Assigment 2 - Alloy Corporation

Pranav Kasela - 846965

Introduction

The **Manifacturing INC** (MNC) makes four different metal alloys: A, B, C and D. The company is barely able to keep up with the increasing demand for these products. However, increasing raw material prices and foreign competition are forcing MNC to watch its margins to ensure it is operating in the most efficient manner possible.

Marketing Demands

To meet marketing demands for the coming week, MNC needs to produce

- at least 1,200 pounds of the A product,
- between 450 and 650 pounds of the B product,
- no more than 120 pounds of the C product, and
- no more than 150 pounds of D product.

Production Requirements

Each pound of the A, B, C, and D product contains, respectively, 55%, 45%, 25%, and 100% aluminum made up of copper.

The company has 1300 pounds of aluminum and 900 pounds of copper copper for use in the next week.

The various products are made using four different machines: forger, reamer, shaper, and planer. The following table summarizes the time required by each product on each machine. Each machine has 60 hours of time available in the coming week.

Financial summary

The controller recently presented management with the following financial summary of MNC's average weekly operations over the past quarter. From this report, the controller is arguing that the company should cease producing its B and C products.

	Minutes required per pound			
Machine	**A**	**B**	**C**	**D**
Forger	1.20	1.00	1.00	1.5
Reamer	1.75	1.00	1.00	1.50
Shaper	1.30	0.60	0.40	0.00
Planer	2.50	1.20	2.00	1.50

Figure 1: Districts

	Product				
	A	B	C	D	Total
Sales Revenue	5769	2000	700	879	9348
Variables costs	3200	1550	300	350	5050
Fixed costs	600	500	250	300	1650
Net profit	1969	-50	150	229	2648
Pounds sold	1200	400	100	250	1950
Net profit per pound	1.64	-0.12	1.50	0.91	1.35

Figure 2: Financial summary

Questions

a. Do you agree with the controller's recommendation? Why or why not?
b. Formulate an LP model for this problem.
c. Create a executable model in R for this problem and solve it.
d. What is the optimal solution?
e. Perform the sensitivity analysis for this solution and answer the following questions.
f. Is the solution degenerate?
g. Is the solution unique?
h. If MNC wanted to decrease the production on any product, which one would you recommend and why?
i. If MNC wanted to increase the production of any product, which one would you recommend and why?
j. Which resources are preventing MNS from making more money? If they could acquire more of this resource how much should they acquire & how much should they be willing to pay to acquire it?
k. How much should MNC be willing to pay to acquire more copper?
l. If the marketing department wanted to decrease the price of the A product by $0.25, would the optimal solution change?

Solutions

a. Do you agree with the controller's recommendation? Why or why not?

We don't agree with the controller's recommendation, using the current financial summary of Products, the *Product B* has negative net profit (-0.12 per Pound) while the *Product C* has positive net profit (1.5 per Pound). If we used the net profit as a performance measure of the products, which implies that the demand of the products will not change in the future, the controller would right on the cessation of the production of the *Product B* but not of the *Product C*.
But as we can see that the demand of the *Product B* changed, using the current minimum demand for the product B, the net profit is atleast $minimun\ quantity \cdot Margin\ profit\ per\ Pound$(From the table below)$-Fixed\ Costs = 1.125 * 450 - 500 = 6.25 > 0$ which is positive thus profitable, but it is a small profit. If we could allocate the resources spent for the Product B in other more profitable Products, such as Product C, we will have a better profit since the marginal profit for the Product B is the lowest one (1.13) but we need also to check the reduced cost for the Product B in our problem, it may consume less resources to produce, so for example with the same Resources with we produce a single unit of Product_X, we could produce more units of Product B, thus making it more profitable than the other Product_X.
The interruption in the production of the *Product B* is still arguable and depends a lot also on the market request, but in the case of the *Product C* we see that the marginal profit per Pound is 4.00, and it has a fixed cost of only 250, lowest one in all four products, thus we need to produce only 62.5 pounds of Product C to have a profit, as long as this demand is satisfied (in the last quarter it was satified) there is no need to interrupt the production of the Product C.

b. Formulate an LP model for this problem.

