HW 5 Mingjia Yu. (my263@).

1. (a) Primal: max $Z = x_1 + Bx_2$

s.t $x_1 + 2x_2 \le 4$ ①

$2x_1 - x_2 \le 4$ ② $\Rightarrow$

$-x_1 - x_2 \le -1$ ③

$x_1, x_2 \ge 0$

Dual: min $Z = 4y_1 + 4y_2 - y_3$

s.t $y_1 + 2y_2 - y_3 \ge 1$ ①

$2y_1 - y_2 - y_3 \ge B$ ②

$y_1, y_2, y_3 \ge 0$

(b) if $x_1 = 0$, $x_2 = 2$, due to complementary slackness

Primal constraint: s.t ① $0 + 4 = 4$ → tight → $y_1 \ne 0$

② $0 - 2 = -2 \ne 4$ → Not tight → $y_2 = 0$

③ $-2 \ne -1$ → Not tight → $y_3 = 0$

Dual: ∵ $x_2 = 2$ → s.t $2y_1 - y_2 - y_3 = B$, and $y_2, y_3 = 0$

$2y_1 = B$

$\boxed{y_1 = B/2}$

check feasibility in constraint ① in dual

$y_1 + 2y_2 - y_3 \ge 1$

$B/2 + 0 - 0 \ge 1$

$\boxed{B \ge 2}$

∴ if $B \ge 2$, then $x_1 = 0$ and $x_2 = 2$ is optimal solution

2 a) weight = cost

Previous pt		d(s)
b	a	$\infty \to 5 \to -2$
a	c	$\infty \to 7 \to 0$
c	b	$\infty \to 8 \to 3$
c	t	$\infty \to 10 \to 3$

$t \to c \to a \to b \to c \to a \to \cdots\cdots$

Due to it's a negative cycle, the shortest path is $-\infty$

b) max $s-a$.

s.t $s-a \le 5$

$s-b \le 8$

$b-a \le -10$

$a-c \le 2$

$c-b \le 3$

$c-t \le 3$

$b-t \le 4$

c) Negative cycle constraints: $\begin{cases} b-a \le -10 & ① \\ a-c \le 2 & ② \\ c-b \le 3. & ③ \end{cases}$

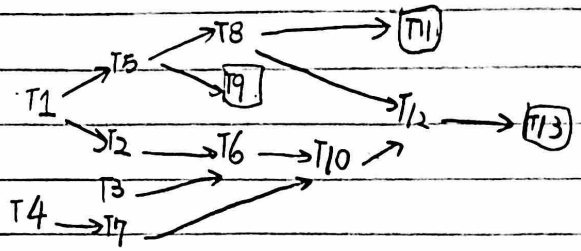
① + ② + ③ : $(b-a)+(a-c)+(c-b) \le -10+2+3$

$0 \le -5$

$\therefore$ dual. Infeasible.

⇒ Verification of dual is included in the next page.

3.



In the diagram, there are 3 terminal activities 9, 11, 13

Primal: $\min -(T_1 + 2T_2 + 3T_3 + 4T_4 + 2T_5 + T_6 + 2T_7 + 6T_8 + 10T_9 + 5T_{10} + 3T_{11} + 3T_{12} + 2T_{13})$

s.t $T_4 + T_3 + T_1 = 1$

$T_7 - T_4 = 0$

$T_6 - T_2 - T_3 = 0$

$T_2 + T_5 - T_1 = 0$

$T_{10} - T_6 - T_7 = 0$

$T_8 + T_9 - T_5 = 0$

$T_{11} + T_{12} - T_{10} - T_8 = 0$

$T_{13} - T_{12} = 0$

$-T_{13} - T_{11} - T_9 = -1$

all nonnegative

Dual: Minimize F, which is the project finish, let X_i be the latest Finish time.

