Goal Programming

Chris Hargis

2022-11-05

x1 = Number of Units Contributed from Product 1

x2 = Number of Units Contributed from Product 2

x3 = Number of Units Contributed from Product 3

Employment Level:

y1 = 6x1 + 4x2 + 5x3 - 50

if y1 >= 0 then y1p if y1 < 0 then y1m

y1 is the difference in the number of employees from the given goal of 5,000 it will take to produce the number of all 3 widgets. This number could be positive or negative.

Earning next year:

y2 = 8x1 + 7x2 + 5x3 - 75

if y2 >= 0 then y2p (y2p means more earnings which is a good thing so we wouldnt penalize for that) if y2 < 0 then y2m

y2 is the same concept as y1 but for earning. Given goal is 75M and y2 will be the difference from that goal. We want more earning that 75M but not too much so we don't set high expectation for shareholders. This number could be positive or negative.

Management's Objective Function:

Maximize: Z = 20x1 + 15x2 + 25x3 - 6(y1p - y1m) - 3(y2m)

Profit Obj Function:

Maximize: Z = 20x1 + 15x2 + 25x3

Constraints:

6x1 + 4x2 + 5x3 - (y1p - y1n) = 50

8x1 + 7x2 + 5x3 - y2n >= 75

x1, x2, x3 >= 0

Solving the Linear programming model

```
library(lpSolveAPI)
```

```
## Warning: package 'lpSolveAPI' was built under R version 4.1.3
```

Objective Function

```r
lprec = make.lp(0, 3)

set.objfn(lprec, c(20, 15, 25))

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

```
## $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] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

Constraints

```
add.constraint(lprec, c(6, 4, 5), "=", 50)
add.constraint(lprec, c(8, 7, 5), ">=", 75)
```

```
solve(lprec)
```

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

```
get.objective(lprec)
```

```
## [1] 208.3333
```

```
get.variables(lprec)
```

```
## [1] 0.000000 8.333333 3.333333
```