DSA/ISE 5113 Advanced Analytics and Metaheuristics

Homework #1

Group: 11 (**Tahsin Tabassum, Marepally, Samhitha Reddy, Iffat Borhan)**

Problem - 1

Din Djarin is a lone Mandalorian gunfighter and bounty hunter sometimes referred to as “Mando” who has recently rescued what appears to be baby Yoda (see Figure 1) from resurgent evil Empire forces. Unfortunately, while baby Yoda understands the Mandalorian’s language, he does not speak it. Baby Yoda in fact only says two words, “gurmp” and “pvlork” which Mando knows must mean ‘yes’ or ‘no’, but he does not know which word means which! Furthermore, it is unknown whether or not this child is a truthful being or a sinister lying creature from the dark side...

Mando’s conversation with the child goes as follows:

Mando: “Does gurmp mean yes?”

Child: “Pvlork!”

Can you determine what ‘gurmp’ means? Can you determine if baby Yoda is a liar or truth-teller or an evil dark side liar? Use truth-tables (or a close variation of a truth-table) to demonstrate your case.

Answer:

Given:

1. Baby Yoda uses two words, **"gurmp"** and **"pvlork"**, which represent **yes** or **no**, but their meaning is unknown.
2. Mando asks the child: **“Does gurmp mean, yes?”**

Assumption:

1. Baby Yoda either **tells the truth** (always truthful) or **always lies** (always false).

The goal is to:

- Determine what **gurmp** means (yes or no).
- Determine if Baby Yoda is a **truth-teller** or a **liar**.

Truth Table

Case	Baby Yoda by nature	“Grump” means	Mando’s Question: “Does gurmp mean yes?”	Baby Yoda’s Answer	Implication
1	Truthful	Yes	True	Grump (Yes)	Grump = yes
2	Truthful	No	False	Pvlork (No)	Grump = No
3	Lair	Yes	False	Pvlork (No)	Grump = yes
4	Lair	Yes	True	Grump (Yes)	Grump = No

Analysis of Each Case:

Case 1:

- If Baby Yoda is truthful and **gurmp means yes**, when asked "Does gurmp mean yes?" the answer is **true**, so Baby Yoda responds "gurmp."
- This matches the behavior of a truthful Baby Yoda. Thus, **gurmp = yes**, and Baby Yoda is truthful.

Case 2:

- If Baby Yoda is truthful and **gurmp means no**, when asked "Does gurmp mean yes?" the answer is **false**, so Baby Yoda responds "pvlork."
- This matches the behavior of a truthful Baby Yoda. Thus, **gurmp = no**, and Baby Yoda is truthful.

Case 3:

- If Baby Yoda is a liar and **gurmp means yes**, when asked "Does gurmp mean yes?" the truthful answer would be **true**, but since Baby Yoda lies, they respond **pvlork (no)**.
- This matches the behavior of a lying Baby Yoda. Thus, **gurmp = yes**, and Baby Yoda is a liar.

Case 4:

- If Baby Yoda is a liar and **gurmp means no**, when asked "Does gurmp mean yes?" the truthful answer would be **false**, but since Baby Yoda lies, they respond **gurmp (yes)**.
- This matches the behavior of a lying Baby Yoda. Thus, **gurmp = no**, and Baby Yoda is a liar.

So, it can be concluded that Baby Yoda is lying. However, based on the given scenario, it is impossible to determine the meaning of the term "gump."

Problem - 2

A portfolio manager in charge of a bank portfolio has $10 million to invest. The securities available for purchase, as well as their respective quality ratings, maturities, and yields, are shown in Table 1.2.

Bond name	Bond type	Quality scales Moody's	Bank's	Years to maturity	Yield to maturity	After-tax yield
A	Municipal	Aa	2	9	4.3%	4.3%
B	Agency	Aa	2	15	5.4	2.7
C	Government	Aaa	1	4	5.0	2.5
D	Government	Aaa	1	3	4.4	2.2
E	Municipal	Ba	5	2	4.5	4.5

The bank places the following policy limitations on the portfolio manager's actions:

1. Government and agency bonds must total at least $4 million.

2. The average quality of the portfolio cannot exceed 1.4 on the bank's quality scale. (Note that a low number on this scale means a high-quality bond.)

