Functions in SamplingStrata package - II

Marco Ballin, Giulio Barcaroli

20 May 2019

Procedural steps (continuous stratification variables)

Function *optimizeStrata2* performs the same task than *optimizeStrata*, but with a different Genetic Algorithm, operating on a genome represented by vector containing real values, instead of integers.

This permits to operate directly on the boundaries of the strata, instead of aggregating the initial atomic strata.

In some situations (not exceedingly too big size of the sampling frame) this new function is much more efficient.

A major limitation is in the nature of the stratification variables, that are required to be all continuous (though categorical ordinal can be handled).

Another one is in the fact that it is necessary to choose the number of strata (it is not optimally determined during the evolution process as in the case of *optimizeStrata*).

Procedural steps (continuous stratification variables)

To operate with *optimizeStrata2* it is no more necessary to produce the *strata* dataframe.

We just need to perform the following steps:

- construction of the **sampling frame**
- choice of **precision constraints**
- **optimization**
- **analysis** of results
- **selection of the sample**

Construction of the sampling frame

Let us consider the following example:

```
data(swissmunicipalities)
# Only two regions
swissmunicipalities <- swissmunicipalities[swissmunicipalities$REG < 3,]
# add a unique identifier
swissmunicipalities$id <- c(1:nrow(swissmunicipalities))
swissframe <- buildFrameDF(df = swissmunicipalities,
                           id = "id",
                           X = c("Surfacesbois",
                                 "Surfacescult"),
                           Y = c("Pop020",
                                 "Pop2040"),
                           domainvalue = "REG")
```

Choice of precision constraints

```
swisserrors <- as.data.frame(list(DOM=rep("DOM1",2),
                                  CV1=c(0.08,0.10),
                                  CV2=c(0.12,0.15),
                                  domainvalue=c(1:2)
                            ))
swisserrors
```

```
##    DOM  CV1  CV2 domainvalue
## 1 DOM1 0.08 0.12           1
## 2 DOM1 0.10 0.15           2
```

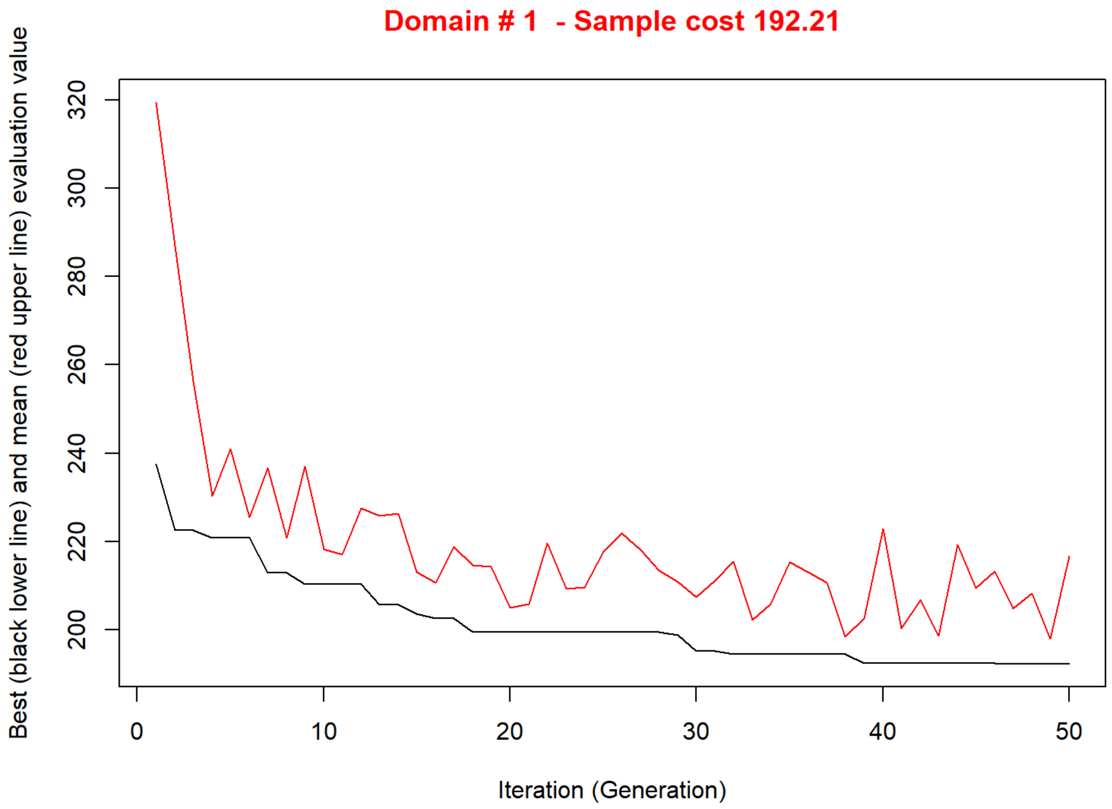
By so doing, we have chosen two stratification variables (*Surfacesbois* and *Surfacescult*) and two target variables (*Pop020* and *Pop2040*), on both of which we have set the same precision constraint (a maximum CV equal to 10%).