Min F

s.t

$X_1 \geq 1$; $X_9 - X_5 \geq 10$

$X_2 - X_1 \geq 2$; $X_{10} - X_6 \geq 5$

$X_3 \geq 4$; $X_{10} - X_7 \geq 5$

$X_4 \geq 4$; $X_{11} - X_8 \geq 3$

$X_5 - X_1 \geq 2$; $X_{11} - X_{10} \geq 3$

$X_6 - X_2 \geq 1$; $X_{12} - X_8 \geq 3$

$X_6 - X_3 \geq 1$; $X_{12} - X_{10} \geq 3$

$X_7 - X_4 \geq 2$; $F - X_{11} \geq 0$

$X_8 - X_5 \geq 6$; $F - X_9 \geq 0$

All, nonnegative.

b) By using Gurobi, the project duration is **16**

Critical path is **T4 - T7 - T10 - T12 - T13**

Prob 3

November 7, 2019

0.0.1 uni: my2638

0.0.2 name: Mingjia Yu

[ ]:
```
from gurobipy import *


# create a model
m = Model()

# create variables
t1 = m.addVar(vtype=GRB.CONTINUOUS, name="t1", lb=0)
t2 = m.addVar(vtype=GRB.CONTINUOUS, name="t2", lb=0)
t3 = m.addVar(vtype=GRB.CONTINUOUS, name="t3", lb=0)
t4 = m.addVar(vtype=GRB.CONTINUOUS, name="t4", lb=0)
t5 = m.addVar(vtype=GRB.CONTINUOUS, name="t5", lb=0)
t6 = m.addVar(vtype=GRB.CONTINUOUS, name="t6", lb=0)
t7 = m.addVar(vtype=GRB.CONTINUOUS, name="t7", lb=0)
t8 = m.addVar(vtype=GRB.CONTINUOUS, name="t8", lb=0)
t9 = m.addVar(vtype=GRB.CONTINUOUS, name="t9", lb=0)
t10 = m.addVar(vtype=GRB.CONTINUOUS, name="t10", lb=0)
t11 = m.addVar(vtype=GRB.CONTINUOUS, name="t11", lb=0)
t12 = m.addVar(vtype=GRB.CONTINUOUS, name="t12", lb=0)
t13 = m.addVar(vtype=GRB.CONTINUOUS, name="t13", lb=0)


# integrate new variables
m.update()

# set objective
m.setObjective(
    -1*(t1 + 2*t2 + 3*t3 + 4*t4 + 2*t5 + t6 + 2*t7 + 6*t8 + 10*t9 + 5*t10 +␣
 ↪3*t11 + 3*t12 + 2*t13),
    GRB.MINIMIZE
)

# add constraints
```

```
m.addConstr(t4 + t3 + t1 == 1)
m.addConstr(t7 - t4 == 0)
m.addConstr(t6 - t2 - t3 == 0)
m.addConstr(t2 + t5 - t1 == 0)
m.addConstr(t10 - t6 - t7 == 0)
m.addConstr(t8 + t9 - t5 == 0)
m.addConstr(t11 + t12 - t10 - t8 == 0)
m.addConstr(t13 - t12 == 0)
m.addConstr(-1*t13 - t11 - t9 == -1)

# optimize
m.optimize()
print("Model status: ", m.status)

# print out decision variables
for v in m.getVars():
    print(v.varName, v.x, "\n")

print("-"*15)
print("Obj Value: ", m.objVal)
```

[ ]:

[ ]:
```
############################
############################
############################
# Dual

# create a model
m = Model()

# create variables
ys = m.addVar(vtype=GRB.CONTINUOUS, name="ys", lb=-0)
yt = m.addVar(vtype=GRB.CONTINUOUS, name="yt", lb=-0)
ya = m.addVar(vtype=GRB.CONTINUOUS, name="ya", lb=-0)
yb = m.addVar(vtype=GRB.CONTINUOUS, name="yb", lb=-0)
yc = m.addVar(vtype=GRB.CONTINUOUS, name="yc", lb=-0)
yd = m.addVar(vtype=GRB.CONTINUOUS, name="yd", lb=-0)
ye = m.addVar(vtype=GRB.CONTINUOUS, name="ye", lb=-0)
yf = m.addVar(vtype=GRB.CONTINUOUS, name="yf", lb=-0)
yg = m.addVar(vtype=GRB.CONTINUOUS, name="yg", lb=-0)


# integrate new variables
m.update()

# set objective
```

