ST1401 Assignment 1

**Name:** Maxim Chopivskyy

**ID number:** 118364841

# Q1)

## (i)

Objective Function:

Minimise:

2000 + 8 + 180 + 320 + 400

i = month i

xi = carpets made in month i

wi = number of workers at end of month i

fi = number of workers fired in month i

hi = number of workers hired in month i

oi = number of carpets made in overtime in month i

si = number of stored carpets at end of month i

di = demand at end of month i (amount of carpets sold)

i = 1,2,3,4,5,6,7,8,9,10,11,12

### Constraints:

xi ≥ 0

wi ≥ 0 w0 = 30

fi ≥ 0

hi ≥ 0

oi ≥ 0

si ≥ 0 s0 = 0, s12 = 0

xi = 20wi + oi

oi ≤ 6wi

(overtime cannot make more than 6 regular hour carpets)

oi – 6wi ≤ 0

si = si-1 + xi – di

(re arrange formula)

di = xi + si-1 – si

(xi = 20wi + oi)

di = 20wi + oi + si-1 – si ≥ 440

di = 20wi + oi + si-1 -si ≤ 920

wi = wi-1 + hi – fi ≥ 0

## (ii)

### Code :

> library(linprog)

Loading required package: lpSolve

> library(lpSolve)

> cvec = c(rep(c(2000, 320, 400, 180, 8), 12))#the min function of every month

>

> names(cvec) = c(rep(c('w','h','f','o','s'),12))

>

> bvec = c(

+ rep(440, 12),

+ rep(920, 12),

+ rep(0, 12),

+

+ 0, #storage is 0 for last month

+

+ 30, #first month worker number

+ rep(0,11), #number of workers

+

+ rep(0,60) #every variable is at least 0

+ )

