Yeo Ying Xuan 1003835

40.014 Engineering Systems Architecture

Spring 2020

Lecture 16

Table of Contents

Activity 1a

Activity 1b

Activity 1c

Activity 1a

```r
# We want to solve two single-objective problems.

# Problem 1: maximize power generation
#
# maximize 7750 X1 + 3250 X2 + 2250 X3 + 1050 X4 + 90 X5 (in GWh)
# subject to:
# cost
# 248 X1 + 206.375 X2 + 102.9375 X3 + 62.2125 X4 + 18.045 X5 <= 3400
# CO2 emissions
# 2.6 X2 + 0.7875 X3 <= 20
# other pollutions
# 5.8125 X1 + 30.0625 X2 + 9.5625 X3 + 0.525 X4 + 0.045 X5 <= 250
# max number of plants
# X2 <= 20
# X3 <= 20
# X4 <= 30
# X5 <= 100
# X1, X2, X3, X4, X5 integers
# X1, X2, X3, X4, X5 represent nuclear plants, coal plants, CCGT plants, wind turbines, solar cells
#
# Define inputs for the function lp -- FILL IN THE LINES BELOW
f.obj_1 <- c(7750, 3250, 2250, 1050, 90)
con_1_1 <- c(248, 206.375, 102.9375, 62.2125, 18.045)
con_2_1 <- c(0, 2.6, 0.7875, 0, 0)
con_3_1 <- c(5.8125, 30.0625, 9.5625, 0.525, 0.045)
con_4_1 <- c(0, 1, 0, 0, 0)
con_5_1 <- c(0, 0, 1, 0, 0)
con_6_1 <- c(0, 0, 0, 1, 0)
con_7_1 <- c(0, 0, 0, 0, 1)
f.con_1 <- rbind(con_1_1, con_2_1, con_3_1, con_4_1, con_5_1, con_6_1, con_7_1)
f.dir_1 <- c("<=", "<=", "<=", "<=", "<=", "<=", "<=")
f.rhs_1 <- c(3400, 20, 250, 20, 20, 30, 100)
#
# Let's now solve the problem
#
result_1 <- lp ("max", f.obj_1, f.con_1, f.dir_1, f.rhs_1, all.int=TRUE)
```

```r
result_1$objval    # 104050
result_1$solution  # 13 0 1 1 0
# This solution suggests to build 13 nuclear plants, 0 coal plants, 1 CCGT plant, 1 wind turbine, 0 solar cell

# Problem 2: to minimize the risks associated to the use of nuclear plants, we can simply remove X1 from the problem.
# remove all X1 from the problem formulation (objective and constraints)
#
# maximize 3250 X2 + 2250 X3 + 1050 X4 + 90 X5 (in GWh)
# subject to:
# cost
# 206.375 X2 + 102.9375 X3 + 62.2125 X4 + 18.045 X5 <= 3400
# CO2 emissions
# 2.6 X2 + 0.7875 X3 <= 20
# other pollutions
# 30.0625 X2 + 9.5625 X3 + 0.525 X4 + 0.045 X5 <= 250
# X2 <= 20
# X3 <= 20
# X4 <= 30
# X5 <= 100
# X2, X3, X4, X5 integers
# X2, X3, X4, X5 represent coal plants, CCGT plants, wind turbines, solar cells
#
# Define inputs for the function lp -- FILL IN THE LINES BELOW
f.obj_2 <- c(3250, 2250, 1050, 90)
con_1_2 <- c(206.375, 102.9375, 62.2125, 18.045)
con_2_2 <- c(2.6, 0.7875, 0, 0)
con_3_2 <- c(30.0625, 9.5625, 0.525, 0.045)
con_4_2 <- c(1, 0, 0, 0)
con_5_2 <- c(0, 1, 0, 0)
con_6_2 <- c(0, 0, 1, 0)
con_7_2 <- c(0, 0, 0, 1)
f.con_2 <- rbind(con_1_2, con_2_2, con_3_2, con_4_2, con_5_2, con_6_2, con_7_2)
f.dir_2 <- c("<=", "<=", "<=", "<=", "<=", "<=", "<=")
f.rhs_2 <- c(3400, 20, 250, 20, 20, 30, 100)
#
# Let's now solve the problem
#
result_2 <- lp ("max", f.obj_2, f.con_2, f.dir_2, f.rhs_2, all.int=TRUE)
result_2$objval    # 67150
result_2$solution  # 1 20 18 0
```