Optimization

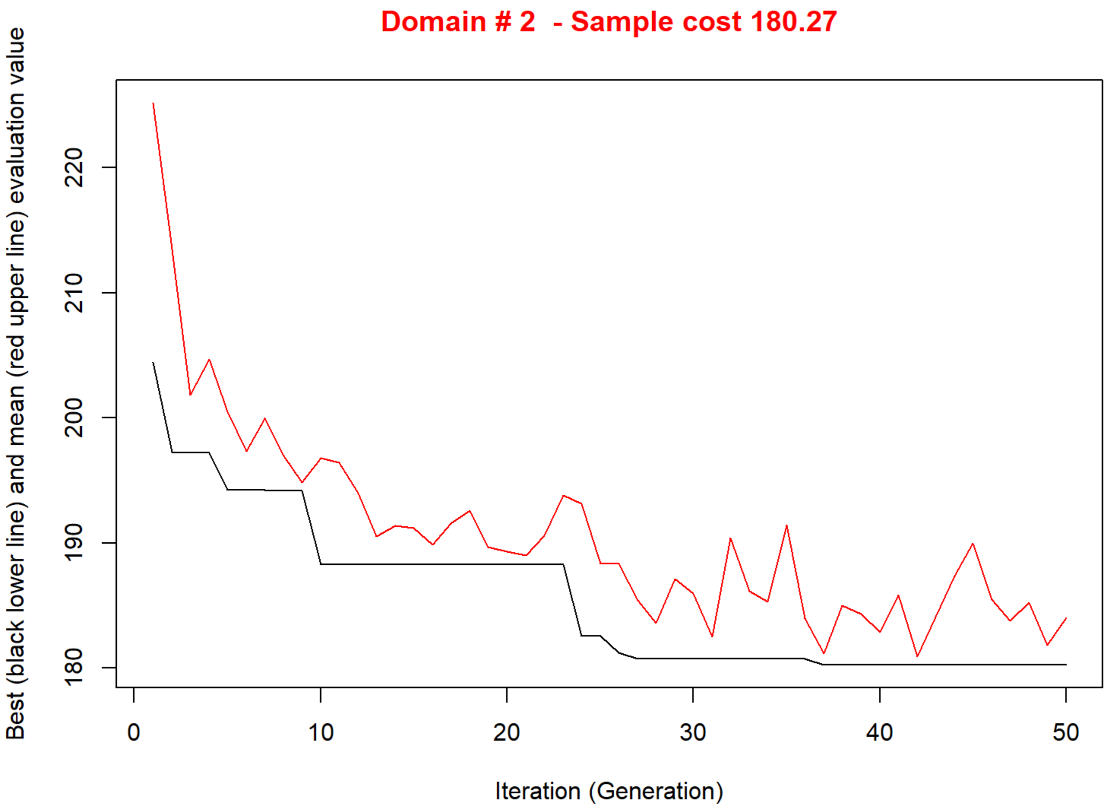
Now the execution of the optimization step (for the domain 4) using *optimizeStrata2* is straightforward, as we do not need to categorize stratification variables and to generate the atomic strata:

```
set.seed(123)
solution2 <- optimizeStrata2 (
  errors = swisserrors,
  framesamp = swissframe,
  iter = 50,
  pops = 10,
  nStrata = 5,
  writeFiles = FALSE,
  showPlot = FALSE,
  parallel = FALSE
)
```

```
##
##  *** Domain :  1   1
##  Number of strata :  589
##  *** Sample cost:  192.2106
##  *** Number of strata:  5
```



```
##
##  *** Domain :  2   2
##  Number of strata :  913
##  *** Sample cost:  180.2669
##  *** Number of strata:  5
```



```
##
##   *** Sample size :  372
##   *** Number of strata :  10
## ---------------------------
```

Optimization

```
sum(round(solution2$aggr_strata$SOLUZ))
```

```
## [1] 372
```

```
expected_CV(solution2$aggr_strata)
```

```
##      cv(Y1) cv(Y2)
## DOM1   0.08  0.076
## DOM2   0.10  0.121
```

This function also outputs the new version of the overall frame, already provided with the labels indicating to which stratum each unit belongs:

```
framenew <- solution2$framenew
table(framenew$DOMAINVALUE,framenew$LABEL)
```

```
##    
##         1   2   3   4   5
##     1  24 144 137 247  37
##     2 385 270 170  51  37
```