3. The average years to maturity of the portfolio must not exceed 5 years.

Answer:

The objective is to maximize the after tax earnings.

.mod file

```
reset;
option solver cplex;
#2

#decision variables

var Xa >=0;# Amount to be invested in bond A
var Xb >=0 ;# Amount to be invested in bond B
var Xc >=0 ;# Amount to be invested in bond C
var Xd >=0 ; # Amount to be invested in bond D
var Xe >=0 ; # Amount to be invested in bond E
var Y  >=0 ;# Amount borrowed
#Objective

maximize aftertaxearnings: 0.043*Xa + 0.027*Xb + 0.025*Xc + 0.022*Xd + 0.045*Xe - 0.0275*Y;

#Constraints

subject to  total: Xa + Xb + Xc + Xd + Xe - Y <= 10;
subject to upperbound : Y <= 1;
subject to govtAgencybonds : Xb + Xc + Xd >=4;
subject to averagequality : 0.6*Xa + 0.6*Xb - 0.4*Xc - 0.4*Xd + 3.6*Xe <= 0;
subject to averagematurity : 4*Xa + 10*Xb - Xc - 2*Xd - 3*Xe <= 0;

solve;

display Xa , Xb , Xc , Xd , Xe , Y;
```

Output:

```
ampl: model '/Users/msamhithareddy/Downloads/2.mod';
CPLEX 20.1.0.0: optimal solution; objective 0.3007
5 dual simplex iterations (3 in phase I)
Xa = 2.4
Xb = 0
Xc = 8.1
Xd = 0
Xe = 0.5
Y = 1
```

Therefore, from the output , it is clear that we buy 2.4 million of bond A, 8.1 million of bond C and 0.5 million of bond E.

Problem – 3

(a) You are in charge of an advertising campaign for a new product, with a budget of $1 million. You can advertise on TV or in magazines. One minute of TV time costs $20,000 and reaches 1.8 million potential customers; a magazine page costs $10,000 and reaches 1 million. You must sign up for at least 10 minutes of TV time. How should you spend your budget to maximize your audi ence? Formulate the problem in AMPL and solve it. Check the solution by hand using at least one of the approaches described in Section 1.1.

(b) It takes creative talent to create effective advertising; in your organization, it takes three person-weeks to create a magazine page, and one person-week to create a TV minute. You have only 100 person-weeks available. Add this constraint to the model and determine how you should now spend your budget. (c) Radio advertising reaches a quarter million people per minute, costs $2,000 per minute, and requires only 1 person-day of time. How does this medium affect your solutions?

(d) How does the solution change if you have to sign up for at least two magazine pages? A maxi mum of 120 minutes of radio?

Answer:

3. a.

.mod file

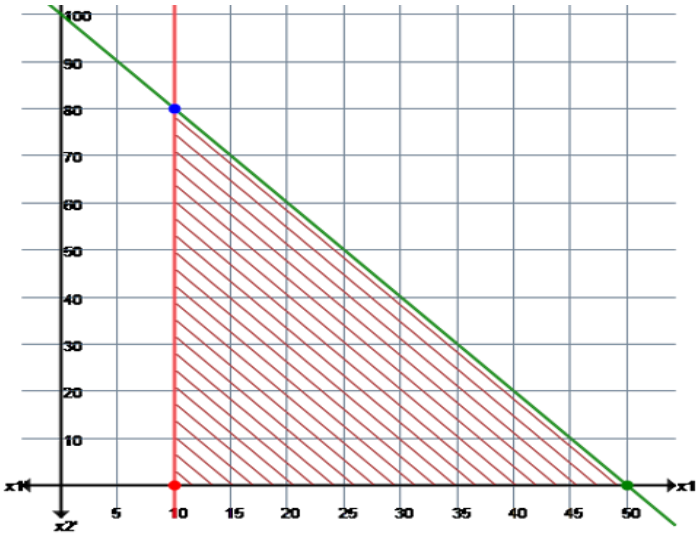
```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

var x1 >= 0; # Budget to spend for advertisement in TV
var x2 >= 0; # Budget to spend for advertisement in magazines

maximize audience: 1.8*x1 + x2; # The objective is to maximize audience

# Must sign up for at least 10 minutes of TV time
subject to cons_TV_time: x1 >= 10;
# One minute of TV time costs $20,000 & a magazine page costs $10,000
subject to cost: 0.02*x1 + 0.01*x2 <= 1;

solve; # Solve the problem

# Print the variables
printf "Budget to spend for advertisement in TV: %d million dollars\n", x1;
printf "Budget to spend for advertisement in magazines: %d million dollars\n", x2;
```

