assignment 4

Noorah

10/24/2021

1. Heart Start produces automated external defibrillators (AEDs) in each of two different plants (A and B). The unit production costs and monthly production capacity of the two plants are indicated in the table below. The AEDs are sold through three wholesalers. The shipping cost from each plant to the warehouse of each wholesaler along with the monthly demand from each wholesaler are also indicated in the table. How many AEDs should be produced in each plant, and how should they be distributed to each of the three wholesaler warehouses so as to minimize the combined cost of production and shipping?

```
###including package
library(lpSolveAPI)
```

Plant A:

Plant A production cost = $600 Plant A| W 1 = $22 Plant A| W 2 = $14 Plant A| W 3 = $30

Plant B: Plant B production cost = $625 Plant B| W 1 = $16 Plant B| W 2 = $20 Plant B| W 3 = $24

Supply and Demand: Total demand = 210 Total supply = 220

We have supply more than demand

Objective Function: C = 622X1 + 614X2 + 630X3 + 641X4 + 645X5 + 649X6

Constraints:

Demand: x1 + x4 = 80 x2 + x5 = 60 X3 + X6 = 70 X7 + X8 = 10

Supply: X1 + X2 + X3 + X7 = 100 X4 + X5 + X6 + X8 = 120

```
q1 <- make.lp(0,8)
set.objfn(q1, c(622, 614, 630, 641, 645, 649, 0, 0))
lp.control(q1,sense="min")
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $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] "minimize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```
add.constraint(q1, c(1, 0, 0, 1, 0, 0, 0, 0), ">=", 80)
add.constraint(q1, c(0, 1, 0, 0, 1, 0, 0, 0), ">=", 60)
add.constraint(q1, c(0, 0, 1, 0, 0, 1, 0, 0), ">=", 70)
add.constraint(q1, c(0, 0, 0, 0, 0, 0, 1, 1), ">=", 10)
add.constraint(q1, c(1, 1, 1, 0, 0, 0, 1, 0), "<=", 100)
add.constraint(q1, c(0, 0, 0, 1, 1, 1, 0, 1), ">=", 120)

set.bounds(q1, lower = c(0, 0, 0, 0, 0, 0, 0, 0), columns =1:8)
solve(q1)
```

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

```
get.objective(q1)
```

```
## [1] 132790
```

```
get.variables(q1)
```

```
## [1]  0 60 40 80  0 30  0 10
```

In conclusion: -Total cost is $132790 -Plant A ship 60 AED to W 2 and 40 AED to W 3 -Plant B ship 80 AED to W 1 and 30 AED to W 3 -Plant B will reduce production by 10

2. Oil Distribution Texxon Oil Distributors, Inc., has three active oil wells in a west Texas oil field. Well 1 has a capacity of 93 thousand barrels per day (TBD), Well 2 can produce 88 TBD, and Well 3 can produce 95 TBD. The company has five refineries along the Gulf Coast, all of which have been operating at stable demand levels. In addition, three pump stations have been built to move the oil along the pipelines from the wells to the refineries. Oil can flow from any one of the wells to any of the pump stations, and from any one of the pump stations to any of the refineries, and Texxon is looking for a minimum cost schedule. The refineries' requirements are as follows.

Well production: Well #1 = 93 TBD Well #2 = 88 TBD Well #3 = 95 TBD

Refinery demand: R #1 = 30 TBD R #2 = 57 TBD R #3 = 48 TBD R #4 = 91 TBD R #5 = 48 TBD

Total supply: 93+88+95 = 276 TBD

Total demand: 30+57+48+91+48 = 274 TBD

Demand is higher than supply by 2 TBD

1. What is the minimum cost of providing oil to the refineries? Which wells are used to capacity in the optimal schedule? Formulation of the problem is enough.

Objective function:

C= 1.52x1 + 1.60x2 + 1.40x3 + 1.70x4 + 1.63x5 + 1.55x6 + 1.45x7 + 1.57x8+ 1.30x9 + 5.15x10 + 5.69x11 + 6.13x12 + 5.63x13 + 5.80x14 + 5.12x15 + 5.47x16 + 6.05x17 + 6.12x18 + 5.71x19 + 5.32x20 + 6.16x21 + 6.25x22 + 6.17x23 + 5.87x24

Well consrains: X1 + X2 + X3 =93 X4 + X5 + X6 =88 X7 + X8 + X9 =95 X25 + X26 + X27 =2

Refinery consrains: X10 + X15 + X20 =30 X11 + X16 + X21 =57 X12 + X17 + X22 =48 X13 + X18 + X23 =91 X14 + X19 + X24 =48

Pump constraints: -(X1 + X4 + X7) + (X10 +X11 + X12 + X13 + X14) = 0 -(X2 + X5 + X8) + (X15 +X16 + X17 + X18 + X19) = 0 -(X3 + X6 + X9) + (X20 +X21 + X22 + X23 + X24) = 0

```
q2 <- make.lp(0, 27)
set.objfn(q2, c(1.52, 1.60, 1.40, 1.70, 1.63, 1.55, 1.45, 1.57, 1.30, 5.15, 5.69, 6.13, 5.63, 5.80, 5.12, 5.47, 6
.05, 6.12, 5.71, 5.32, 6.16, 6.25, 6.17, 5.87, 0, 0, 0))
lp.control(q2, sense= 'min')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $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] "minimize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```
add.constraint(q2, c(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0), "<=", 93)
add.constraint(q2, c(0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0), "<=", 88)
add.constraint(q2, c(0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1), "<=", 95)
add.constraint(q2, c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0), "=", 2)
add.constraint(q2, c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0), "=", 30)
add.constraint(q2, c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0), "=", 57)
add.constraint(q2, c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0), "=", 48)
add.constraint(q2, c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0), "=", 91)
add.constraint(q2, c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1), "=", 48)
add.constraint(q2, c(-1, 0, 0, -1, 0, 0, -1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),"=", 0)
add.constraint(q2, c(0, -1, 0, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0),"=", 0)
add.constraint(q2, c(0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0),"=", 0)

set.bounds(q2, lower = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), column
s = 1:27)

solve(q2)
```

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

```
get.objective(q2)
```

```
## [1] 1963.82
```

```
get.variables(q2)
```

```
##  [1] 93  0  0  0 86  0 28  0 67 30  0  0 91  0  0 57 29  0  0  0  0 19  0 48  0
## [26]  2  0
```

-Well #1 and Well #3 used the full capacity -Well #2 reduce the capacity by 2 TBD
-Minimum cost is 1963.82 TBD

2. Show the network diagram corresponding to the solution in (a). That is, label each of the arcs in the solution and verify that the flows are consistent with the given information.

Please look at the png