Quantitative management modelling:Assignment 2

(1)(Computer Center Staffing)You are the Director of the Computer Center for Gaillard College and responsible for scheduling the staffing of the center. It is open from 8 am until midnight. You have monitored the usage of the center at various times of the day and determinedthat the following numbers of computer consultants are required. Time of dayMinimum number of consultants required to be on duty8 am–noon 4Noon–4 pm84 am–8 pm108 am–midnight 6Two types of computer consultants can be hired: full-time and part-time. The full-time consultants work for eight consecutive hours in any of the following shifts: morning (8 am –4 pm), afternoon (noon –8 pm), and evening (4 pm –midnight). Full-time consult-ants are paid $14 per hour. Part-time consultants can be hired to work any of the four shifts listed in the table. Part-time consultants are paid $12 per hour. An additional requirement is that during every time period, at least one full-time consultant must be on duty for every part-time consultant on duty.

a)Determine a minimum-cost staffing plan for the center. In your solution, how many consultants will be paid to work full time and how many will be paid to work part time? What is the minimum cost?

(i)F_i = No. of full-time workers working in 3 shifts
(0800-1600hrs)
(1200-2000hrs)
(1600-0000hrs)
Where i= 1,2,3

(ii)P_i = No. of part-time workers working in 4 shifts
(0800-1200 hrs)
(1200-1600 hrs)
(1600-2000 hrs)
(2000-0000 hrs)
Where i= 1,2,3,4

Salary of full-time workers = $14/hr
Salary of part-time workers= $12/hr

(iii)Z_{min} = Minimum Cost
Z_{min}= (8X14X($F_1 + F_2 + F_3$)) +((4X12X(P_1 + P_2 + P_3 + P_4)

Constraints:

(i) $F_1 + P_1 \geq 4$
(ii) $F_1 + F_2 + P_2 \geq 8$
(iii) $F_2 + F_3 + P_3 \geq 10$
(iv) $F_3 + P_4 \geq 6$

(v) $F_1 \geq P_1$
(vi) $F_1 + F_2 \geq P_2$
(vii) $F_2 + F_3 \geq P_3$
(viii) $F_{3.} \geq P_4$
(ix) $F_i \geq 0$
(x) $P_i \geq 0$
(xi) $P_1 = 2, P_2 = 4, P_3 = 5, P_4 = 3$ & $F_1 = 2, F_2 = 2, F_3 = 3$

Z_{MIN} = 112 X (7) + 48 X (14) = 1456

Full-time workers = 7
Part-time workers = 14

b)After thinking about this problem for a while, you have decided to recognize meal breaks explicitly in the scheduling of full-time consultants. In particular, full-time consultants are entitled to a one-hour lunch break during their eight-hour shift. In addition, employment rules specify that the lunch break can start after three hours of work or after four hours of work, but those are the only alternatives. Part-time consultants do not receive a meal break. Under these conditions, find a minimum-cost staffing plan. What is the minimum cost?

