# load packages

library(sp)

library(rgdal)

library(raster)

library(prioritizr)

library(prioritizrdata)

library(gurobi)

# Load the data into Rstudio

amsterdam <- readOGR(dsn = "C:/Users/krast/Documents/UVA/MFS/WGS/WGS\_files/Amsterdam\_Urban\_Planning.shp")

fs <- list.files(path="C:/Users/krast/Documents/UVA/MFS/WGS/WGS\_files/", pattern = "tif$", full.names = TRUE)

features <- raster::stack(fs)

# Plot data

plot(amsterdam)

plot(features)

# Transform data to numbers and logics

amsterdam$cost <- as.numeric(amsterdam$grid\_code)

amsterdam$locked\_in <- as.logical(as.numeric(amsterdam$PARK))

#Base problem

problem(ams\_map, features, cost\_column = "cost") %>%

add\_max\_utility\_objective(38500) %>%

add\_binary\_decisions() %>%

add\_default\_solver()

#Scenario 1

#budget 38.5 million

#maximize NO2 and distance

#minimize population that has to move

#added manual targets

p1 <- problem(ams\_map, features, cost\_column = "cost") %>%

add\_min\_shortfall\_objective(38500) %>%

add\_manual\_targets(data.frame( feature =

names(features)[1:3], type = "relative",

sense = c(">=", ">=", "<="), target = c(0.02, 0.02, 0.01))) %>%

add\_binary\_decisions() %>%

add\_default\_solver()

s1 <- solve(p1)

print(eval\_cost\_summary(p1, s1[, "solution\_1"]), width = Inf)

print(eval\_target\_coverage\_summary(p1, s1[, "solution\_1"]), width = Inf)

spplot(s1, "solution\_1", main = "Solution Scenario 1", col.regions = c("white", "darkgreen"))

#Scenario 2 - values the size of green spaces, that should be at least 2ha = 2 cells

#adding neighbor constraints

#every green space cell should have at least 1 neighbor in order to complete the 2ha size suggestion

p2 <- p1 %>%

add\_neighbor\_constraints(1)

s2 <- solve(p2)

print(eval\_cost\_summary(p2, s2[, "solution\_1"]), width = Inf)

print(eval\_target\_coverage\_summary(p2, s2[, "solution\_1"]), width = Inf)

spplot(s2, "solution\_1", main = "Solution Scenario 2", col.regions = c("white", "darkgreen"))

#Scenario 3 - see how much of the city can be converted into green spaces by meeting specific targets, without a set budget

#meet all the targets and minimise the cost of the solution

p3 <- problem (ams\_map, features, cost\_column = "cost") %>%

add\_min\_set\_objective() %>%

add\_manual\_targets(data.frame( feature =

names(features)[1:3], type = "relative",

sense = c(">=", ">=", "<="), target = c(0.1, 0.1, 0.5))) %>%

add\_binary\_decisions() %>%

add\_default\_solver()

s3 <- solve(p3)

print(eval\_cost\_summary(p3, s3[, "solution\_1"]), width = Inf)

print(eval\_target\_coverage\_summary(p3, s3[, "solution\_1"]), width = Inf)

spplot(s3, "solution\_1", main = "Solution Scenario 3", col.regions = c("white", "darkgreen"))