Output

```
ampl: model HW1_group11_p3a.mod
CPLEX 22.1.1:              CPLEX 22.1.1: optimal solution; objective 98
0 simplex iterations
Budget to spend for advertisement in TV: 10 million dollars
Budget to spend for advertisement in magazines: 80 million dollars
ampl:
```

According to the graphical method, Optimal point (10, 80) Value of the objective function

= 1.8*10 + 80 = 98



The results obtained from AMPL are similar to the results obtained from the graphical method.

3.b.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

var x1 >= 0; # Budget to spend for advertisement in TV
var x2 >= 0; # Budget to spend for advertisement in magazines

maximize audience: 1.8*x1 + x2; # The objective is to maximize audience

# Must sign up for at least 10 minutes of TV time
subject to cons_TV_time: x1 >= 10;
# One minute of TV time costs $20,000 & a magazine page costs $10,000
subject to cost: 0.02*x1 + 0.01*x2 <= 1;
# Three person-weeks to create a magazine page, and one person-week to
# create a TV minute
subject to time: 1*x1 + 3*x2 <= 100;


solve; # Solve the problem

# Print the variables
printf "Budget to spend for advertisement in TV: %d million dollars\n", x1;
printf "Budget to spend for advertisement in magazines: %d million dollars\n", x2;
```

Output

```
CPLEX 22.1.1:                CPLEX 22.1.1: optimal solution; objective 92
2 simplex iterations
Budget to spend for advertisement in TV: 40 million dollars
Budget to spend for advertisement in magazines: 20 million dollars
ampl:
```

3.c.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

var x1 >= 0; # Budget to spend for advertisement in TV
var x2 >= 0; # Budget to spend for advertisement in magazines
var x3 >= 0; # Budget to spend for advertisement in radio

maximize audience: 1.8*x1 + x2 + 0.25*x3; # The objective is to maximize audience

# Must sign up for at least 10 minutes of TV time
subject to cons_TV_time: x1 >= 10;
# One minute of TV time costs $20,000 & a magazine page costs $10,000
subject to cons_cost: 0.02*x1 + 0.01*x2 + 0.002*x3 <= 1;
# Three person-weeks to create a magazine page, and one person-week to
# create a TV minute
subject to cons_time: 1*x1 + 3*x2 + (1/7)*x3 <= 100;


solve; # Solve the problem

# Print the variables
printf "Budget to spend for advertisement in TV: %d million dollars\n", x1;
printf "Budget to spend for advertisement in magazines: %d million dollars\n", x2;
printf "Budget to spend for advertisement in radio: %d million dollars\n", x3;
```

Output

```
CPLEX 22.1.1:                CPLEX 22.1.1: optimal solution; objective 118
1 simplex iterations
Budget to spend for advertisement in TV: 10 million dollars
Budget to spend for advertisement in magazines: 0 million dollars
Budget to spend for advertisement in radio: 400 million dollars
ampl:
```

3.d.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

var x1 >= 0; # Budget to spend for advertisement in TV
var x2 >= 0; # Budget to spend for advertisement in magazines
var x3 >= 0; # Budget to spend for advertisement in radio

maximize audience: 1.8*x1 + x2 + 0.25*x3; # The objective is to maximize audience

# Must sign up for at least 10 minutes of TV time
subject to cons_TV_time: x1 >= 10;
# One minute of TV time costs $20,000 & a magazine page costs $10,000
subject to cons_cost: 0.02*x1 + 0.01*x2 + 0.002*x3 <= 1;
# Three person-weeks to create a magazine page, and one person-week to
# create a TV minute
subject to cons_time: 1*x1 + 3*x2 + (1/7)*x3 <= 100;
# Have to sign up for at least two magazine pages
subject to C4: x2 >= 2;
# Have to sign up for a maximum of 120 minutes of radio
subject to C5: x3 <= 120;


solve; # Solve the problem

# Print the variables
printf "Budget to spend for advertisement in TV: %d million dollars\n", x1;
printf "Budget to spend for advertisement in magazines: %d million dollars\n", x2;
printf "Budget to spend for advertisement in radio: %d million dollars\n", x3;
```

Output

