down the value of the flow f.

> **Solution:** 9, by considering the arcs leaving s, for example.

(b) (8 points) Determine either i) a flow of strictly larger value, or ii) argue convincingly that f is a maximum flow.

> **Solution:** The flow is a maximum flow. To see this, we note that $S = \{s, a, c, d, f\}$ is an s-t-cut of capacity $4 + 2 + 3 = 9$ (the arc from b to d does *not* contribute to the capacity). Since this is the same as the value of the flow f, it follows by the max-flow min-cut theorem that it is maximal.

(c) (10 points) Find a shortest s-t-path in the digraph using Dijkstra's algorithm, assuming that every arc has length 1 (do *not* use the capacities of the arcs as lengths!). Show your working.

Solution:

	a	b	c	d	e	f	t
Iteration 1: $S = \{s\}$	$(1, s)$	$(\infty, \emptyset)$	$(1, s)$	$(\infty, \emptyset)$	$(1, s)$	$(\infty, \emptyset)$	$(\infty, \emptyset)$
Iteration 2: $S = \{s, a\}$	–	$(2, a)$	$(1, s)$	$(\infty, \emptyset)$	$(1, s)$	$(\infty, \emptyset)$	$(2, a)$
Iteration 3: $S = \{s, a, c\}$	–	$(2, a)$	–	$(2, c)$	$(1, s)$	$(\infty, \emptyset)$	$(2, a)$
Iteration 4: $S = \{s, a, c, e\}$	–	$(2, a)$	–	$(2, c)$	–	$(2, e)$	$(2, a)$

So the shortest s-t-path has length 2, and following the parent pointers back from t, we see that t's parent is a, and a's parent is s. So this path is $s - a - t$.

(The above is the shortest solution; if one makes different choices about what to put in S in each iteration, it can be longer.)

5. (a) (4 points) You are considering two algorithms for solving a certain problem. On an input of size n, the worst-case running time of the first algorithm n is $n^2 + n2^n + 100$, and of the second algorithm is $100n^4 + 200n^3 + 3000$.

Which algorithm is faster, assuming a large input size? Explain your answer.

> **Solution:** The exponential function 2^n grows much faster than any polynomial, and $n2^n$ grows only faster. So the second algorithm will be much faster for sufficiently large n.

(b) (6 points) What is the length of a longest increasing subsequence of the following sequence?

$$(1, 3, 2, 6, 4, 10, 7, 9).$$

You *must* solve this problem using dynamic programming – show your working. (You do not need to provide the actual subsequence, but it may be a good idea for checking your answer)

Solution:

i	1	2	3	4	5	6	7	8
d_i	1	2	2	3	3	4	4	5

So the longest increasing subsequence has length $\max_i d_i = 5$ (e.g., 1, 3, 6, 7, 9).

6. (10 points) **This question is challenging; I recommend completing the rest of the exam before attempting it.**

Consider the following variation of the edit distance, which I'll call the modified edit distance. One can add a letter, or modify a letter, as before. But now, one can delete any sequence of consecutive letters, and this counts as just one operation. For example: given the initial word REASON and final word WRONG, one could delete EAS to get RON, then add W to get WRON, and then add G to get WRONG. This is in fact the best solution, so the modified edit distance is 3.

Thinking of this in terms of columns: the configuration is

-	R	E	A	S	O	N	-
W	R	-	-	-	O	N	G
1			1				1

But instead of paying 1 for each column with a change, we pay only 1 for columns 3-5.

Describe a dynamic programming algorithm to determine the modified edit distance, given an initial word $x_1 x_2 \ldots x_m$ that must be changed to $y_1 y_2 \ldots y_n$. You only need to determine the modified edit distance itself, not the way that the word should be transformed.

You should use the same subproblems as with the original edit-distance problem; thus, define

$$E(i, j) = \text{ modified edit distance between } x_1 x_2 \ldots x_i \text{ and } y_1 y_2 \ldots y_j.$$

In fact, you can use the following template. All you need to do is decide what should go in the placeholders [[1]], [[2]] and [[3]] (with explanation).

$E(0, 0) = 0$
$E(i, 0) =$ [[1]] for all $1 \leq i \leq m$
$E(0, j) =$ [[2]] for all $1 \leq j \leq n$
for $i = 1, 2, \ldots, m$ **do**
  for $j = 1, 2, \ldots, n$ **do**
    $E(i, j) =$ [[3]]
return $E(m, n)$

Solution: For placeholder [[1], we must set $E(i, 0) = 1$, since we can delete the entire word at a cost of 1.

For placeholder [[2]], we must set $E(0, j) = j$; we still have to add letters one at a time. (Unlike the normal edit distance, the modified edit distance can change if we swap the role of the two words involved!)

For placeholder [[3]], we have to think pretty hard. Again, consider the final column of the optimal solution corresponding to $E(i, j)$. If both have letters, or the first row is "-" and the second has a letter, nothing changes. But if the first row has a letter and the second a "-", we must allow for the possibility that multiple consecutive elements are deleted. But this isn't a serious problem: if we delete k elements in a row, that means that the final k columns must look like

$\cdots$	y_{i-k+1}	$\cdots$	y_{i-1}	y_i
$\cdots$	-	$\cdots$	-	-

So the cost of the solution is then equal to $E(i-k+1, j)+1$.

So [[3]] should be

$$E(i, j) = \min\big\{E(i-1, j-1) + \text{diff}(x_i, y_j), E(i, j-1) + 1, \min_{k \le j} E(i-k, j) + 1\big\}.$$