

Yeo Ying Xuan 1003835

40.014 Engineering Systems Architecture

Spring 2020

Lecture 17 Activity

# Activity 1

What are the three goals that management has established for addressing the production-planning problem?

- Achieve a total profit of at least \$125 million
- Maintain the current employment level of 4,000 employees
- Hold the capital investment down to no more than \$55 million

What is the order of importance associated to three goals?

1. Achieve a total profit of at least \$125 million
2. Avoid decreasing current employment level of 4,000 employees
3. Hold the capital investment down to no more than \$55 million
4. Avoid increasing current employment level of 4,000 employees

What are the decision variables?

- P1: Product 1
- P2: Product 2
- P3: Product 3

# Activity 1a

```
##### Activity 1a

# Problem formulation -- FILL IN THE LINE BELOW
#
# Minimize [5 n1, 4 n2, 3 p3, 2 p2]
# subject to:
# 12 P1 + 9 P2 + 15 P3 - p1 + n1 = 125
# 5 P1 + 3 P2 + 4 P3 - p2 + n2 = 40
# 5 P1 + 7 P2 + 8 P3 - p3 + n3 = 55
# P1, P2, P3 >= 0
# n1, n2, n3 >= 0
# p1, p2, p3 >= 0

# Let's rewrite the problem formulation to account for the relative
importance of all goals -- FILL IN THE LINE BELOW
# ie. rewrite the same objective twice but with two deviation variables to
split up the priority of one single objective (becomes 2 partial
objectives)
# Minimize [5 n1, 4 n2, 3 p3, 2 p4]
# subject to:
# 12 P1 + 9 P2 + 15 P3 - p1 + n1 = 125
# 5 P1 + 3 P2 + 4 P3 - p2 + n2 = 40
# 5 P1 + 7 P2 + 8 P3 - p3 + n3 = 55
# 5 P1 + 3 P2 + 4 P3 - p4 + n4 = 40
# P1, P2, P3 >= 0
# n1, n2, n3, n4 >= 0
# p1, p2, p3, p4 >= 0

# To set up the problem with goalprog, we must define three variables
#
# A matrix of coefficients for the linear objective functions -- FILL IN
THE LINE BELOW
coefficients_1a <- matrix (c(12, 9, 15, 5, 3, 4, 5, 7, 8, 5, 3, 4), nrow=4,
byrow=TRUE)
# A vector of target values for the objective functions -- FILL IN THE LINE
BELOW
targets_1a <- c(125, 40, 55, 40)
```

```

# A data frame with the deviation variables for each objective together
with the priority level
objective_1a <- c(1, 2, 3, 4)      # index for a particular problem object
priority_1a <- c(1, 2, 3, 4)       # level to which the row (i.e. objective)
is assigned
# priority level based on objective: first value of vector referring to
first objective (goal)
p_1a <- c(0, 0, 3, 2)            # penalty associated with the positive
deviation variable
n_1a <- c(5, 4, 0, 0)            # penalty associated with the negative
deviation variable
# if penalty value is not given, use binary 0 and 1 to indicate existence
of penalty
achievements_1a <- data.frame(objective=objective_1a, priority=priority_1a,
p=p_1a, n=n_1a)

# To solve the problem, we use the function llgp -- FILL IN THE LINE BELOW
soln_1a <- llgp(coefficients_1a, targets_1a, achievements_1a)

# Finally, we retrieve the most interesting aspect of the solution
out_1a <- llgpout(soln_1a$tab, coefficients_1a, targets_1a)
#
# Decision variables
# X
# X1  8.333333e+00
# X2  0.000000e+00
# X3  1.666667e+00
#
# Summary of objectives
# Objective          Over        Under       Target
# G1    1.250000e+02  0.000000e+00  0.000000e+00  1.250000e+02
# G2    4.833333e+01  8.333333e+00  0.000000e+00  4.000000e+01
# G3    5.500000e+01  0.000000e+00  0.000000e+00  5.500000e+01
# G4    4.833333e+01  8.333333e+00  0.000000e+00  4.000000e+01
#
#
# G1: exactly hit the target (both over and under are 0)
# G2: go slightly over but did not go under, which is the most important
thing (did not go under)
# (because goal was to minimise negative deviation n2)
# G3: exactly hit targets
# G4: went over target slightly
#
# able to meet the first three goals, and not the last one
# ie. only paying penalty for the fourth goal (shown in achievements below)