```
ampl: model HW1_group11_p3d.mod
CPLEX 22.1.1:                CPLEX 22.1.1: optimal solution; objective 100.1942857
3 simplex iterations
Budget to spend for advertisement in TV: 29 million dollars
Budget to spend for advertisement in magazines: 17 million dollars
Budget to spend for advertisement in radio: 120 million dollars
```

Problem – 4

The steel model of this chapter can be further modified to reflect various changes in produc tion requirements. For each part below, explain the modifications to Figures 1-6a and 1-6b that would be required to achieve the desired changes. (Make each change separately, rather than accu mulating the changes from one part to the next.)

(a) How would you change the constraints so that total hours used by all products must equal the total hours available for each stage? Solve the linear program with this change, and verify that you get the same results. Explain why, in this case, there is no difference in the solution.

(b) How would you add to the model to restrict the total weight of all products to be less than a new parameter, max_weight? Solve the linear program for a weight limit of 6500 tons, and explain how this extra restriction changes the results.

(c) The incentive system for mill managers may tend to encourage them to produce as many tons as possible. How would you change the objective function to maximize total tons? For the data of our

example, does this make a difference to the optimal solution? (d) Suppose that instead of the lower bounds represented by commit[p] in our model, we want to require that each product represent a certain share of the total tons produced. In the algebraic nota tion of Figure 1-1, this new constraint might be represented as

$X_j \geq s_j \Sigma X_k$, for each $j \in P$

where sj is the minimum share associated with project j. How would you change the AMPL model to use this constraint in place of the lower bounds commit[p]? If the minimum shares are 0.4 for bands and plate, and 0.1 for coils, what is the solution?

Verify that if you change the minimum shares to 0.5 for bands and plate, and 0.1 for coils, the lin ear program gives an optimal solution that produces nothing, at zero profit. Explain why this makes sense.

(e) Suppose there is an additional finishing stage for plates only, with a capacity of 20 hours and a rate of 150 tons per hour. Explain how you could modify the data, without changing the model, to incorporate this new stage.

Answer:

4.a.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

set PROD; # set of products
set STAGE; # set of stages

param rate {PROD,STAGE} > 0; # tons per hour in each stage
param avail {STAGE} >= 0; # hours available/week in each stage
param profit {PROD}; # profit per ton
param commit {PROD} >= 0; # lower limit on tons sold in week
param market {PROD} >= 0; # upper limit on tons sold in week

var Make {p in PROD} >= commit[p], <= market[p]; # tons produced

maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];
# Objective: total profits from all products

subject to Time {s in STAGE}:
sum {p in PROD} (1/rate[p,s]) * Make[p] = avail[s];
# In each stage: total of hours used by all
# products may not exceed hours available

data HW1_group11_p4a_data.dat;
solve;

for {p in PROD} {
    printf "Value of %s: %g\n", p, Make[p];
}
```

.dat file

```
data;

set PROD := bands coils plate;
set STAGE := reheat roll;
param rate: reheat roll :=
bands 200 200
coils 200 140
plate 200 160 ;
param: profit commit market :=
bands 25 1000 6000
coils 30 500 4000
plate 29 750 3500 ;
param avail := reheat 35 roll 40 ;
```

Output

```
ampl: model HW1_group11_p4a.mod
CPLEX 22.1.1:               CPLEX 22.1.1: optimal solution; objective 190071.4286
2 simplex iterations
Value of bands: 3357.14
Value of coils: 500
Value of plate: 3142.86
```

The "Time" constraints have been modified from "≤" (less than or equal to) to "=" (equal to), ensuring that the total hours used by all products exactly match the total hours available at each production stage.

Despite the changes in the time constraints from inequality to equality, the results remained unchanged. This indicates that the left-hand side (LHS) of the constraint fully utilizes the available resources on the right-hand side (RHS). Therefore, when the model seeks the optimal solution, both cases yield identical results, suggesting that the solution lies on the boundary of the constraints.