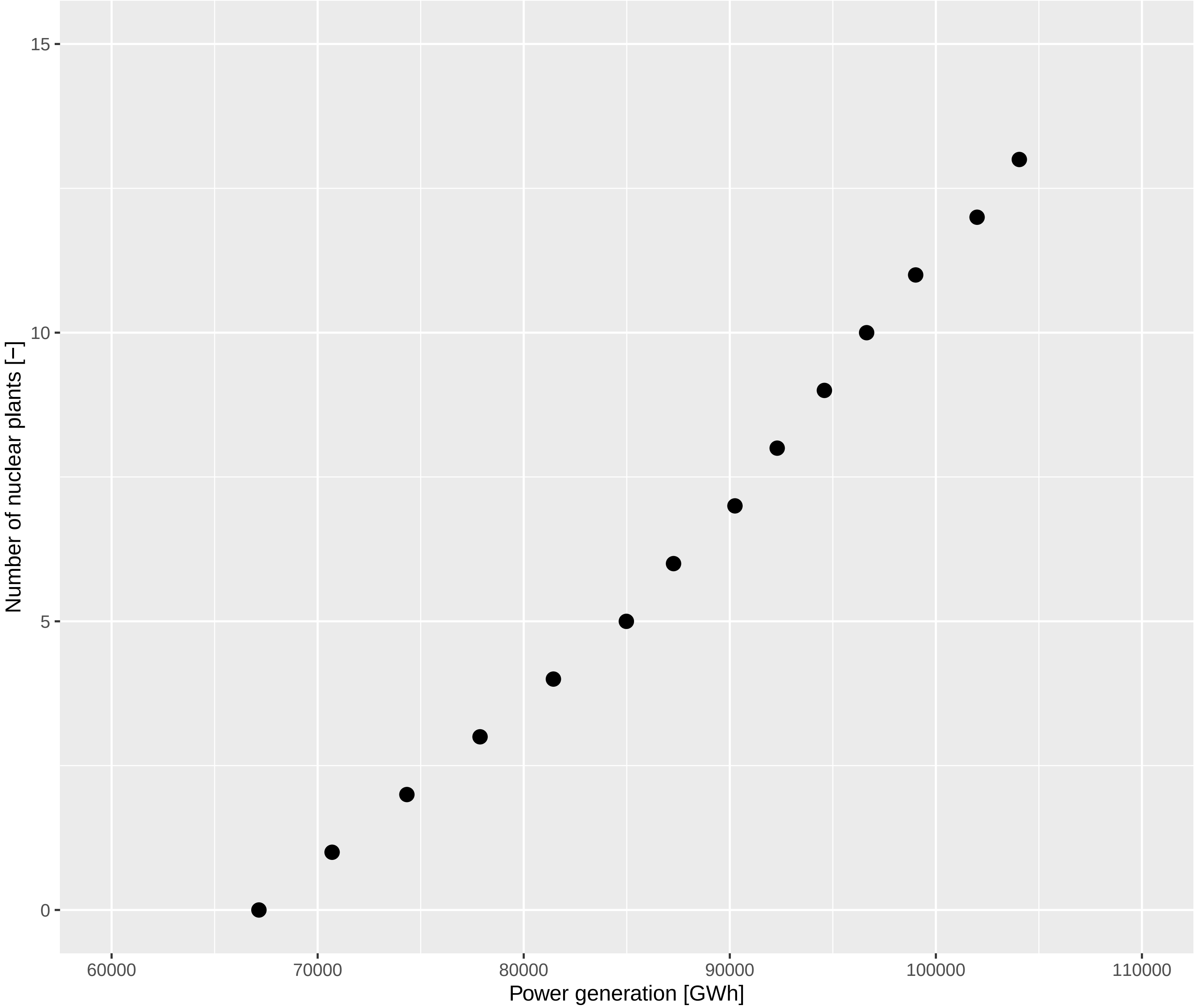
```
# The 6 nuclear plants found in Problem 1 are repalced by 1 coal plant, 20 CCGT plants, 18 wind turbines
```