```

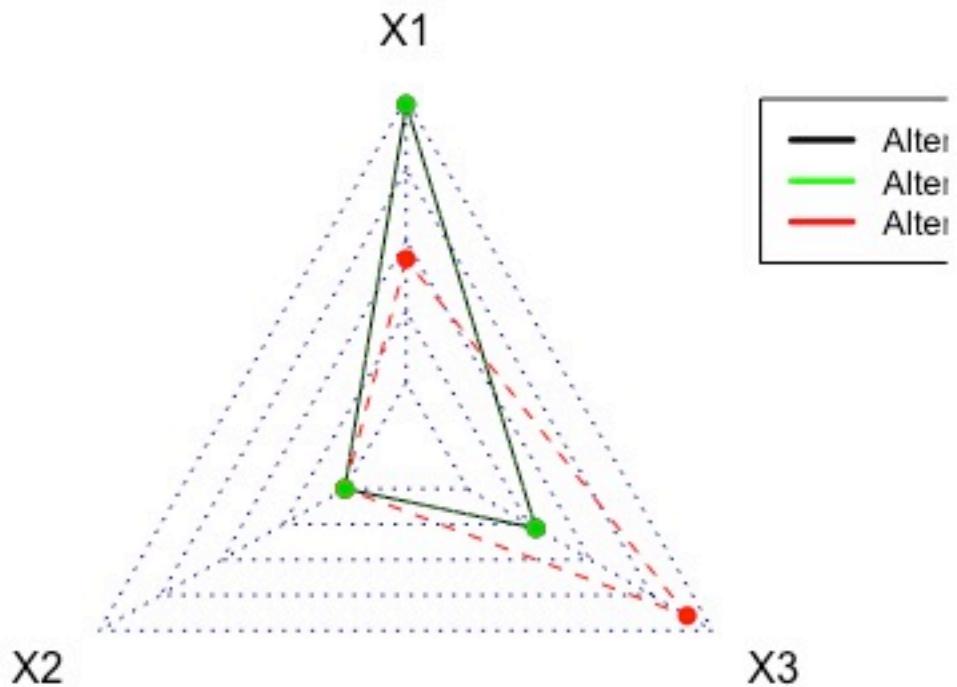
```

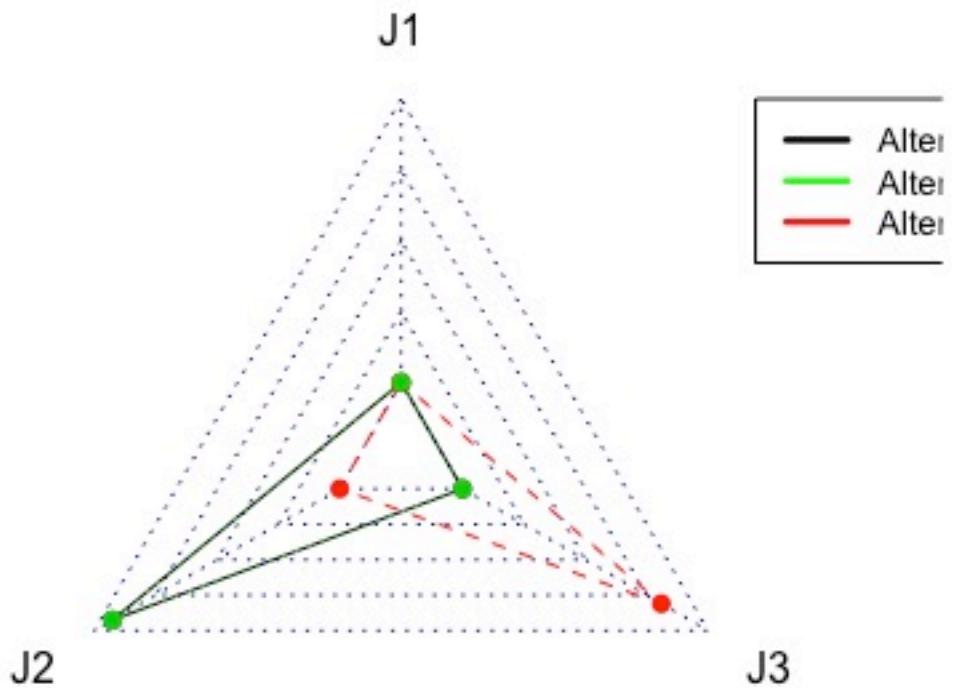
# Achievement function
# indicates the penalty we are paying
# A
# P1  0.000000e+00
# P2  0.000000e+00
# P3  0.000000e+00
# P4  1.666667e+01
# amount of penalty paying for P4: 2 * the deviation 1.666667e+01

# Let's try with the following combinations of lexicographic ordering --
# FILL IN THE LINES BELOW
priority_1a2 <- c(1, 2, 3, 2) # giving second priority to both second and
# fourth goal
priority_1a3 <- c(1, 3, 2, 3)
achievements_1a2 <- data.frame(objective=objective_1a,
priority=priority_1a2, p=p_1a, n=n_1a)
achievements_1a3 <- data.frame(objective=objective_1a,
priority=priority_1a3, p=p_1a, n=n_1a)
soln_1a2 <- llgp(coefficients_1a, targets_1a, achievements_1a2)
soln_1a3 <- llgp(coefficients_1a, targets_1a, achievements_1a3)
out_1a2 <- llgpout(soln_1a2$tab, coefficients_1a, targets_1a)
out_1a3 <- llgpout(soln_1a3$tab, coefficients_1a, targets_1a)
#
# Finally, we do a graphical comparison of the decisions and objective
functions for all problems
#
# Load the package for radar charts
library(fmsb)
#
# Decision variables
mydata_sol_1a <- data.frame(X1=c(out_1a$x[1],out_1a2$x[1],out_1a3$x[1]),
X2=c(out_1a$x[2],out_1a2$x[2],out_1a3$x[2]),
X3=c(out_1a$x[3],out_1a2$x[3],out_1a3$x[3]))
mydata_sol_1a = rbind(c(8.5,3,6),c(0,0,0),mydata_sol_1a) # To use the fmsb
package, we have to add 2 lines to the dataframe: the max and min of each
topic to show on the plot!
radarchart(mydata_sol_1a)
legend(1,1,lty=c(1,1),lwd=c(2.5,2.5),col=c("black","green","red"),c("Alternative 1","Alternative 2","Alternative 3"),cex=0.8)
# plot interpretation: first and second configuration of decision variables
do not make any difference
#
# Objective functions -- FILL IN THE LINES BELOW