Break to full-time workers = 1 hr
Break to part-time workers = 0

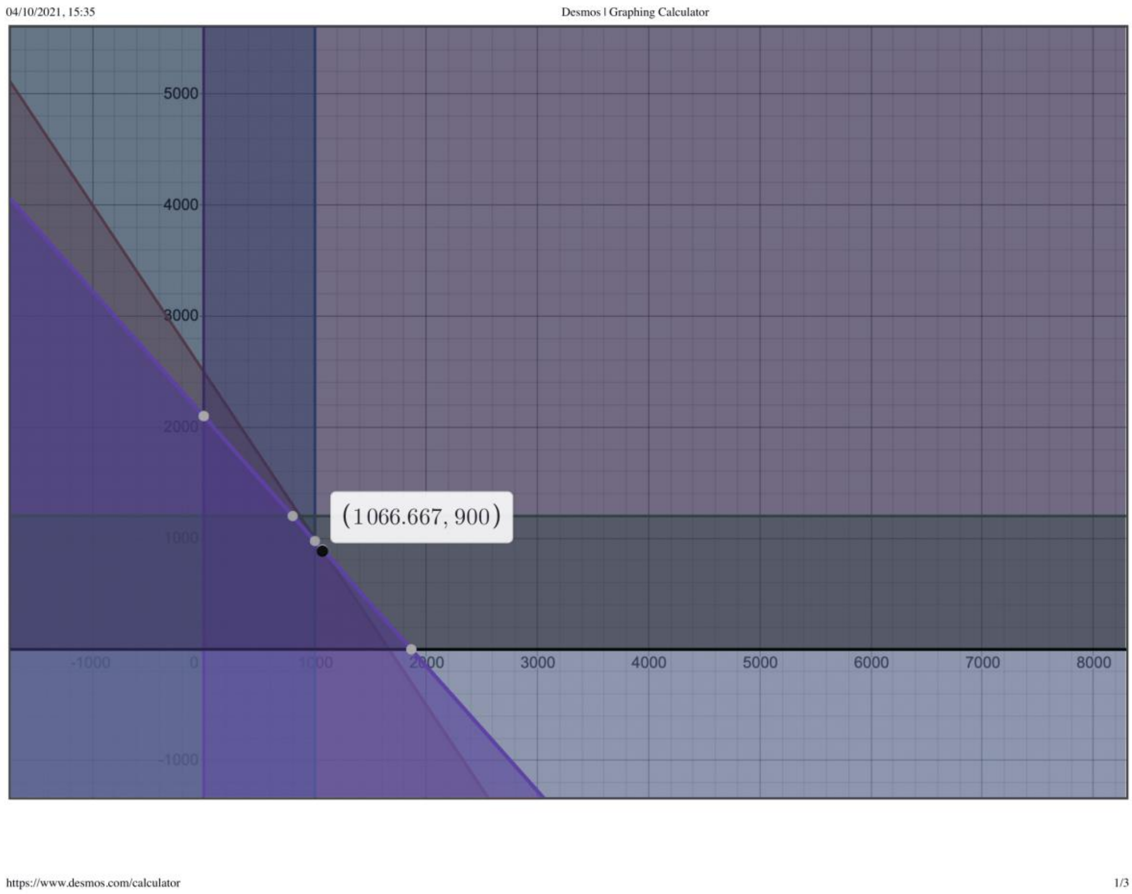
According to the problem , break should start in 3rd or 4th hour of the shift.
Salary removal from full-timers=1hr from 3 hr shifts
New equation:
$Z_{min1} = 8 \times 14 \times (F_1 + F_2 + F_3) - 14 \times (F_1 + F_2 + F_3) + 4 \times 12 \times (P_1 + P_2 + P_3 + P_4)$