4.b.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

set PROD; # set of products
set STAGE; # set of stages

param rate {PROD,STAGE} > 0; # tons per hour in each stage
param avail {STAGE} >= 0; # hours available/week in each stage
param profit {PROD}; # profit per ton
param commit {PROD} >= 0; # lower limit on tons sold in week
param market {PROD} >= 0; # upper limit on tons sold in week
param max_weight >= 0;

var Make {p in PROD} >= commit[p], <= market[p]; # tons produced
maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];
# Objective: total profits from all products
subject to Time {s in STAGE}:
sum {p in PROD} (1/rate[p,s]) * Make[p] <= avail[s];
# In each stage: total of hours used by all
# products may not exceed hours available
subject to WEIGHT :
sum {p in PROD} Make[p] <= max_weight;
# Restrict the total weight of all products to be less than
# a weight limit of 6500 tons

data HW1_group11_p4b_data.dat;
solve;

for {p in PROD} {
    printf "Value of %s: %g\n", p, Make[p];
}
```

.dat file

```
data;

set PROD := bands coils plate;
set STAGE := reheat roll;
param rate: reheat roll :=
bands 200 200
coils 200 140
plate 200 160 ;
param: profit commit market:=
bands 25 1000 6000
coils 30 500 4000
plate 29 750 3500 ;
param avail := reheat 35 roll 40 ;
param max_weight := 6500;
```

Output

```
ampl: model HW1_group11_p4b.mod
CPLEX 22.1.1:                CPLEX 22.1.1: optimal solution; objective 183791.6667
3 simplex iterations
Value of bands: 1541.67
Value of coils: 1458.33
Value of plate: 3500
```

The introduction of the extra weight restriction significantly alters the solution. The production of bands decreases substantially, while the production of coils increases dramatically, and the production of plates experiences a slight increase.

4.c.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

set PROD; # set of products
set STAGE; # set of stages

param rate {PROD,STAGE} > 0; # tons per hour in each stage
param avail {STAGE} >= 0; # hours available/week in each stage
param profit {PROD}; # profit per ton
param commit {PROD} >= 0; # lower limit on tons sold in week
param market {PROD} >= 0; # upper limit on tons sold in week

var Make {p in PROD} >= commit[p], <= market[p]; # tons produced

maximize Total_Production: sum {p in PROD} Make[p];
# Objective: total production of all products

subject to Time {s in STAGE}:
sum {p in PROD} (1/rate[p,s]) * Make[p] <= avail[s];
# In each stage: total of hours used by all
# products may not exceed hours available

data HW1_group11_p4c_data.dat;
solve;

for {p in PROD} {
    printf "Value of %s: %g\n", p, Make[p];
}
```

.dat file

```
data;

set PROD := bands coils plate;
set STAGE := reheat roll;
param rate: reheat roll :=
bands 200 200
coils 200 140
plate 200 160 ;
param: profit commit market :=
bands 25 1000 6000
coils 30 500 4000
plate 29 750 3500 ;
param avail := reheat 35 roll 40 ;
```

Output

```
ampl: model HW1_group11_p4c.mod
CPLEX 22.1.1:                CPLEX 22.1.1: optimal solution; objective 7000
1 simplex iterations
Value of bands: 5750
Value of coils: 500
Value of plate: 750
ampl:
```

The updated objective function, which focuses on maximizing production instead of profit, alters the optimal solution for bands and plates. The production of bands increases, while the production of plates decreases. However, the production of coils remains unchanged.

4.d.1

Case 1: the minimum shares are 0.4 for bands and plate, and 0.1 for coils.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

set PROD; # set of products
set STAGE; # set of stages

param rate {PROD,STAGE} > 0; # tons per hour in each stage
param avail {STAGE} >= 0; # hours available/week in each stage
param profit {PROD}; # profit per ton
param commit {PROD} >= 0; # lower limit on tons sold in week
param market {PROD} >= 0; # upper limit on tons sold in week
param share {PROD} >= 0;

var Make {p in PROD} <= market[p]; # tons produced
maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];
# Objective: total profits from all products
subject to Time {s in STAGE}:
sum {p in PROD} (1/rate[p,s]) * Make[p] <= avail[s];
# In each stage: total of hours used by all
# products may not exceed hours available
subject to Amount {p in PROD}:
Make[p] >= share[p] * sum {k in PROD} Make[k];
# each product represent a certain share of the total tons produced