```

```
mydata_obj_1a <- data.frame(J1=c(abs(out_1a$b[1]-  
out_1a$f[1]),abs(out_1a2$b[1]-out_1a2$f[1]),abs(out_1a3$b[1]-  
out_1a3$f[1])), J2=c(abs(out_1a$b[2]-out_1a$f[2]),abs(out_1a2$b[2]-  
out_1a2$f[2]),abs(out_1a3$b[2]-out_1a3$f[2])), J3=c(abs(out_1a$b[3]-  
out_1a$f[3]),abs(out_1a2$b[3]-out_1a2$f[3]),abs(out_1a3$b[3]-  
out_1a3$f[3]))  
mydata_obj_1a = rbind(c(2, 9, 8), c(0, 0, 0), mydata_obj_1a) # To use the  
fmsb package, we have to add 2 lines to the dataframe: the max and min of  
each topic to show on the plot!  
radarchart(mydata_obj_1a)  
legend(1,1,lty=c(1,1),lwd=c(2.5,2.5),col=c("black","green","red"),c("Alternative 1","Alternative 2","Alternative 3"),cex=0.8)
```





# Activity 1b

```
##### Activity 1b

# Problem formulation
#
# Minimize [5 n1 + 4 n2 + 3 p3 + 2 p2]
# the coefficient values above are not the weights, but the deviation
coefficients (penalty values)
# not a vector, but one single objective function
# subject to:
# 12 P1 + 9 P2 + 15 P3 - p1 + n1 = 125
# 5 P1 + 3 P2 + 4 P3 - p2 + n2 = 40
# 5 P1 + 7 P2 + 8 P3 - p3 + n3 = 55
# P1, P2, P3 >= 0
# n1, n2, n3 >= 0
# p1, p2, p3 >= 0

# Let's rewrite the problem formulation to account for the relative
importance of all goals
#
# Minimize [5 n1 + 4 n2 + 3 p3 + 2 p4]
# subject to:
# 12 P1 + 9 P2 + 15 P3 - p1 + n1 = 125
# 5 P1 + 3 P2 + 4 P3 - p2 + n2 = 40
# 5 P1 + 7 P2 + 8 P3 - p3 + n3 = 55
# 5 P1 + 3 P2 + 4 P3 - p4 + n4 = 40
# P1, P2, P3 >= 0
# n1, n2, n3, n4 >= 0
# p1, p2, p3, p4 >= 0

# To set up the problem with goalprog, we must define three variables
#
# A matrix of coefficients for the linear objective functions
coefficients_1b <- matrix (c(12, 9, 15, 5, 3, 4, 5, 7, 8, 5, 3, 4), nrow=4,
byrow=TRUE)
# A vector of target values for the objective functions
targets_1b <- c(125, 40, 55, 40)
```

```

# A data frame with the deviation variables for each objective together
with the priority level -- FILL IN THE ROW BELOW
objective_1b <- c(1, 2, 3, 4)      # index for a particular problem object
priority_1b <- c(1, 1, 1, 1)       # level to which the row (i.e. objective)
is assigned
# priority given to each deviation is equal: ie. set all values to 1
# since in the objective function is summed, there is no point of setting
different priority levels either
p_1b <- c(0, 0, 3, 2)            # penalty associated with the positive
deviation variable
n_1b <- c(5, 4, 0, 0)            # penalty associated with the negative
deviation variable
w_1b <- c(4, 3, 1, 2)            # weight associated with the specific
priority level
# since no weight is given, set the w
achievements_1b <- data.frame(objective=objective_1b, priority=priority_1b,
p=p_1b, n=n_1b, w=w_1b)

# To solve the problem, we use the function llgp
soln_1b <- llgp(coefficients_1b, targets_1b, achievements_1b)

# Finally, we retrieve the most interesting aspect of the solution
out_1b <- llgpout(soln_1b$tab, coefficients_1b, targets_1b)
#
# Decision variables
# X
# X1  8.333333e+00
# X2  0.000000e+00
# X3  1.666667e+00
#
# Summary of objectives
# Objective          Over        Under       Target
# G1    1.250000e+02  0.000000e+00  0.000000e+00  1.250000e+02
# G2    4.833333e+01  8.333333e+00  0.000000e+00  4.000000e+01
# G3    5.500000e+01  0.000000e+00  0.000000e+00  5.500000e+01
# G4    4.833333e+01  8.333333e+00  0.000000e+00  4.000000e+01
#
# Achievement function
# A
# P1   1.666667e+01

```