>

>

>

>

>

>

> Amat = rbind(c(20, 0, 0, 1, -1, rep(0, 55)),

+ c(rep(0,4), 1,20,0,0,1,-1, rep(0, 50)),

+ c(rep(0, 9),1,20,0,0,1,-1,rep(0,45)),

+ c(rep(0,14),1,20,0,0,1,-1,rep(0,40)),

+ c(rep(0,19),1,20,0,0,1,-1,rep(0,35)),

+ c(rep(0,24),1,20,0,0,1,-1,rep(0,30)),

+ c(rep(0,29),1,20,0,0,1,-1,rep(0,25)),

+ c(rep(0,34),1,20,0,0,1,-1,rep(0,20)),

+ c(rep(0,39),1,20,0,0,1,-1,rep(0,15)),

+ c(rep(0,44),1,20,0,0,1,-1,rep(0,10)),

+ c(rep(0,49),1,20,0,0,1,-1,rep(0,5)),

+ c(rep(0,54),1,20,0,0,1,-1),

+ #check if over 440

+

+ c(20, 0, 0, 1, -1, rep(0, 55)),

+ c(rep(0,4), 1,20,0,0,1,-1, rep(0, 50)),

+ c(rep(0, 9),1,20,0,0,1,-1,rep(0,45)),

+ c(rep(0,14),1,20,0,0,1,-1,rep(0,40)),

+ c(rep(0,19),1,20,0,0,1,-1,rep(0,35)),

+ c(rep(0,24),1,20,0,0,1,-1,rep(0,30)),

+ c(rep(0,29),1,20,0,0,1,-1,rep(0,25)),

+ c(rep(0,34),1,20,0,0,1,-1,rep(0,20)),

+ c(rep(0,39),1,20,0,0,1,-1,rep(0,15)),

+ c(rep(0,44),1,20,0,0,1,-1,rep(0,10)),

+ c(rep(0,49),1,20,0,0,1,-1,rep(0,5)),

+ c(rep(0,54),1,20,0,0,1,-1),

+ #check if under 920

+

+ c(-6,0,0,1,0,rep(0,55)),

+ c(rep(0,5),-6,0,0,1,0,rep(0,50)),

+ c(rep(0,10),-6,0,0,1,0,rep(0,45)),

+ c(rep(0,15),-6,0,0,1,0,rep(0,40)),

+ c(rep(0,20),-6,0,0,1,0,rep(0,35)),

+ c(rep(0,25),-6,0,0,1,0,rep(0,30)),

+ c(rep(0,30),-6,0,0,1,0,rep(0,25)),

+ c(rep(0,35),-6,0,0,1,0,rep(0,20)),

+ c(rep(0,40),-6,0,0,1,0,rep(0,15)),

+ c(rep(0,45),-6,0,0,1,0,rep(0,10)),

+ c(rep(0,50),-6,0,0,1,0,rep(0,5)),

+ c(rep(0,55),-6,0,0,1,0),

+ #checks if overtime is less than 30%

+

+

+

+ c(rep(0,59),1),

+ #checks if storage on last month is 0

+

+ c(1,-1,1,0,0,rep(0,55)),

+ c(-1,0,0,0,0,1,-1,1,rep(0,52)),

+ c(rep(0,5),-1,0,0,0,0,1,-1,1,rep(0,47)),

+ c(rep(0,10),-1,0,0,0,0,1,-1,1,rep(0,42)),

+ c(rep(0,15),-1,0,0,0,0,1,-1,1,rep(0,37)),

+ c(rep(0,20),-1,0,0,0,0,1,-1,1,rep(0,32)),

+ c(rep(0,25),-1,0,0,0,0,1,-1,1,rep(0,27)),

+ c(rep(0,30),-1,0,0,0,0,1,-1,1,rep(0,22)),

+ c(rep(0,35),-1,0,0,0,0,1,-1,1,rep(0,17)),

+ c(rep(0,40),-1,0,0,0,0,1,-1,1,rep(0,12)),

+ c(rep(0,45),-1,0,0,0,0,1,-1,1,rep(0,7)),

+ c(rep(0,50),-1,0,0,0,0,1,-1,1,rep(0,2))

+ #finds amount of w and h and f in month

+

+

+

+

+ )

>

>

> s=c(rep(0,60))

>

> for (x in 1:60){

+ s[x]=1

+

+ Amat = rbind(Amat,s)

+ s=c(rep(0,60))

+

+ }

> #lets us check every value of cvec individually

>

>

> solveLP(cvec, bvec, Amat, F,const.dir=c(rep('>=',12),rep('<=',12), rep('<=',12),'=', rep('=',12), rep('>=', 60)), lpSolve=TRUE)

### Output:

Results of Linear Programming / Linear Optimization

(using lpSolve)