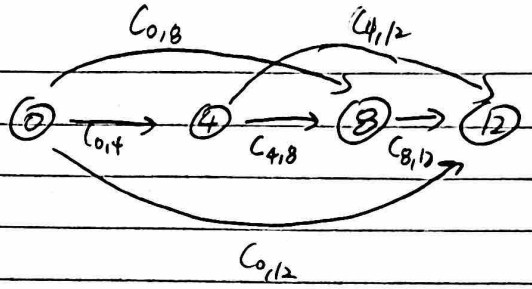
```
m.setObjective(
    yt - ys,
    GRB.MINIMIZE
)

# add constraints
m.addConstr(-1*ys + yc >= 1)
m.addConstr(-1*yc + yb >= 2)
m.addConstr(-1*ys + yb >= 3)
m.addConstr(-1*ys + ya >= 4)
m.addConstr(-1*yc + ye >= 2)
m.addConstr(-1*yb + yd >= 1)
m.addConstr(-1*ya + yd >= 2)
m.addConstr(-1*yc + yf >= 6)
m.addConstr(-1*yc + yt >= 10)
m.addConstr(-1*yd + yf >= 5)
m.addConstr(-1*yf + yt >= 3)
m.addConstr(-1*yf + yg >= 3)
m.addConstr(-1*yg + yt >= 2)


# optimize
m.optimize()
print("Model status: ", m.status)

# print out decision variables
for v in m.getVars():
    print(v.varName, v.x, "\n")

print("-"*15)
print("Obj Value: ", m.objVal)
```

```
Academic license - for non-commercial use only
Optimize a model with 9 rows, 13 columns and 26 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 1e+00]
Presolve removed 6 rows and 6 columns
Presolve time: 0.00s
Presolved: 3 rows, 7 columns, 14 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0   -2.1016000e+01   4.008000e+00   0.000000e+00      0s
       3   -1.6000000e+01   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.00 seconds
Optimal objective -1.600000000e+01
Model status:  2
t1 0.0

t2 0.0

t3 0.0

t4 1.0

t5 0.0

t6 0.0

t7 1.0

t8 0.0

t9 0.0

t10 1.0

t11 0.0

t12 1.0

t13 1.0
```

Dual Result

```
Obj Value:  -16.0
Optimize a model with 13 rows, 9 columns and 26 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 1e+01]
Presolve removed 8 rows and 5 columns
Presolve time: 0.00s
Presolved: 5 rows, 4 columns, 10 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0   -2.0000000e+30   3.000000e+30   2.000000e+00      0s
       3    1.6000000e+01   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.00 seconds
Optimal objective  1.600000000e+01
Model status:  2
ys 0.0

yt 16.0

ya 4.0

yb 5.0

yc 1.0

yd 6.0

ye 3.0

yf 11.0

yg 14.0

---------------
Obj Value:  16.0
```

4.



Network for minimizing cost

Four node are taken w/ 0, 4, 8, 12 and c_{ij} refers to the total cost of shelving all books of height $> i$ and $\leq j$ on a single shelf).

$C_{0,4} = 2300 + 200 \times 5(4 \times 0.5) = \$4300$

$C_{4,8} = 2300 + 100 \times 5(8 \times 0.5) = \$4300$

$C_{8,12} = 2300 + 80 \times 5(4 \times 0.5) = \$4700$

$C_{0,8} = 2300 + 300 \times 5(8 \times 0.5) = \$8300$

$C_{4,12} = 2300 + 180 \times 5(12 \times 0.5) = \$7700$

$C_{0,12} = 2300 + 380 \times 5(12 \times 0.5) = \$13,700$

Shortest path from $0 \rightarrow 12$ is $0 \rightarrow 4 \rightarrow 12$ as shown.

∴ a 4" shelf should be built for 4" books, and a 12" shelf is built for 8" and 12" books.

5 let variable x_j be defined as # of boxes produced of type j.

$$y_j = \begin{cases} 1 & \text{if type j box is produced} \\ 0 & \text{otherwise.} \end{cases}$$

Total cost = Fixed cost + Variable cost.