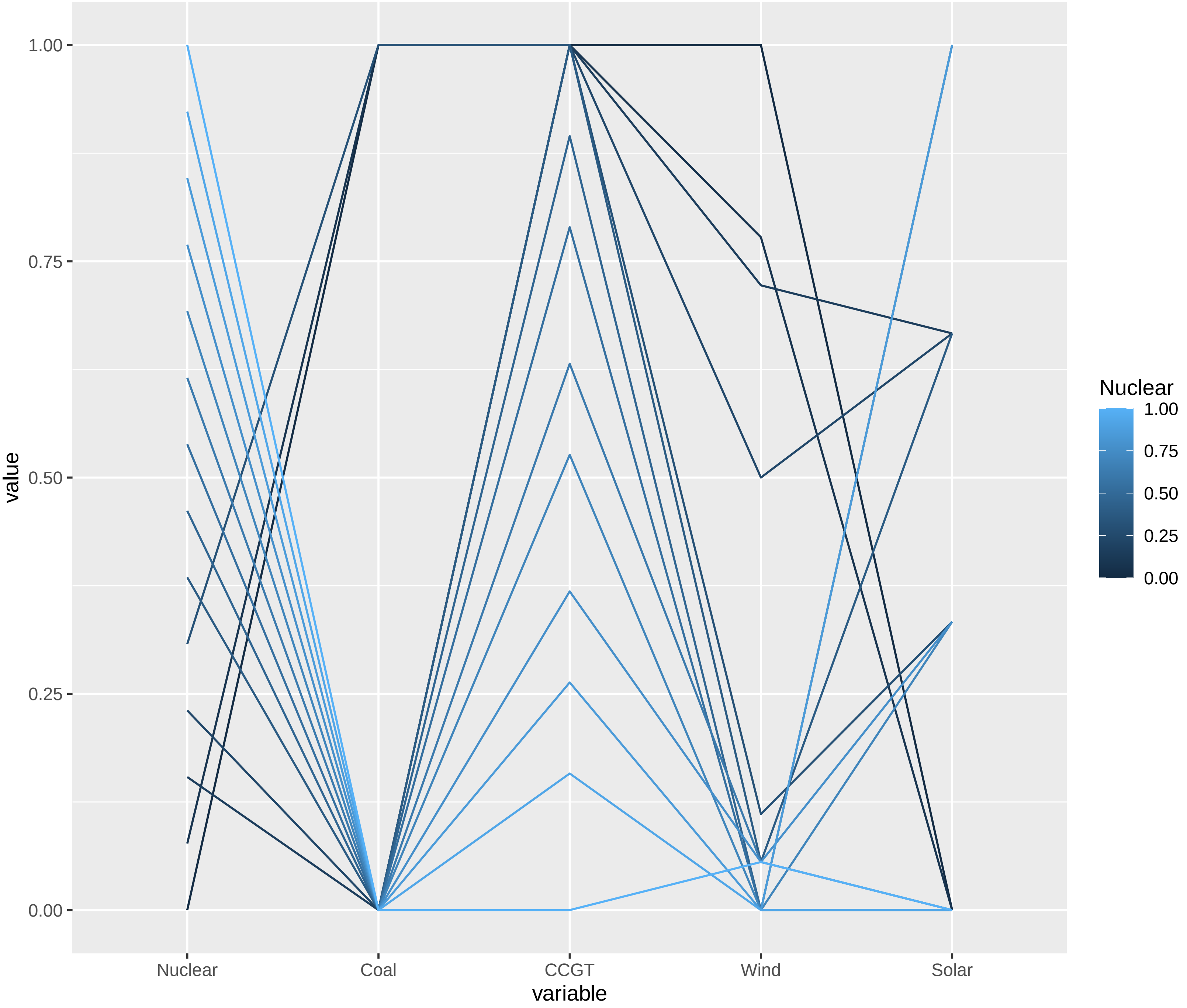
Activity 1b

```
# We apply the epsilon-constraint method to solve the two-objective problem

# Problem: maximize power generation, subject to a constraint on the number
of nuclear plants.
#
# maximize 7740 x1 + 3250 X2 + 2250 X3 + 1050 X4 + 90 X5 (in GWh)
# subject to:
# 248 X1 + 206.375 X2 + 102.9375 X3 + 62.2125 X4 + 18.045 X5 <= 3400 (in
million SGD)
# 0 X1 + 2.6 X2 + 0.7875 X3 + 0 X4 + 0 X5 <= 20 (in million tonnes)
# 5.8125 X1 + 30.0625 X2 + 9.5625 X3 + 0.525 X4 + 0.045 X5 <= 250 (in
million tonnes)
# X2 <= 20
# X3 <= 20
# X4 <= 30
# X5 <= 100
# X1 <= epsilon
# X1, X2, X3, X4, X5 integers
# X1, X2, X3, X4, X5 represent nuclear plants, coal plants, CCGT plants,
wind turbines, solar cells
#
# from above, we know that number of nuclear plants cannot be smaller than
0 and larger than 13
# ie. 0 <= X1 <= 13
#
# Define inputs for the function lp
f.obj_eps <- c(7750, 3250, 2250, 1050, 90)
con_1_eps <- c(248, 206.375, 102.9375, 62.2125, 18.045)
con_2_eps <- c(0, 2.6, 0.7875, 0, 0)
con_3_eps <- c(5.8125, 30.0625, 9.5625, 0.525, 0.045)
con_4_eps <- c(0, 1, 0, 0, 0)
con_5_eps <- c(0, 0, 1, 0, 0)
con_6_eps <- c(0, 0, 0, 1, 0)
con_7_eps <- c(0, 0, 0, 0, 1)
con_8_eps <- c(1, 0, 0, 0, 0)
f.con_eps <- rbind(con_1_eps, con_2_eps, con_3_eps, con_4_eps, con_5_eps,
con_6_eps, con_7_eps, con_8_eps)
f.dir_eps <- c("<=", "<=", "<=", "<=", "<=", "<=", "<=", "<=")
```

```
epsilon <- c(0:13)
#
# We create a for loop to solve all problems for the different values of epsilon
# We initialise two matrices to store the values of objective functions and decision variables
# 2 columns, one for each objective, 14 rows for each instance of the objective to solve (epsilon 0 to 13)
# the two objectives are: maximise power production and minimise number of nuclear plants
objval_eps <- matrix( c(rep(0,14*2)), nrow =14, byrow=TRUE)