The constraints will remain the same.
By putting the values

Z_{min1} = 112X(7) -14X(7) +48X(14) =1358

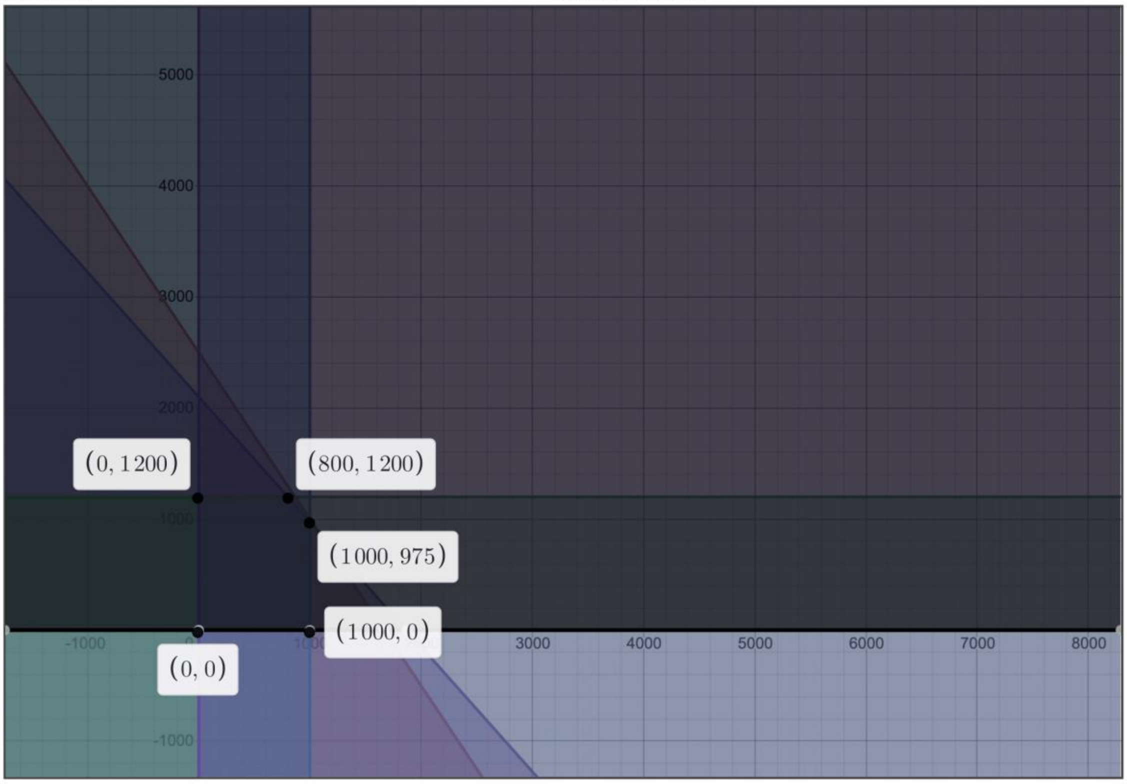
$Z_{min} - Z_{min1}$ = 1456-1358
= 98

(2)Back Savers is a company that produces backpacks primarily for students. They are considering offering some combination of two different models—the Collegiate and the Mini. Both are made out of the same rip-resistant nylon fabric. Back Savers has a long-term contract with a supplier of the nylon and receives a 5000 square-foot shipment of the material each week. Each Collegiate requires 3 square feet while each Mini requires 2 square feet. The sales forecasts indicate that at most 1000 Collegiates and 1200 Minis can be sold per week. Each Collegiate requires 45 minutes of labor to produce and generates a unit profit of $32. Each Mini requires 40 minutes of labor and generates a unit profit of $24. Back Savers has 35 laborers that each provides 40 hours of labor per week. Management wishes to know what quantity of each type of backpack to produce per week. Solve this problem graphically.



Here we can see that point (1066.667,900) is representing the optimal solution.
In the below image we can see that the area bounded by the points show feasible region.

04/10/2021, 15:36 Desmos | Graphing Calculator



https://www.desmos.com/calculator 1/3

(3)(Weigelt Production)The Weigelt Corporation has three branch plants with excess production capacity. Fortunately, the corporation has a new product ready to begin production, and all three plants have this capability, so some of the excess capacity can be used in this way. This product can be made in three sizes--large, medium, and small--that yield a net unit profit of $420, $360, and $300, respectively. Plants 1, 2, and 3 have the excess capacity to produce 750, 900, and 450 units per day of this product, respectively, regardless of the size or combination of sizes involved.The amount of available in-process storage space also imposes a limitation on the production rates of the new product. Plants 1, 2, and 3 have 13,000, 12,000, and 5,000 square feet, respectively, of in-process storage space available for a day's production of this product. Each unit of the large, medium, and small sizes produced per day requires 20, 15, and 12 square feet, respectively.Sales forecasts indicate that if available, 900, 1,200, and 750 units of the large, medium, and small sizes, respectively, would be sold per day.At each plant, some employees will need to be laid off unless most of the plant’s excess production capacity can be used to produce the new product. To avoid layoffs if possible, management has decided that the plants should use the same percentage of their excess capacity to produce the new product.Management wishes to know how much of each of the sizes should be produced by each of the plants to maximize profit.

(a)Define the decision variables

Decision Variables:
$\mathbf{A_{xy}}$ (x = 1,2,3 and y = l,m,s)
$\mathbf{A_{1l}, A_{1m}, A_{1s}}$ **:** variables for plant 1
$\mathbf{A_{2l}, A_{2m}, A_{2s}}$ **:** variables for plant 2
$\mathbf{A_{3l}, A_{3m}, Y_{3s}}$ **:** variables for plant 3

(b) Formulating linear programming model:

Let Z_{max} represent the maximum profit

$$\mathbf{Z_{max} = 420 \text{ X } (A_{1l} + A_{2l} + A_{3l}) + 360 \text{ X } (A_{1m} + A_{2m} + A_{3m}) + 300 \text{ X } (A_{1s} + A_{2s} + A_{3s})}$$

Constraints:

(i) $A_{1l} + A_{1m} + A_{1s} \leq 750$
(ii) $A_{2l} + A_{2m} + A_{2s} \leq 900$
(iii) $A_{3l} + A_{3m} + A_{3s} \leq 450$