data HW1_group11_p4d1_data.dat;
solve;

for {p in PROD} {
    printf "Value of %s: %g\n", p, Make[p];
}
```

.dat file

```
data;

set PROD := bands coils plate;
set STAGE := reheat roll;
param rate: reheat roll :=
bands 200 200
coils 200 140
plate 200 160 ;
param: profit commit market share :=
bands 25 1000 6000 0.4
coils 30 500 4000 0.1
plate 29 750 3500 0.4 ;
param avail := reheat 35 roll 40 ;
```

Output

```
ampl: model HW1_group11_p4d1.mod
CPLEX 22.1.1:               CPLEX 22.1.1: optimal solution; objective 189700
5 simplex iterations
Value of bands: 3500
Value of coils: 700
Value of plate: 2800
```

4.d.2

Case 2: the minimum shares are 0.5 for bands and plate, and 0.1 for coils.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

set PROD; # set of products
set STAGE; # set of stages

param rate {PROD,STAGE} > 0; # tons per hour in each stage
param avail {STAGE} >= 0; # hours available/week in each stage
param profit {PROD}; # profit per ton
param commit {PROD} >= 0; # lower limit on tons sold in week
param market {PROD} >= 0; # upper limit on tons sold in week
param share {PROD} >= 0;

var Make {p in PROD} <= market[p]; # tons produced
maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];
# Objective: total profits from all products
subject to Time {s in STAGE}:
sum {p in PROD} (1/rate[p,s]) * Make[p] <= avail[s];
# In each stage: total of hours used by all
# products may not exceed hours available
subject to Amount {p in PROD}:
Make[p] >= share[p] * sum {k in PROD} Make[k];
# each product represent a certain share of the total tons produced

data HW1_group11_p4d2_data.dat;
solve;

for {p in PROD} {
    printf "Value of %s: %g\n", p, Make[p];
}
```

.dat file

```
data;

set PROD := bands coils plate;
set STAGE := reheat roll;
param rate: reheat roll :=
bands 200 200
coils 200 140
plate 200 160 ;
param: profit commit market share :=
bands 25 1000 6000 0.5
coils 30 500 4000 0.1
plate 29 750 3500 0.5 ;
param avail := reheat 35 roll 40 ;
```

Output

```
ampl: model HW1_group11_p4d2.mod
CPLEX 22.1.1:               CPLEX 22.1.1: optimal solution; objective 0
4 simplex iterations
Value of bands: 0
Value of coils: 0
Value of plate: 0
```

The question states that the shares of bands and plates must each be at least 50% of the total production, while the share of coils must be 10%. This is logically impossible because the combined production shares of these three items would exceed 100%, reaching 110%. Therefore, the results are consistent and make sense.

4.e.

.mod file

```
reset; # Reset the console
option solver cplex; # set the default solver to CPLEX

set PROD; # set of products
set STAGE; # set of stages

param rate {PROD,STAGE} > 0; # tons per hour in each stage
param avail {STAGE} >= 0; # hours available/week in each stage
param profit {PROD}; # profit per ton
param commit {PROD} >= 0; # lower limit on tons sold in week
param market {PROD} >= 0; # upper limit on tons sold in week

var Make {p in PROD} >= commit[p], <= market[p]; # tons produced
maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];
# Objective: total profits from all products
subject to Time {s in STAGE}:
sum {p in PROD} (1/rate[p,s]) * Make[p] <= avail[s];
# In each stage: total of hours used by all
# products may not exceed hours available

data HW1_group11_p4e_data.dat;
solve;

for {p in PROD} {
    printf "Value of %s: %g\n", p, Make[p];
}
```

.dat file

```
data;

set PROD := bands coils plate;
set STAGE := reheat roll finishing;
param rate: reheat roll finishing :=
bands 200 200 1e20
coils 200 140 1e20
plate 200 160 150;
param: profit commit market :=
bands 25 1000 6000
coils 30 500 4000
plate 29 750 3500 ;
param avail := reheat 35 roll 40 finishing 20;
```