# to keep track of the value of each objective value of the 14 epsilon values
decvar_eps <- matrix( c(rep(0,14*5)), nrow =14, byrow=TRUE)

for (i in 1:14){
  # define the numerical value of the constraint epsilon -- FILL IN THE LINE BELOW
  f.rhs_eps <- c(3400, 20, 250, 20, 20, 30, 100, epsilon[i])
  # solve the optimization problem
  result <- lp ("max", f.obj_eps, f.con_eps, f.dir_eps, f.rhs_eps, all.int=TRUE)
  # save the value of objective functions and decision variables
  objval_eps[i,1] <- result$objval
  objval_eps[i,2] <- epsilon[i]
  decvar_eps[i,]  <- result$solution
}
#
# Finally, we plot the solutions in both objective and decision space
#
# Objective space
b_plot1 <- ggplot(data = data.frame(Power.Generation=objval_eps[,1], Nuclear.Plants=objval_eps[,2]),
       mapping = aes(x = Power.Generation, y = Nuclear.Plants)) +
  geom_point(size=3) +
  coord_cartesian(xlim=c(60*10^3,110*10^3), ylim=c(0,15)) +
  labs(x="Power generation [GWh]", y="Number of nuclear plants [-]")
b_plot1
ggsave("1b Plot1.pdf", b_plot1)
#
# Decision space
# It is difficult to use a scatter plot, because we have 5 decision variables.
# Let's try with a parallel coordinate plot
```

```r
b_plot2 <-
ggparcoord(data.frame(Nuclear=decvar_eps[,1],Coal=decvar_eps[,2],CCGT=decvar_eps[,3], Wind=decvar_eps[,4], Solar=decvar_eps[,5]),
           columns = 1:5, groupColumn = "Nuclear", scale = "uniminmax")
b_plot2
# The plot shows that the number of CCGT plants and Wind turbines increases
# as the number of nulcear plants is reduced (and viceversa)!
ggsave("1b Plot2.pdf", b_plot2)
```