Storage Space:

$20 \text{ X}(A_{1l}) + 15 \text{ X}(A_{1m}) + 12 \text{ X}(A_{1s}) \leq 13{,}000$
$20 \text{ X } (A_{2l}) + 15 \text{ X } (A_{2m}) + 12 \text{ X}(A_{2s}) \leq 12{,}000$
$20 \text{ X } (A_{3l}) + 15 \text{ X } (A_{3m}) + 12 \text{ X } (A_{3s}) \leq 5{,}000$

Percentage of Capacity:

$900 \text{ X } (A_{1l} + A_{2l} + A_{3l}) - 750 \text{ X } (A_{2l} + A_{2m} + A_{2s}) = 0$

$900 \text{ X } (A_{2l} + A_{2m} + A_{2s}) - 750 \text{ X } (A_{3l} + A_{3m} + A_{3s}) = 0$

$900 \text{ X } (A_{1l} + A_{Im} + A_{1s}) - 750 \text{ X } (A_{3l} + A_{3m} + A_{3s}) = 0$

$A_{1l} + A_{2l} + A_{3l} \leq 900$
$A_{1m} + A_{2m} + A_{3m} \leq 1200$
$A_{1s} + A_{2s} + A_{3s} \leq 750$

$A_{xy} \geq 0$ (x = 1,2,3 & y = l,m,s)

Solving the problem using lpsolveAPI

###Installing the library

```
library(lpSolveAPI)
```

###Setting the work directory

```
setwd("~/Desktop/SEMESTER 2/QUANT MANAGEMENT")
```

Creating a linear programming object with 0 constraints and 9 decision variables

```
lpmodel1 <- make.lp(0, 9)
lpmodel1

## Model name:
##   a linear program with 9 decision variables and 0 constraints
```

Creating the objective function. since we need to maximize profit, change the sense to max.

```
set.objfn(lpmodel1, c(420, 360, 300, 420, 360, 300, 420, 360, 300))
```

As we need to maximize profit, we need to change the sense to maximum.

```
lp.control(lpmodel1,sense='max')

## $anti.degen
## [1] "none"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
```

```
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

Adding the constraints

```
add.constraint(lpmodel1, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 750)
add.constraint(lpmodel1, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 900)
add.constraint(lpmodel1, c(0, 0, 0, 0, 0, 0,1, 1, 1), "<=", 450)
add.constraint(lpmodel1, c(20, 15, 12, 0, 0, 0, 0, 0, 0), "<=", 13000)
add.constraint(lpmodel1, c(0, 0, 0, 20, 15, 12, 0, 0, 0), "<=", 12000)
add.constraint(lpmodel1, c(0, 0, 0, 0, 0, 0, 20, 15, 12), "<=", 5000)
add.constraint(lpmodel1, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 900)
add.constraint(lpmodel1, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 1200)
add.constraint(lpmodel1, c(0, 0, 0, 0, 0, 0, 1, 1, 1), "<=", 750)
add.constraint(lpmodel1, c(6, 6, 6, -5, -5, -5, 0, 0, 0), "=", 0)
add.constraint(lpmodel1, c( 3, 3, 3, 0, 0, 0, -5, -5, -5), "=", 0)

set.bounds(lpmodel1, lower = c(0, 0, 0, 0, 0, 0, 0, 0, 0), columns = c(1,
2,3,4,5,6,7,8,9))
```

Setting variable names and name the constraints for identifying the variables and constraints

```
Rows <- c("CCon1", "CCon2", "CCon3", "SCon1", "SCon2", "SCon3", "S1Con1",
"S1Con2", "S1Con3", "%C1", "%C2")

Columns <- c("A1Large", "A1Medium", "A1Small", "A2Large", "A2Medium", "A2S
mall", "A3Large", "A3Medium", "A3Small")
```

```
dimnames(lpmodel1) <- list(Rows, Columns)

lpmodel1

## Model name:
##   a linear program with 9 decision variables and 11 constraints

write.lp(lpmodel1, filename = "Quantsassignment2.lp", type = "lp")

solve(lpmodel1)

## [1] 0

get.objective(lpmodel1)

## [1] 696000

get.variables(lpmodel1)

## [1] 516.6667 177.7778   0.0000   0.0000 666.6667 166.6667   0.0000   0.
0000
## [9] 416.6667
```