Objective function (Minimum): 531200

Solution

opt

w 22

h 0

f 8

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

w 22

h 0

f 0

o 0

s 0

Constraints

actual dir bvec free

1 440 >= 440 0

2 440 >= 440 0

3 440 >= 440 0

4 440 >= 440 0

5 440 >= 440 0

6 440 >= 440 0

7 440 >= 440 0

8 440 >= 440 0

9 440 >= 440 0

10 440 >= 440 0

11 440 >= 440 0

12 440 >= 440 0

13 440 <= 920 480

14 440 <= 920 480

15 440 <= 920 480

16 440 <= 920 480

17 440 <= 920 480

18 440 <= 920 480

19 440 <= 920 480

20 440 <= 920 480

21 440 <= 920 480

22 440 <= 920 480

23 440 <= 920 480

24 440 <= 920 480

25 -132 <= 0 132

26 -132 <= 0 132

27 -132 <= 0 132

28 -132 <= 0 132

29 -132 <= 0 132

30 -132 <= 0 132

31 -132 <= 0 132

32 -132 <= 0 132

33 -132 <= 0 132

34 -132 <= 0 132

35 -132 <= 0 132

36 -132 <= 0 132

37 0 = 0 0

38 30 = 30 0

39 0 = 0 0

40 0 = 0 0

41 0 = 0 0

42 0 = 0 0

43 0 = 0 0

44 0 = 0 0

45 0 = 0 0

46 0 = 0 0

47 0 = 0 0

48 0 = 0 0

49 0 = 0 0

50 22 >= 0 22

51 0 >= 0 0

52 8 >= 0 8

53 0 >= 0 0

54 0 >= 0 0

55 22 >= 0 22

56 0 >= 0 0

57 0 >= 0 0

58 0 >= 0 0

59 0 >= 0 0

60 22 >= 0 22

61 0 >= 0 0

62 0 >= 0 0

63 0 >= 0 0

64 0 >= 0 0

65 22 >= 0 22

66 0 >= 0 0

67 0 >= 0 0

68 0 >= 0 0

69 0 >= 0 0

70 22 >= 0 22

71 0 >= 0 0

72 0 >= 0 0

73 0 >= 0 0

74 0 >= 0 0

75 22 >= 0 22

76 0 >= 0 0

77 0 >= 0 0

78 0 >= 0 0

79 0 >= 0 0

80 22 >= 0 22

81 0 >= 0 0

82 0 >= 0 0

83 0 >= 0 0

84 0 >= 0 0

85 22 >= 0 22

86 0 >= 0 0

87 0 >= 0 0

88 0 >= 0 0

89 0 >= 0 0

90 22 >= 0 22

91 0 >= 0 0

92 0 >= 0 0

93 0 >= 0 0

94 0 >= 0 0

95 22 >= 0 22

96 0 >= 0 0

97 0 >= 0 0

98 0 >= 0 0

99 0 >= 0 0

100 22 >= 0 22

101 0 >= 0 0

102 0 >= 0 0

103 0 >= 0 0

104 0 >= 0 0

105 22 >= 0 22

106 0 >= 0 0

107 0 >= 0 0

108 0 >= 0 0

109 0 >= 0 0

# Q2)

## (a)

### Objective Function :

### Minimise :

15 + 0.75 +

25 0.47Em-1,m + 1Em-2,m)

i = month in terms of m

m = month

Pm = number of units Produced

Si,m = number of units at Start of month m, made in month i

Ei,m = number of units at End of month m, made in month i

Di,m = Demand for units in month m, made in month i

Dm = Total demand for month m

(amount sold in month m)

### *Constraints*

Di,m ≥ 0

D3 = 4000

D4 = 6000

D5 = 7500

D6 = 7200

D7 = 8400

D8 = 8200

Pm ≥ 0

Pm ≤ 6000

P0 = 500

P1 = 2000

P2 = 1000

Ei,m ≥ 0

Si,m ≥ 0

Sm-1,m = 0.89(Em-1,m-1)

(re arrange formla)

Sm-1,m – 0.89(Em-1,m-1) = 0

Sm-2,m = 0.53(Em-2,m-1)

(re arrange formula)

Sm-2,m - 0.53(Em-2,m-1) = 0

Em,m = Pm – Dm,m

Em,m – Pm + Dm,m = 0

Em-1,m = Sm-1,m – Dm-1,m

Em-1,m – Sm-1,m + Dm-1,m = 0

Em-2,m = Sm-2,m – Dm-2,m

Em-2,m – Sm-2,m + Dm-2,m = 0

Dm-2,m + Dm-1,m + Dm,m ≥ Dm

Dm-2,m + Dm-1,m + Dm,m – Dm ≥ 0

## (b)

In my formulation of the problem, I would specify in my formulas that if ‘Dm’ was bigger than ‘Sm-2,m’, then I would specify that ‘Dm-2,m’ was equal to

‘Sm-2,m’, therefore Em-2,m would be 0. Then,

**Dm-1,m = Dm – Dm-2,m**

The same process applies to month m-1 and m.

If Dm-1,m is bigger than Sm-1,m,

**Em-1,m = 0**

**Dm,m = Dm-1,m – Sm-1,m**

If Sm-1,m is bigger than Dm-1,m, then

**Em-1,m = Sm-1,m – Dm-1,m**

**Em-1,m != 0**

If ‘Dm’ was smaller than the ‘Sm-2,m’, then ‘Dm-2,m’ would equal ‘Dm’ and:

**Em-2,m = Sm-2,m – Dm-2,m**

**Em-2,m != 0**

‘Em-1,m’ and ‘Em,m’ would stay the same as Sm-2,m.

*In summary, the earlier Ei,m variables will be calculated depending on the Dm.*