Math 340 HW2

Tony Liang
39356493

1) Steel company in LP problem in standard form

Bands 200 tons/h Bands $25/ton Bands make up to 6000 tons
Coils 140 tons/h Coils $30/ton Coils make up to 4000 tons

40 h of production time available

let x_1 = tons of bands produced per hour
x_2 = tons of coils produced per hour

$$\max \quad 25x_1 + 30x_2$$
$$\text{s.t.} \quad x_1 \leq 6000$$
$$x_2 \leq 4000$$
$$\frac{x_1}{200} + \frac{x_2}{140} \leq 40$$
$$x_1, x_2 \geq 0$$

b) Pulp code

See attached code snippet screenshot / ipynb notebook / pdf in canvas submission

2) Write this LP problem Vanderbei 5th edition, Ex 1.2

a) Passengers traveling from Ithaca to Newark seats 30 passengers
b) Passengers " " Newark to Boston in the aircraft
c) " " " Ithaca to Boston

Three farmers dinner

a) Y dinner
b) B dinner
c) C dinner

Ticket prices:

	Ithaca-Newark	Newark-Boston	Ithaca-Boston
Y	300	160	360
B	220	130	280
M	100	80	140

Upper bound of potential customers in each nine possible origin-destination fare class combinations:

Decide how many tickets from each nine origin-destination / fare class combinato to sell:

=

let the following:

x_1 = Y class Ithaca-Newark

x_2 = Y class Newark-Boston

x_3 = Y class Ithaca-Boston

x_4 = B class Ithaca-Newark

x_5 = B class Newark-Boston

x_6 = B class Ithaca-Boston

x_7 = M class Ithaca-Newark

x_8 = M class Newark-Boston

x_9 = M class Ithaca-Boston

each represent # of passengers of class + destination

$$\max\ 300x_1 + 160x_2 + 360x_3 + 220x_4 + 130x_5 + 280x_6 + 100x_7 + 80x_8 + 140x_9$$

s.t. forecasted maximal

$x_1 \le 4 \quad x_4 \le 8 \quad x_7 \le 22$

$x_2 \le 8 \quad x_5 \le 13 \quad x_8 \le 20$

$x_3 \le 3 \quad x_6 \le 10 \quad x_9 \le 18$

Either of two legs cannot be overbooked:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \le 30$$

and $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 \ge 0$

b) Solve LP with python PuLp

see attached screenshots / ipynb / pdf in Canvas submission

hw2-question-2

January 26, 2023

1 HW Question 2, Vanderbei Ex 1.2

Author: Tony Liang

Math 340

39356993

[1]:
```
# imports
import pulp
from pulp import *
```

[2]:
```
# Defined given information
seat_class = ["Y", "B", "M"]
ticket = ["IN", "NB", "IB"]
prices = [300,160,360,220,130,280,100,80,140]
uppers = [4, 8, 3, 8, 13, 10, 10, 22, 20, 18]

# Creates the combination of seat class and destination
comb = []
for s in seat_class:
    for t in ticket:
        n = s + "_" + t
        comb.append(n)
# Creates the costs using comb and ticket price
costs = dict(zip(comb, prices))
demands = dict(zip(comb, uppers))
```

[3]:
```
# Create lp variables for each ticket price, lowerBound 0 and
# being integer variable only
ticket_vars = LpVariable.dicts("ticket", comb,
                               lowBound=0, cat='Integer')
# Creates LP problem
PROBLEM_NAME="ivy_airline"
Lp_prob = LpProblem(PROBLEM_NAME, LpMaximize) # default is Minimize, but course␣
 ↪uses max
Lp_prob += lpSum([costs[i] * ticket_vars[i] for i in comb]), "TicketPrice"
```

[4]:
```
Lp_prob += lpSum([ticket_vars[i] for i in comb]) <= 30, "SeatRequirement"
# Each individual demand upper bound requirement
for (k, v), (k2,v2) in zip(ticket_vars.items(), demands.items()):
    if k == k2:
        Lp_prob += lpSum([ticket_vars[k]]) <= v2, f"{k}_Demand"
```

[5]:
```
# Write problem to .lp file
Lp_prob.writeLP("IvyAirline.lp")
# Shows the problem
print(Lp_prob)
```

```
ivy_airline:
MAXIMIZE
280*ticket_B_IB + 220*ticket_B_IN + 130*ticket_B_NB + 140*ticket_M_IB +
100*ticket_M_IN + 80*ticket_M_NB + 360*ticket_Y_IB + 300*ticket_Y_IN +
160*ticket_Y_NB + 0
SUBJECT TO
SeatRequirement: ticket_B_IB + ticket_B_IN + ticket_B_NB + ticket_M_IB
 + ticket_M_IN + ticket_M_NB + ticket_Y_IB + ticket_Y_IN + ticket_Y_NB <= 30

Y_IN_Demand: ticket_Y_IN <= 4

Y_NB_Demand: ticket_Y_NB <= 8

Y_IB_Demand: ticket_Y_IB <= 3

B_IN_Demand: ticket_B_IN <= 8

B_NB_Demand: ticket_B_NB <= 13

B_IB_Demand: ticket_B_IB <= 10

M_IN_Demand: ticket_M_IN <= 10

M_NB_Demand: ticket_M_NB <= 22

M_IB_Demand: ticket_M_IB <= 20

VARIABLES
0 <= ticket_B_IB Integer
0 <= ticket_B_IN Integer
0 <= ticket_B_NB Integer
0 <= ticket_M_IB Integer
0 <= ticket_M_IN Integer
0 <= ticket_M_NB Integer
0 <= ticket_Y_IB Integer
0 <= ticket_Y_IN Integer
0 <= ticket_Y_NB Integer
```

[6]:
```python
# Solve the lp problem
Lp_prob.solve()
# check lp problem status if equals Optimal,
if not LpStatus[Lp_prob.status] == "Optimal":
    print(f"Optimal Solution was not found, the problem was {LpStatus[Lp_prob.
 ↪status]}")
else:
    print (("Status:"), LpStatus[Lp_prob.status])
    for variable in Lp_prob.variables():
        print(variable.name, "=", variable.varValue)
    print("Optimal value is z = ", value(Lp_prob.objective))
```

```
Status: Optimal
ticket_B_IB = 10.0
ticket_B_IN = 8.0
ticket_B_NB = 0.0
ticket_M_IB = 0.0
ticket_M_IN = 0.0
ticket_M_NB = 0.0
ticket_Y_IB = 3.0
ticket_Y_IN = 4.0
ticket_Y_NB = 5.0
Optimal value is z =  7640.0
```