We create two different models one using the binary variable (called bin_model) and the other one without using the binary variables (called model)

Our objective is to maximize the profit, for that we define the *Margin Profit* as the *Sales Revenue-Variable Costs* and the Margin Profit per Pound as *Margin Profit/Pounds Sold*.

Product	A	B	C	D
Sales Revenue	5769	2000	700	879
Variable Costs	3200	1550	300	350
Margin Profit	2569	450	400	529
Pounds Sold	1200	400	100	250
Margin Profit per Pound	2.14	1.13	4.00	2.12

We start by the one **without** the binary variables:

In this case we ignore the fixed costs because if we included them, since they are constant, they will be overlooked by the problem, so will not change the problem itself.

We indicate with A the pound sold of Product A (same for B, C and D) , the **objective function**, we want to **maximize**, is:

$$2.14 \cdot A + 1.13 \cdot B + 4 \cdot C + 2.12 \cdot D$$

The domain of the variables are:

$$\begin{aligned} 1200 \leq A &< +\infty \\ 450 \leq B &\leq 650 \\ 0 \leq C &\leq 120 \\ 0 \leq D &\leq 150 \end{aligned}$$

Based on the composition of the material in each Product and the limited resources we also have that:

$$\begin{aligned} &0.55 \cdot A + 0.45 \cdot B + 0.25 \cdot C + 1.00 \cdot D \leq 1300 && \rightarrow \text{Maximum Alminium} \\ &0.45 \cdot A + 0.55 \cdot B + 0.75 \cdot C + 0.00 \cdot D \leq 900 && \rightarrow \text{Maximum Copper} \end{aligned}$$

We also have a time limit on each Machine, so the following constraint is also valid:

$$\begin{aligned} &1.20 \cdot A + 1.00 \cdot B + 1.00 \cdot C + 1.50 \cdot D \leq 3600 && \rightarrow \text{Forger} \\ &1.75 \cdot A + 1.00 \cdot B + 1.00 \cdot C + 1.50 \cdot D \leq 3600 && \rightarrow \text{Reamer} \\ &1.30 \cdot A + 0.60 \cdot B + 0.40 \cdot C + 0.00 \cdot D \leq 3600 && \rightarrow \text{Shaper} \\ &2.50 \cdot A + 1.20 \cdot B + 2.00 \cdot C + 1.50 \cdot D \leq 3600 && \rightarrow \text{Planer} \end{aligned}$$

The model **with** the binary variables:

In this case, the fixed cost is activated only in the case the specific Product is produced, and to do so we introduce binary variable Y_i for $i = A, B, C, D$ defined as:

$$\begin{aligned} Y_i &= 1 && \text{if } i > 0 \\ Y_i &= 0 && \text{if } i = 0 \\ \text{where } i &= A, B, C, D \end{aligned}$$

The **objective function** to **maximize** in this case is:

$$2.14 \cdot A + 1.13 \cdot B + 4 \cdot C + 2.12 \cdot D - 600 \cdot Y_A - 500 \cdot Y_B - 250 \cdot Y_C - 300 \cdot Y_D$$

The domain of the variables are:

$$\begin{gathered} 1200 \leq A < +\infty \\ 450 \leq B \leq 650 \\ 0 \leq C \leq 120 \\ 0 \leq D \leq 150 \\ Y_i \in \{0, 1\} \text{ for } i = A, B, C, D \end{gathered}$$

The following constraints are the same as the problem without the binary problem:

$$\begin{aligned} 0.55 \cdot A + 0.45 \cdot B + 0.25 \cdot C + 1.00 \cdot D &\leq 1300 && \rightarrow \text{Maximum Alminium} \\ 0.45 \cdot A + 0.55 \cdot B + 0.75 \cdot C + 0.00 \cdot D &\leq 900 && \rightarrow \text{Maximum Copper} \\ 1.20 \cdot A + 1.00 \cdot B + 1.00 \cdot C + 1.50 \cdot D &\leq 3600 && \rightarrow \text{Forger} \\ 1.75 \cdot A + 1.00 \cdot B + 1.00 \cdot C + 1.50 \cdot D &\leq 3600 && \rightarrow \text{Reamer} \\ 1.30 \cdot A + 0.60 \cdot B + 0.40 \cdot C + 0.00 \cdot D &\leq 3600 && \rightarrow \text{Shaper} \\ 2.50 \cdot A + 1.20 \cdot B + 2.00 \cdot C + 1.50 \cdot D &\leq 3600 && \rightarrow \text{Planer} \end{aligned}$$

