Goal_Programming

Pardeepan Subramaniam

2023-11-27

Summary:

The company's objective function is to generate a profit of 225 million dollars, and this goal programming approach, in my observation, takes into account specific objectives and constraints related to workforce stability and earnings growth, making it easier to analyze production decisions in a systematic way.The business must combine X1, X2, and X3 in order to use the combination. To optimize profit, Product 3, the company's only offering, can only be manufactured in 15 units. However, X3 changed. Even though employee levels were supposed to stabilize at a maximum of 50,000 workers, the company in this case had 2,500 more workers (x1p). The overage would require payment of a penalty. Because of y2p and y2m, the earnings of the next year are assessed in respect to the current level; in this case, the current level is interpreted as "0," indicating that the earnings of the next year and the current year are the same.

Problem:

The Research and Development Division of the Emax Corporation has developed three new products. A decision now needs to be made on which mix of these products should be produced. Management wants primary consideration given to three factors: total profit, stability in the workforce, and achieving an increase in the company's earnings next year from the $75 million achieved this year. In particular, using the units given in the following table, they want to Maximize Z = P - 6C - 3D, where P = total (discounted) profit over the life of the new products, C = change (in either direction) in the current level of employment, D = decrease (if any) in next year's earnings from the current year's level. The amount of any increase in earnings does not enter into Z, because management is concerned primarily with just achieving some increase to keep the stockholders happy. (It has mixed feelings about a large increase that then would be difficult to surpass in subsequent years.)

Formulate and solve the linear programming model. What are your findings?

```r
library(lpSolveAPI)

x <-
read.lp("C:\\Users\\spard\\OneDrive\\Desktop\\QMM\\Assignment_4\\goal_programming.lp")
x
```

```
## Model name:
##             X1    X2    X3   Y1N   Y1M   Y2M   Y1P   Y2P
## Maximize    20    15    25    -6    -6    -3     0     0
## R1           6     4     5     0     1     0    -1     0  =  50
## R2           8     7     5     0     0     1     0    -1  =  75
## Kind       Std   Std   Std   Std   Std   Std   Std   Std
## Type      Real  Real  Real  Real  Real  Real  Real  Real
## Upper      Inf   Inf   Inf   Inf   Inf   Inf   Inf   Inf
## Lower        0     0     0     0     0     0     0     0
```

```r
# These terms, along with the necessary factors, are used to calculate deviations from predefined employment and earnings targets for the upcoming year.

solve(x)
```

```
## [1] 3
```

```r
get.objective(x)
```

```
## [1] -1e+30
```

```r
get.variables(x)
```

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

```r
get.constraints(x)
```

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

```r
# Create the matrix
x_tab <- matrix(c("Total Profit", "Employment Level", "Earnings Next Year",
                  20, 6, 8,
                  15, 4, 7,
                  25, 5, 5,
"Maximize", "=50", ">=75",
"Millions of Dollars", "Hundreds of Employees", "Millions of Dollars"), ncol = 6, byrow = FALSE)


# Set column names
colnames(x_tab) <- c("Factor", "Product 1", "Product 2", "Product 3", "Goal", "Units")
as.table(x_tab)
```

```
##   Factor             Product 1 Product 2 Product 3 Goal
## A Total Profit       20        15        25        Maximize
## B Employment Level   6         4         5         =50
## C Earnings Next Year 8         7         5         >=75
##   Units
## A Millions of Dollars
## B Hundreds of Employees
## C Millions of Dollars
```

```
get.sensitivity.rhs(x)
```

```
## $duals
##  [1]   0   5 -20 -20   0   0  -6  -8   0   5
##
## $dualsfrom
##  [1] -1.0e+30 -2.5e+01 -1.0e+30 -1.0e+30 -1.0e+30 -1.0e+30 -2.5e+01 -
1.0e+30
##  [9] -1.0e+30 -2.5e+01
##
## $dualstill
##  [1] 2.500000e+01 1.000000e+30 9.375000e+00 8.333333e+00 1.000000e+30
##  [6] 1.000000e+30 1.000000e+30 2.500000e+01 1.000000e+30 1.000000e+30
```

```
get.sensitivity.objex(x)
```

```
## $objfrom
## [1] -1e+30 -1e+30 -1e+30 -1e+30 -1e+30 -1e+30 -1e+30 -1e+30
##
## $objtill
## [1]  4.0e+01  3.5e+01  1.0e+30 -6.0e+00  0.0e+00  5.0e+00  6.0e+00 -
5.0e+00
##
## $objfromvalue
## [1]  9.375000e+00  8.333333e+00 -1.000000e+30 -1.000000e+30 -1.000000e+30
## [6]  2.500000e+01 -1.000000e+30 -1.000000e+30
##
## $objtillvalue
## [1] NA NA NA NA NA NA NA NA
```