15
10
5
0
60000
70000
80000
90000
100000
110000
Number of nuclear plants [−]
Power generation [GWh]

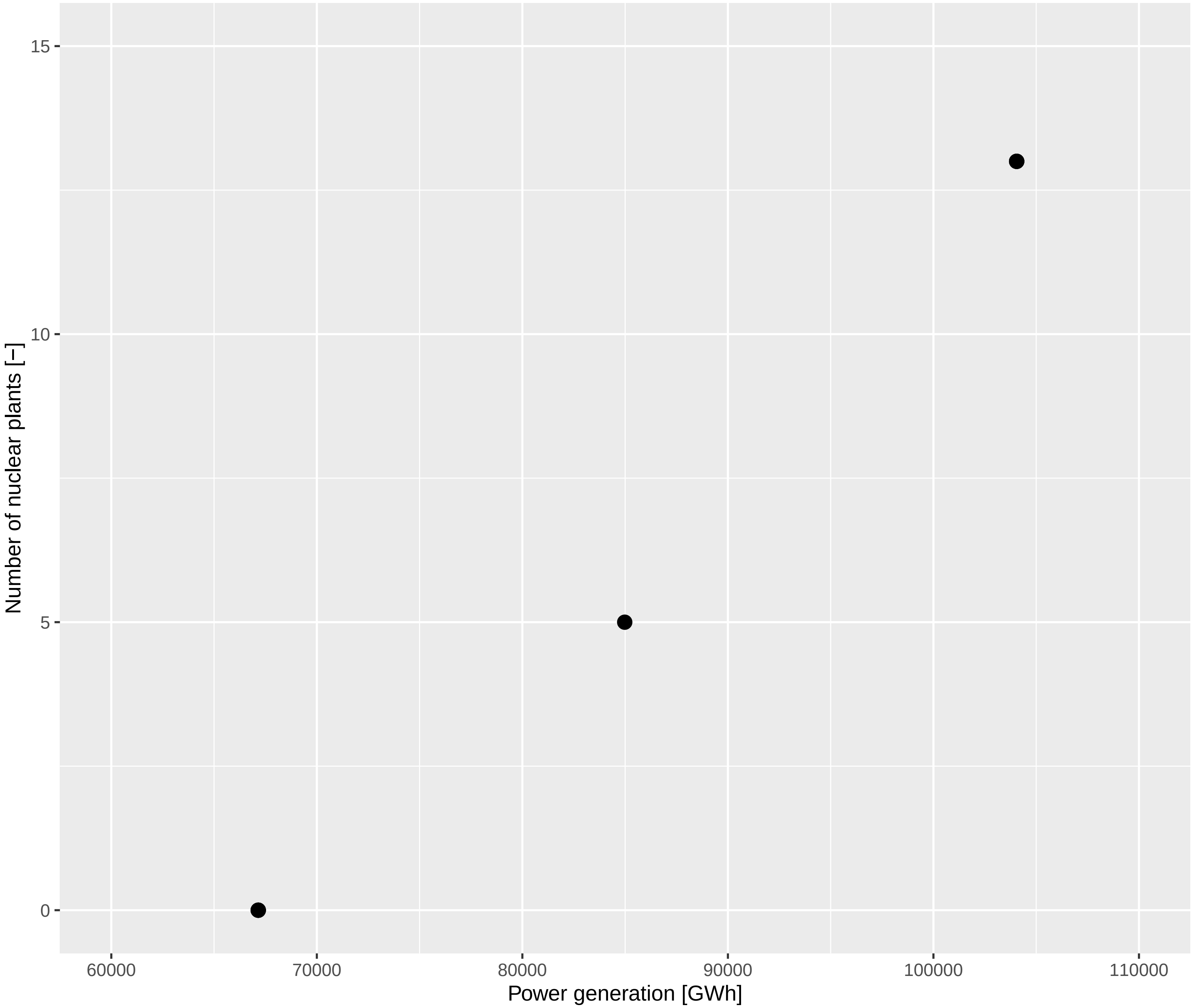

value
1.00
0.75
0.50
0.25
0.00
Nuclear
Coal
CCGT
Wind
Solar
variable
Nuclear
1.00
0.75
0.50
0.25
0.00