We have the linking constraints:

$$\begin{aligned} A &\leq M_A Y_A \text{ with } M_A = \min(1300/0.55, 900/0.45, 3600/1.2, 3600/1.75, 3600/1.3, 3600/2.5) = 1440 \\ B &\leq M_B Y_B \text{ with } M_B = \min(1300/0.45, 900/0.55, 3600/1, 3600/1, 3600/0.6, 3600/1.2) = 1636.36 \\ C &\leq M_C Y_C \text{ with } M_C = \min(1300/0.25, 900/0.75, 3600/1, 3600/1, 3600/0.4, 3600/2) = 1200 \\ D &\leq M_D Y_D \text{ with } M_D = \min(1300/1, 3600/1.5, 3600/1.5, 3600/1.5) = 1300 \end{aligned}$$

c. Create a executable model in R for this problem and solve it.

Here is the implementation of problem defined above in point **b.** of model in R:

```r
if(require(lpSolveAPI)==FALSE) install.packages("lpSolveAPI")
model = make.lp(0,4)                                 # 4 Variables (A,B,C,D)
lp.control(model, sense = "max")                     #Maximazing Problem
set.objfn(model,obj = c(2.141,1.125,4,2.116))    #Objective function coefficients

row.add.mode(model,"on")

add.constraint(model,
               xt = c(0.55,0.45,0.25,1),
               type = "<=",
               rhs = 1300,
               indices = c(1:4)) #Aluminium Constraint

add.constraint(model,
               xt = c(0.45,0.55,0.75,0),
               type = "<=",
               rhs = 900,
               indices = c(1:4)) #Copper Constraint

add.constraint(model,
               xt = c(1.2,1,1,1.5),
               type = "<=",
               rhs = 3600,
               indices = c(1:4)) #Forger Time Constraint

add.constraint(model,
               xt = c(1.75,1,1,1.5),
               type = "<=",
               rhs = 3600,
               indices = c(1:4)) #Reamer Time Constraint

add.constraint(model,
               xt = c(1.3,0.6,0.4,0),
               type = "<=",
               rhs = 3600,
               indices = c(1:4)) #Shaper Time Constraint

add.constraint(model,
               xt = c(2.5,1.2,2,1.5),
               type = "<=",
               rhs = 3600,
               indices = c(1:4)) #Planer Time Constraint

#Setting lower and upper bounds
set.bounds(model,lower = c(1200,450,0,0), upper = c(Inf,650,120,150))

row.add.mode(model,"off")

#Giving names to the contraints and variables
dimnames(model)<- list(c("Aluminium","Copper","Forger","Reamer","Shaper","Planer"),
                       c("A","B","C","D"))
name.lp(model, "Alloy Corporation")
```

```r
print(model) #To see to model
```

```
## Model name: Alloy Corporation
##                A      B      C      D
## Maximize   2.141  1.125      4  2.116
## Aluminium   0.55   0.45   0.25      1  <=  1300
## Copper      0.45   0.55   0.75      0  <=   900
## Forger       1.2      1      1    1.5  <=  3600
## Reamer      1.75      1      1    1.5  <=  3600
## Shaper       1.3    0.6    0.4      0  <=  3600
## Planer       2.5    1.2      2    1.5  <=  3600
## Kind         Std    Std    Std    Std
## Type        Real   Real   Real   Real
## Upper        Inf    650    120    150
## Lower       1200    450      0      0
```