LP

$$\text{Min } Z = 33x_1 + 30x_2 + 26x_3 + 24x_4 + 19x_5 + 18x_6 + 17x_7 + 1000(y_1+y_2+y_3+y_4+y_5+y_6+y_7)$$

Demand constraint) $x_1 \geq 400$

$x_1 + y_2 \geq 700$

$x_1 + x_2 + x_3 \geq 1200$

$x_1 + x_2 + x_3 + x_4 \geq 1900$

$x_1 + x_2 + x_3 + x_4 + x_5 \geq 2100$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 2500$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \geq 2700$

$x_j \leq M_j y_j \Rightarrow$

$x_1 \leq 400 y_1$

$x_2 \leq 700 y_2$

$x_3 \leq 1200 y_3$

$x_4 \leq 1900 y_4$

$x_5 \leq 2100 y_5$

$x_6 \leq 2500 y_6$

$x_7 \leq 2700 y_7$

We can use Gurobi to solve this problem and the total cost is 72200 by using gurobi. (shown in the following Pdf)

s.t

Prob 5

November 8, 2019

0.0.1 uni: my2638

0.0.2 name: Mingjia Yu

[ ]:
```
from gurobipy import *


# create a model
m = Model()

# create variables
x1 = m.addVar(vtype=GRB.CONTINUOUS, name="x1", lb=0)
x2 = m.addVar(vtype=GRB.CONTINUOUS, name="x2", lb=0)
x3 = m.addVar(vtype=GRB.CONTINUOUS, name="x3", lb=0)
x4 = m.addVar(vtype=GRB.CONTINUOUS, name="x4", lb=0)
x5 = m.addVar(vtype=GRB.CONTINUOUS, name="x5", lb=0)
x6 = m.addVar(vtype=GRB.CONTINUOUS, name="x6", lb=0)
x7 = m.addVar(vtype=GRB.CONTINUOUS, name="x7", lb=0)
y1 = m.addVar(vtype=GRB.INTEGER, name="y1", lb=0, ub=1)
y2 = m.addVar(vtype=GRB.INTEGER, name="y2", lb=0, ub=1)
y3 = m.addVar(vtype=GRB.INTEGER, name="y3", lb=0, ub=1)
y4 = m.addVar(vtype=GRB.INTEGER, name="y4", lb=0, ub=1)
y5 = m.addVar(vtype=GRB.INTEGER, name="y5", lb=0, ub=1)
y6 = m.addVar(vtype=GRB.INTEGER, name="y6", lb=0, ub=1)
y7 = m.addVar(vtype=GRB.INTEGER, name="y7", lb=0, ub=1)

# integrate new variables
m.update()

# set objective
m.setObjective(
    33*x1 + 30*x2 + 26*x3 + 24*x4 + 19*x5 + 18*x6 + 17*x7 + 1000*(y1 + y2 + y3␣
 ↪+ y4 + y5 + y6 + y7),
    GRB.MINIMIZE
)

# add constraints
```

```
m.addConstr(x1 >= 400)
m.addConstr(x1 + x2 >= 700)
m.addConstr(x1 + x2 + x3 >= 1200)
m.addConstr(x1 + x2 + x3 + x4 >= 1900)
m.addConstr(x1 + x2 + x3 + x4 + x5 >= 2100)
m.addConstr(x1 + x2 + x3 + x4 + x5 + x6 >= 2500)
m.addConstr(x1 + x2 + x3 + x4 + x5 + x6 + x7 >= 2700)
m.addConstr(x1 <= 400*y1)
m.addConstr(x2 <= 700*y2)
m.addConstr(x3 <= 1200*y3)
m.addConstr(x4 <= 1900*y4)
m.addConstr(x5 <= 2100*y5)
m.addConstr(x6 <= 2500*y6)
m.addConstr(x7 <= 2700*y7)

# optimize
m.optimize()
print("Model status: ", m.status)

# print out decision variables
for v in m.getVars():
    print(v.varName, v.x, "\n")

print("-"*15)
print("Obj Value: ", m.objVal)
```

```
Explored 1 nodes (14 simplex iterations) in 0.01 seconds
Thread count was 4 (of 4 available processors)

Solution count 3: 72200 72600 73400

Optimal solution found (tolerance 1.00e-04)
Best objective 7.220000000000e+04, best bound 7.220000000000e+04, gap 0.0000%
Model status:  2
x1 400.0

x2 300.0

x3 499.99999999999994

x4 700.0000000000001

x5 799.9999999999999

x6 0.0

x7 0.0

y1 1.0

y2 1.0

y3 1.0

y4 1.0

y5 1.0

y6 0.0

y7 0.0

---------------
Obj Value:  72200.0
```