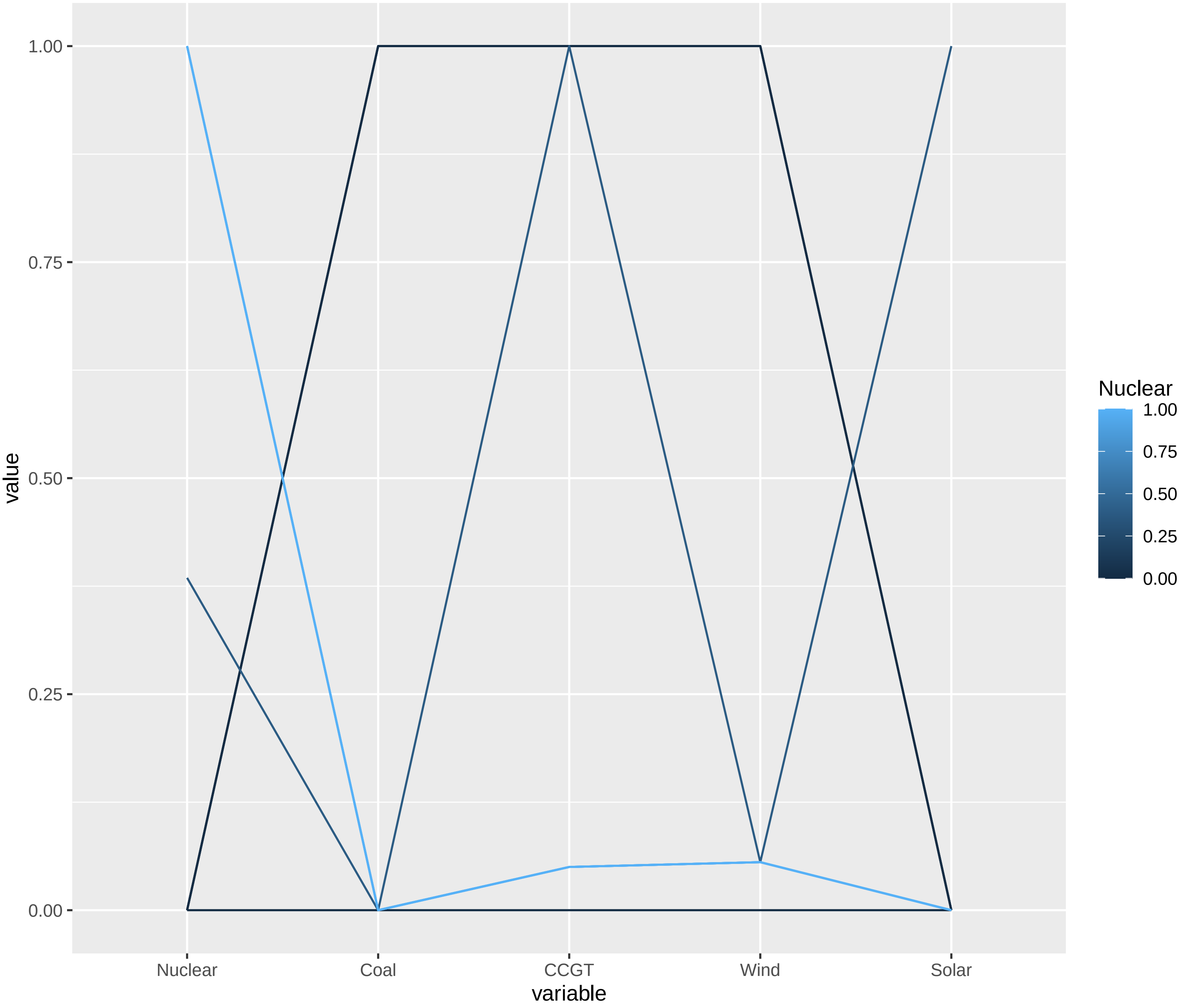
Activity 1c

```
# We apply the weighting method to solve the two-objective problem

# Problem: optimize power generation and number of nuclear plants (by aggregating them in a single objective function)
#
# maximize lambda*(7750 X1 + 3250 X2 + 2250 X3 + 1050 X4 + 90 X5) + (1-lambda)(-X1)
# subject to:
# 248 X1 + 206.375 X2 + 102.9375 X3 + 62.2125 X4 + 18.045 X5 <= 3400 (in million SGD)
# 0 X1 + 2.6 X2 + 0.7875 X3 + 0 X4 + 0 X5 <= 20 (in million tonnes)
# 5.8125 X1 + 30.0625 X2 + 9.5625 X3 + 0.525 X4 + 0.045 X5 <= 250 (in million tonnes)
# X2 <= 20
# X3 <= 20
# X4 <= 30
# X5 <= 100
# X1, X2, X3, X4, X5 integers
# X1, X2, X3, X4, X5 represent nuclear plants, coal plants, CCGT plants, wind turbines, solar cells
# Note: we change the verse of optimization for the second objective
#
# How to handle the two objectives?
# First, we have to normalize them. To simplify the algebraic operations, we simply divide each objective by the range of variability.
# --> lambda*((7750 X1 + 3250 X2 + 2250 X3 + 1050 X4 + 90 X5)/(104,050-67,150)) + (1-lambda)((-X1)/(13-0)) =
#     lambda*((7750 X1 + 3250 X2 + 2250 X3 + 1050 X4 + 90 X5)/(36,900)) + (1-lambda)((-X1)/13) =
# Then, we rewrite the equation in order to make it suitable for the lp function
# --> X1*(lambda*(7750/36900)+lambda/13-1/13) + X2*lambda*(3250/36900) + X3*lambda*(2250/36900) + X4*lambda*(1050/36900) + X5*lambda*(90/36900)
#
# Define inputs for the function lp
f.con_weight <- f.con_1
f.dir_weight <- f.dir_1
f.rhs_weight <- f.rhs_1
```

```
lambda <- c(0:10/10)
#
# We create a for loop to solve all problems for the different values of
lambda and
# we initialise two matrices to store the values of objective functions and
decision variables
objval_weight <- matrix( c(rep(0,2*11)), nrow =11, byrow=TRUE)
decvar_weight <- matrix( c(rep(0,5*11)), nrow =11, byrow=TRUE)
for (i in 1:11){
  # define the coeffcients of the objective function (as a function of
lambda) -- FILL IN THE LINE BELOW
  f.obj_weight <- c(lambda[i] * (7750/36900) + lambda[i]/13-1/13, lambda[i]
* (3250/36900), lambda[i] * (2250/36900), lambda[i] * (1050/36900),
lambda[i] * (90/36900))
  # solve the optimization problem
  result <- lp ("max", f.obj_weight, f.con_weight, f.dir_weight,
