Goal Programming

Jeetender Bhati

2024-11-04

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.3.3
```

```
df= data.frame(Factor=c("Total Profit","Employement Level","Earning next year"),
"1"=c(15,8,6),
"2"=c(12,6,5),
"3"=c(20,5,4),
Goal=c("Maximize","=70",">=60"))

df %>%
kable(align = "c") %>%
kable_classic() %>%
add_header_above(header = c(" "=1,"Product"=3," "=1)) %>%
add_header_above(header = c(" "=1,"Unit contribution"=3," "=1)) %>%
column_spec(1,border_right = TRUE) %>%
column_spec(4,border_right = TRUE) %>%
column_spec(5,border_right = TRUE)
```

	Unit contribution			
	Product			
Factor	X1	X2	X3	Goal
Total Profit	15	12	20	Maximize
Employement Level	8	6	5	=70
Earning next year	6	5	4	>=60

```
#Question.1
```

The production rates of the first, second, and third products are represented by x_1 , x_2 and x_3, respectively. We can express the constraints for these products as follows:

$8x_1 + 6x_2 + 5x_3 = 70$
$6x_1 + 5x_2 + 4x_3 \geq 60$

converted both the constraints in deviation form as below:

$y_1 = 8x_1 + 6x_2 + 5x_3 - 70$
$y_2 = 6x_1 + 5x_2 + 4x_3 - 60$

Here, the actual employment is represented by $8x_1 + 6x_2 + 5x_3$ and the employment requirement is 70. Thus, y_1 can be positive, negative, or zero, depending on whether the positive or negative part is larger. A similar explanation applies to the other constraints.

Let's define $y_i = y_i^+ - y_i^-$

That is,

$y_1 = y_1^+ - y_1^-$
$y_2 = y_2^+ - y_2^-$

Where,
y_1^+ is a positive deviation or over achievement of employment.
y_1^- is a negative deviation or under achievement of employment.
y_2^+ is a positive deviation or over achievement of earnings.
y_2^- is a negative deviation or under achievement of earnings.

Thus writing the above two constraints as:
$y_1^+ - y_1^- = 8x_1 + 6x_2 + 5x_3 - 70$
$y_2^+ - y_2^- = 6x_1 + 5x_2 + 4x_3 - 60$

math yields:
$8x_1 + 6x_2 + 5x_3 - (y_1^+ - y_1^-) = 70$
$6x_1 + 5x_2 + 4x_3 - (y_2^+ - y_2^-) = 60$

```
#Question.2
```

The objective function aims to maximize the overall goal, which is to ensure stable employment at 70 workers and maintain earnings of at least 60 million dollars. If the earnings fall below this threshold, a penalty of 2 is applied for each million-dollar decrease. However, there is no penalty for exceeding the earnings target (as this is a one-sided lower-bound profit constraint). Therefore, the total penalty applied to the profit will reflect the shortfall in earnings from the required 60 million dollars.

deviation $= 2y_2^-$.

Similarly, if the employment level deviates from the target of 70 workers, a penalty of 5 is applied for each unit of deviation, whether it is an increase or a decrease. Therefore, the total penalty from employment deviation will account for the difference from the target, with a penalty of 5 applied for every unit of deviation in either direction. $= 5y_1^+ + 5y_1^-$.

Thus Objective function is
$MAXZ = 15x_1 + 12x_2 + 20x_3 - 5(y_1^+ - y_1^-) - 2y_2^-$

Subjective function constraints
$8x_1 + 6x_2 + 5x_3 - (y_1^+ - y_i^-) = 70$
$6x_1 + 5x_2 + 4x_3 - (y_2^+ - y_2^-) = 60$

Non-negativity of the decision variables
$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$
$y_1^+ \geq 0, y_1^- \geq 0, y_2^+ \geq 0, y_2^- \geq 0$

```
#Question.3

library(lpSolveAPI)
```

```
## Warning: package 'lpSolveAPI' was built under R version 4.3.3
```

```r
lprec = make.lp(2,7)

set.objfn(lprec, c(15,12,20,-5,5,0,-2))

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"
```

```r
set.row(lprec, 1, c(8,6,5,-1,1,0,0), indices = c(1,2,3,4,5,6,7))

set.row(lprec, 2, c(6,5,4,0,0,-1,1), indices = c(1,2,3,4,5,6,7))

rhs = c(70,60)

set.rhs(lprec,rhs)

set.constr.type(lprec,c("=","="))

set.bounds(lprec,lower = rep(0,7))

lp.rownames = c("Employment","Earnings")
lp.colnames = c("x1","x2","x3","y1p","y1m","y2p","y2m")

solve(lprec)
```

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

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

```
## [1] 275
```

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

```
## [1]  0  0 15  5  0  0  0
```

Key Takeaways from the Goal Programming Problem

The linear programming problem was effectively resolved by applying the given constraints and objective function.

The best achievable objective value of the linear programming problem is 275.

The values of the decision variables in the optimal solution are:

$x_1 = 0$
$x_2 = 0$
$x_3 = 15$
$y_1^+ = 5$
$y_1^- = 0$
$y_2^+ = 0$
$y_2^- = 0$

These results illustrate the optimal solution for the goal programming problem. The values of the decision variables offer guidance on the suggested production levels and any deviations from the targets for each factor, all while adhering to the specified constraints and penalties. In this scenario, the aim was to maximize profit while considering employment and earnings.

It seems that only y_1^+ has a non-zero value, indicating a positive deviation from the employment goal (y_1). To determine the penalty related to y_1^+, you need to reference the penalty coefficient, which is 5 according to the code.

Therefore, the penalty for the deviation in employment is calculated as follows $= (5 * 5) = 25$

The penalty for earnings is zero since the associated variables (y_2^+, y_2^-) both have values of zero in the optimal solution.

As a result, the total penalty for the goal programming problem is $25, which is linked to the positive deviation in employment (y_1).