Here is the implementation of problem defined above in point **b.** of bin_model in R:

```r
bin_model = make.lp(0,8)                    # 8 Variables
set.type(bin_model, c(5:8), "binary")
lp.control(bin_model, sense = "max")        #Maximazing Problem
set.objfn(bin_model,obj = c(2.141,1.125,4,2.116,-600,-500,-250,-300))   #Objective function coefficient


row.add.mode(bin_model,"on")

add.constraint(bin_model,
               xt = c(0.55,0.45,0.25,1),
               type = "<=",
               rhs = 1300,
               indices = c(1:4)) #Aluminium Constraint

add.constraint(bin_model,
               xt = c(0.45,0.55,0.75,0),
               type = "<=",
               rhs = 900,
               indices = c(1:4)) #Copper Constraint

add.constraint(bin_model,
               xt = c(1.2,1,1,1.5),
               type = "<=",
               rhs = 3600,
               indices = c(1:4)) #Forger Time Constraint

add.constraint(bin_model,
               xt = c(1.75,1,1,1.5),
               type = "<=",
               rhs = 3600,
               indices = c(1:4)) #Reamer Time Constraint

add.constraint(bin_model,
               xt = c(1.3,0.6,0.4,0),
```

```r
                type = "<=",
                rhs = 3600,
                indices = c(1:4)) #Shaper Time Constraint

add.constraint(bin_model,
               xt = c(2.5,1.2,2,1.5),
               type = "<=",
               rhs = 3600,
               indices = c(1:4)) #Planer Time Constraint

add.constraint(bin_model,
               xt = c(1,-1440),
               type = "<=",
               rhs = 0,
               indices = c(1,5)) #linking contraint for A

add.constraint(bin_model,
               xt = c(1,-1636.36),
               type = "<=",
               rhs = 0,
               indices = c(2,6)) #linking contraint for B

add.constraint(bin_model,
               xt = c(1,-1200),
               type = "<=",
               rhs = 0,
               indices = c(3,7)) #linking contraint for C

add.constraint(bin_model,
               xt = c(1,-1300),
               type = "<=",
               rhs = 0,
               indices = c(4,8)) #linking contraint for D

#Setting lower and upper bounds
set.bounds(bin_model, lower = c(1200,450,0,0,0,0,0,0),
                      upper = c(Inf,650,120,150,1,1,1,1))

row.add.mode(bin_model,"off")

#Giving names to the contraints and variables
dimnames(bin_model)<- list(c("Aluminium","Copper","Forger","Reamer","Shaper","Planer",
                             "Linking A","Linking B","Linking C","Linking D"),
                           c("A","B","C","D","Y_A","Y_B","Y_C","Y_D"))
name.lp(bin_model, "Alloy Corporation Binary Problem")
```

```r
print(bin_model)
```

```
## Model name: Alloy Corporation Binary Problem
##                    A          B          C          D        Y_A        Y_B        Y_C        Y_D
## Maximize       2.141      1.125          4      2.116       -600       -500       -250       -300
## Aluminium       0.55       0.45       0.25          1          0          0          0          0  <=   1300
## Copper          0.45       0.55       0.75          0          0          0          0          0  <=    900
## Forger           1.2          1          1        1.5          0          0          0          0  <=   3600
```

```
## Reamer        1.75     1     1   1.5      0        0      0      0 <= 3600
## Shaper         1.3   0.6   0.4     0      0        0      0      0 <= 3600
## Planer         2.5   1.2     2   1.5      0        0      0      0 <= 3600
## Linking A        1     0     0     0  -1440        0      0      0 <=    0
## Linking B        0     1     0     0      0 -1636.36      0      0 <=    0
## Linking C        0     0     1     0      0        0  -1200      0 <=    0
## Linking D        0     0     0     1      0        0      0  -1300 <=    0
## Kind           Std   Std   Std   Std    Std      Std    Std    Std
## Type          Real  Real  Real  Real    Int      Int    Int    Int
## Upper          Inf   650   120   150      1        1      1      1
## Lower         1200   450     0     0      0        0      0      0
```

d. What is the optimal solution?