f.rhs_weight, all.int=TRUE)
  # save the value of objective functions and decision variables
  objval_weight[i,1] <- sum(result$solution*c(7750, 3250, 2250, 1050, 90))
  objval_weight[i,2] <- result$solution[1]
  decvar_weight[i,]  <- result$solution
}
#
# Finally, we plot the solutions in both objective and decision space
#
# Objective space
c_plot1 <- ggplot(data = data.frame(Power.Generation=objval_weight[,1],
Nuclear.Plants=objval_weight[,2]),
       mapping = aes(x = Power.Generation, y = Nuclear.Plants)) +
  geom_point(size=3) +
  coord_cartesian(xlim=c(60*10^3,110*10^3), ylim=c(0,15)) +
  labs(x="Power generation [GWh]", y="Number of nuclear plants [-]")
c_plot1
ggsave("1c Plot1.pdf", c_plot1)
# The Pareto frontier is not convex, so we cannot find all solutions we
found with the epsilon-constraint method
#

# most upper right data point represents lambda = 1
# ie. giving entire weight to power generation
# most lower left data point represents lambda = 0
# ie. giving entire weight to number of nuclear of power plants
# middle data point some lambda value representingng the trade-off between
the two objectives
# ie. different weights giving same objective value function
# Decision space
# It is difficult to use a scatter plot, because we have 5 decision
variables.
# Let's try with a parallel coordinate plot
c_plot2 <-
ggparcoord(data.frame(Nuclear=decvar_weight[,1],Coal=decvar_weight[,2],CCGT
=decvar_weight[,3], Wind=decvar_weight[,4], Solar=decvar_weight[,5]),
           columns = 1:5, groupColumn = "Nuclear", scale = "uniminmax")
c_plot2
ggsave("1c Plot2.pdf", c_plot2)
```

```
b_plot2 <-
ggparcoord(data.frame(Nuclear=decvar_eps[,1],Coal=decvar_eps[,2],CCGT=decva
r_eps[,3], Wind=decvar_eps[,4], Solar=decvar_eps[,5]),
           columns = 1:5, groupColumn = "Nuclear", scale = "uniminmax")
b_plot2
# The plot shows that the number of CCGT plants and Wind turbines increases
# as the number of nulcear plants is reduced (and viceversa)!
ggsave("1b Plot2.pdf", b_plot2)
```


15
10
5
0
Number of nuclear plants [–]
60000
70000
80000
90000
100000
110000
Power generation [GWh]


value
1.00
0.75
0.50
0.25
0.00
Nuclear
Coal
CCGT
Wind
Solar
variable
Nuclear
1.00
0.75
0.50
0.25
0.00