Analysis of the results

To inspect the structure of the strata, the function *summaryStrata* is available:

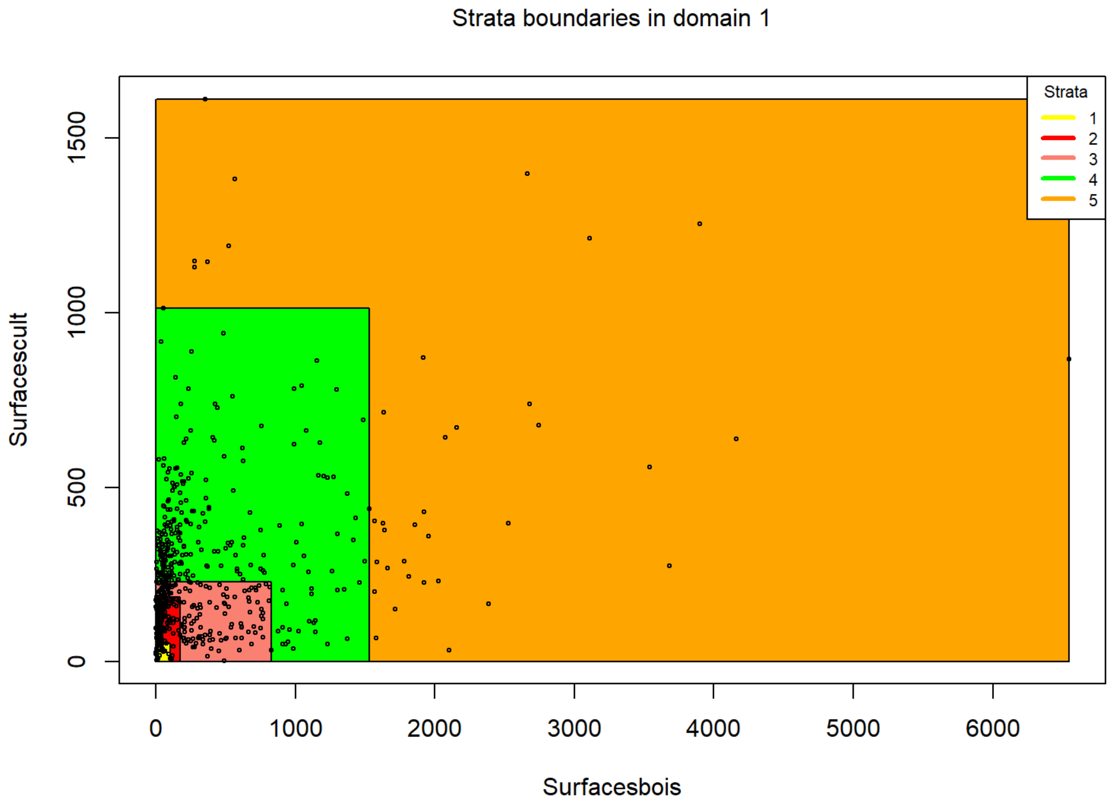
```
strataStructure <- summaryStrata(solution2$framenew,solution2$aggr_strata,progress=FALSE)
strataStructure
```

```
##    Domain Stratum Population Allocation SamplingRate Lower_X1 Upper_X1 Lower_X2 Upper_X2
## 1       1       1         24         24     1.000000        0      105        6       54
## 2       1       2        144         47     0.328727        1      175        9      186
## 3       1       3        137         12     0.085822        6      826        3      230
## 4       1       4        247         72     0.291969        6     1533       37     1013
## 5       1       5         37         37     1.000000      276     6545       34     1613
## 6       2       1        385         34     0.088990        0      141        1      604
## 7       2       2        270         48     0.176803       48      410       10     1022
## 8       2       3        170         54     0.319946      212     1009       48     1145
## 9       2       4         51          8     0.162209      256     1597      165     1767
## 10      2       5         37         36     0.962311      409     4021      192     2621
```

Analysis of the results

It is also possible to visually investigate the distribution of population units in the strata by using the function *plotStrata2d*:

```
outstrata <- plotStrata2d(
                    solution2$framenew, 
                    solution2$aggr_strata,
                    domain = 1, 
                    vars = c("X1","X2"),
                    labels =     c("Surfacesbois","Surfacescult")
                    )
```



Selection of the sample

Finally, the selection of the sample is performed by using the same function selectSample, but this time is no more necessary to handle separately units to be sampled and units to be censused:

```
samp <- selectSample(solution2$framenew,solution2$aggr_strata)
```

```
##
## *** Sample has been drawn successfully ***
##  372  units have been selected from  10  strata
##
## ==> There have been  2  take-all strata
## from which have been selected  61 units
```