Output

```
CPLEX 22.1.1:                CPLEX 22.1.1: optimal solution; objective 189916.6667
3 simplex iterations
Value of bands: 3416.67
Value of coils: 583.333
Value of plate: 3000
```

Problem- 5

Reformulate the problem in Question 2, restricting the bonds available only to bonds A and D. Further add a constraint that the holdings of municipal bonds must be less than or equal to $3 million.

a. What is the optimal solution?
b. What is the shadow price on the municipal limit?
c. How much can the municipal limit be relaxed before it becomes a nonbinding constraint?
d. Below what interest rate is it favorable to borrow funds to increase the overall size of the portfolio?
e. Why is this rate less than the earnings rate on the portfolio as a whole?

5.a.

.mod file

```
reset;
option solver cplex;

#5

# desicion variables
var Xa >=0; # Amount to be invested in bond A
var Xd >=0; # Amount to be invested in bond D
var y >=0; #Amount to be borrowed
#Objective

maximize aftertaxearnings: 0.043*Xa + 0.022*Xd - 0.0275*y;

#Constraints
subject to muncipallimit: Xa <=3;
subject to borrowing : y <= 1;
subject to  total: Xa + Xd <= 10 + y;
subject to govtAgencybonds : Xd >=4;
subject to averagequality : 0.6*Xa - 0.4*Xd <= 0;
subject to averagematurity : 4*Xa - 2*Xd <= 0;
#a)
 solve;

 display Xa, Xd, y;

#b)
printf"Shadow price of muncipallimit constraint :\n ";
display muncipallimit;


#d)
display y.rc; # deduct this reduced cost from y coefficient to get the favorable  interest rate
```

Output

```
ampl: model '/Users/msamhithareddy/Downloads/5a,b,d.mod';
CPLEX 20.1.0.0: optimal solution; objective 0.283
0 dual simplex iterations (0 in phase I)
Xa = 3
Xd = 7
y = 0

Shadow price of muncipallimit constraint :
 muncipallimit = 0.021

y.rc = -0.0055
```

From the output we can see that the optimal solution is 0.283. And we buy 3 million of bond A and 7 million of bond D. y=0 means we borrow zero dollars.

5.b.

Shadow price is the additional after-tax earnings that can be achieved when we increase the municipal limit. In this output we can see that the shadow price of municipal limit is 0.021.

5.c.

.mod file

```
reset;
option solver cplex;

#5c

# desicion variables
var Xa >=0; # Amount to be invested in bond A
var Xd >=0; # Amount to be invested in bond D
var y >=0; #Amount to be borrowed
#Objective

maximize aftertaxearnings: 0.043*Xa + 0.022*Xd - 0.0275*y;

#Constraints

subject to  total: Xa + Xd <= 10+y;
subject to borrowing : y <= 1;
subject to govtAgencybonds : Xd >=4;
subject to averageequality : 0.6*Xa - 0.4*Xd <= 0;
subject to averagematurity : 4*Xa - 2*Xd <= 0;

 solve;

 display Xa, Xd , y ;
```

Output

```
ampl: model '/Users/msamhithareddy/Downloads/5c.mod';
CPLEX 20.1.0.0: optimal solution; objective 0.2915
3 dual simplex iterations (3 in phase I)
Xa = 3.66667
Xd = 7.33333
y = 1
```

When we remove the municipal limit constraint , the amount that we invest in bond A will increase by $0.66667 which becomes 3.66667 which implies that we cannot relax the municipal limit by more than 0.66667 otherwise it becomes a non-binding constraint.

5.d.

Now the amount we borrow is at zero but for it to not be zero we need to find a minimum coefficient for y. And this can be done by finding the reduced cost i.e. y.rc. From the output we can conclude that the y coefficient should be reduced by 0.0055 for y to not be zero. Therefore 2.2%(0.022) is the interest rate that is favorable to borrow funds.

5.e.

It is due to the restriction on buying Bond A. As we see in the output, we are buying all the possible 3 million Bond A. The 2.2% represents the equivalent of the after-tax yield of Bond D, which would be the bond we would buy, if we have money available at the right interest rate. When we remove the municipal limit constraint, we can see that the amount we invest in A increases. This is the reason why the rate is less than the earnings rate .