We use the solve() function to solve the **model**:

```
solve(model) #result = 0 -> solved
```

```
## [1] 0
```

```
solution <- get.variables(model)
cat("The optimal variables values are:","\n",
    "A <-",solution[1],", B <-",solution[2],
    ", C <-",solution[3],", D <-",solution[4])
```

```
## The optimal variables values are:
##  A <- 1200 , B <- 450 , C <- 30 , D <- 0
```

```
solution_value <- get.objective(model)
fixed_cost <- 1650
cat("The optimal objective function value is: ",solution_value, "\n",
    "Thus the optimal net profit value is",solution_value - fixed_cost)
```

```
## The optimal objective function value is:  3195.45
##  Thus the optimal net profit value is 1545.45
```

We use the solve() function to solve the **bin_model**:

```
solve(bin_model) #result = 0 -> solved
```

```
## [1] 0
```

```
bin_solution <- get.variables(bin_model)
cat("The optimal variables values are:","\n",
    "A <-",bin_solution[1],", B <-",bin_solution[2],
    ", C <-",bin_solution[3],", D <-",bin_solution[4] , "\n",
    "Y_A <-",bin_solution[5],", Y_B <-",bin_solution[6],
    ", Y_C <-",bin_solution[7],", Y_D <-",bin_solution[8])
```

```
## The optimal variables values are:
##  A <- 1200 , B <- 500 , C <- 0 , D <- 0
##  Y_A <- 1 , Y_B <- 1 , Y_C <- 0 , Y_D <- 0
```

```
cat("The optimal objective function value is: ",get.objective(bin_model))
```

```
## The optimal objective function value is:  2031.7
```

In the case the fixed cost is activated only on production, if we don't use the binary variable, we see that the the result given without is binary problem will be $1545.45 + 300 = 1845.45$ (Since it didn't produce Product

D), which is way off the optimal solution of 2031.7 using the binary variables.

e. Perform the sensitivity analysis for this solution and answer the following questions.

From here on we will consider only the non binary model

```
printSensitivityObj(model)
```

Objs	Sensitivity
C1	-inf <= C1 <= 5
C2	-inf <= C2 <= 2.4
C3	2.821 <= C3 <= inf
C4	-inf <= C4 <= 3

We have that the base remains if the coeffiecient are in the following range:

$$
\begin{aligned}
-\infty < C_1 \leq 5 \\
-\infty < C_2 \leq 2.4 \\
2.821 \leq C_3 < +\infty \\
-\infty < C_4 < 3
\end{aligned}
$$

The fact that C_1 and C_2 can decrease all they want is due to the fact that the optimal solution satisfies the minimum constraint given for Product A and B, thus if we decrease the marginal profit per pound from how it is now, the problem still needs to satify the minimum quantity for Product A and B, so the solution will not change.
A similar argument applies for the coefficient C_4 of Product D, if we decrease C_4 the solution will not change since we will continue to not produce the Product D with the given constraints.
While for the coefficient C_3 if we increase it, since we produce only the product C, because it's the most profitable while satifying the given contraints even if we increased the obtained profit we will have that the optimal solution will not change, but if we reduce C_3 too much it might change the optimal solution.

```
printSensitivityRHS(model)
```

Rhs	duals	Sensitivity
B1	0	-inf <= B1 <= inf
B2	0	-inf <= B2 <= inf
B3	0	-inf <= B3 <= inf
B4	0	-inf <= B4 <= inf
B5	0	-inf <= B5 <= inf
B6	2	3540 <= B6 <= 3780

So we have the the dual variables are

$$y_i = 0, \forall i = 1, ..., 5$$

and

$$y_6 = 2$$

and the interval for the resources are:

$$
\begin{aligned}
-\infty < b_1 < +\infty \\
-\infty < b_2 < +\infty \\
-\infty < b_3 < +\infty \\
-\infty < b_4 < +\infty \\
-\infty < b_5 < +\infty \\
3540 \leq b_6 \leq 3780
\end{aligned}
$$

All the shadow prices except for the last one are 0 thus a small change in these contraints with null shadow prices will not affect the solution.
The last contraint (Planer Time) has a shadow price of 2, which means that every minute we add/subtract to the planer time (staying in the given interval [3540,3780]), the value of the optimal solution increases/decreases by 2.

f. Is the solution degenerate?

There are two was of checking if a solution is degenrate, the first one is using the theory: since no constraint has an allowable increase or decrease = 0(from the table in point **e.**), the solution is not degenerate.
Another way is using the value of the following function:

```
solve.lpExtPtr(model)
```

```
## [1] 0
```

In the function description it is stated that if the result of the function has value 4, it means that the solution is degenerate, while 0 means that the optimal solution is found. In this case the value of the function confirms that the solution is not degenerate.

g. Is the solution unique?

```
get.solutioncount(model)
```

```
## [1] 1
```

This function counts the number of solution for the model, in this case the result is 1, thus there is only one unique solution.
We could also see that the first two quantities for Product A and B are the minimum market's requirement and the rest of the resources are allocated for the Product C which is the most profitable one. So it is reasonable to think (but not obvious) that the obtained solution is the only best possibility. For example after satifying the market's minimun requirement for the Product A and B, if we were to allocate the resources for any product but C, the net profit will not go up as much as with the Product C, given that the production time of C is more or less the same as the other products but has approximately twice the marginal profit compared to the other Products.

h. If MNC wanted to decrease the production on any product, which one would you recommend and why?

```r
#Use only the last 4 elements of the array
red_cost <- tail(get.sensitivity.rhs(model)$duals,4)
kable(data.frame("Variables"=c("A","B","C","D"),
          "Final_Quantity"=get.variables(model),
          "Reduced_Cost"=red_cost))
```

Variables	Final_Quantity	Reduced_Cost
A	1200	-2.859
B	450	-1.275
C	30	0.000
D	0	-0.884

One would be tempted to use the marginal profit per Pound, in this case he/she would recommend decreasing the Production of the Product B, in order to allocate the resources for a more profitable Product and using a similar idea as we mentioned in the point **a.**, but we can see that, using the Reduced Cost the Product which will gain us more value reducing its production is the Product A with a reduce cost of -2.859.

i. If MNC wanted to increase the production of any product, which one would you recommend and why?

In this case we would recommend the increase in production of the Product C, which is the most profitable one. All the other Products have a negative Reduced Cost, so increasing their Production will decrease the Net Profit since the consumed resources by them will reduce the quantity produced of Product C.

j. Which resources are preventing MNS from making more money? If they could acquire more of this resource how much should they acquire & how much should they be willing to pay to acquire it?

```r
get.constraints(model)
```

```
## [1]  870  810 1920 2580 1842 3600
```

The Planer time is the only resource which is preventing us to produce more products (because it's final value is $= 3600$ which is the maximum value it can have), since it's shadow price is $y_6 = 2$, if we buy $\Delta_6 = 180$ minutes more (which is the maximum amount that can be purchases without changing the optimal solution), the value of the function will increase by $y_6 \times \Delta_6 = 2 \times 180 = 360$. So the company must pay at most 360 to increase the time by 180 minutes in the Planer machine.

k. How much should MNC be willing to pay to acquire more copper?

From the point **j.** we can see that the quantity of Copper used is 810, while the company has an available quantity of 900, thus MNC doesn't need more copper, so the answer is 0.

l. If the marketing department wanted to decrease the price of the A product by $0.25, would the optimal solution change?

During the sensitivity analysis we saw that the the coefficient C_1 for the product A can decrease a lot without changing the optimal solution, so if we decrease it by 0.25 the optimal solution will not change, only the

value of the function will change.
We can confirm it (even though it's unnecessary) by creating a new model with the new coefficient for Product A.

```
model2 <- model #We just change the objective function
set.objfn(model2,obj=c(2.141-0.25,1.125,4,2.116)) #here we change the first coefficient
solve(model2)
```

```
## [1] 0
```

```
cat(" New Model: ", get.variables(model2),"\n",
    "Old Model: ", solution)
```

```
##  New Model:  1200 450 30 0
##  Old Model:  1200 450 30 0
```

```
cat(" New value of the function: ", get.objective(model2),"\n",
    "Old value of the function: ", solution_value)
```

```
##  New value of the function:  2895.45
##  Old value of the function:  3195.45
```