

Estudo de Caso 03: Comparação de desempenho de duas configurações de um algoritmo de otimização

Diego Pontes, Elias Vieira, Matheus Bitarães

Março, 2021

Descrição do problema

Introdução

Design do Experimento

```
# Importação dos pacotes (descomentar caso não tenha instalado)
# install.packages("ExpDE")
# install.packages("smoof")
```

Definição da função de Rosenbrock para uma dada dimensão *dim* (código abaixo copiado da descrição do problema)

```
suppressPackageStartupMessages(library(smoof))
fn <- function(X){
  if(!is.matrix(X)) X <- matrix(X, nrow = 1) # <- if a single vector is passed as X
  Y <- apply(X, MARGIN = 1,
    FUN = smoof::makeRosenbrockFunction(dimensions = dim))
  return(Y)
}
```

?(código abaixo copiado da descrição do problema)

```
# EXEMPLO
dim <- 10
selpars <- list(name = "selection_standard")
stopcrit <- list(names = "stop_maxeval", maxevals = 5000 * dim, maxiter = 100 * dim)
probpars <- list(name = "fn", xmin = rep(-5, dim), xmax = rep(10, dim))
popsize = 5 * dim
```

Este artigo é importante: <https://link.springer.com/article/10.1007%2Fs10732-018-9396-7>

Configurações que devem ser comparadas:

```
# Equipe C
## Config 1
```

```
recpars1 <- list(name = "recombination_blxAlphaBeta", alpha = 0, beta = 0)
mutpars1 <- list(name = "mutation_rand", f = 4)
```

```
## Config 2
```

```
recpars2 <- list(name = "recombination_linear")
mutpars2 <- list(name = "mutation_rand", f = 1.5)
```

```
suppressPackageStartupMessages(library(ExpDE))
```

```
# Run algorithm on problem:
#out <- ExpDE(mutpars = mutparsX,
#recpars = recparsX,
#popsize = popsize,
#selpars = selpars,
#stopcrit = stopcrit,
#probpars = probpars,
#showpars = list(show.its = "dots", showevery = 20))
```

```
# Extract observation:
#out$Fbest
```

- Necessario calcular o numero de instancias necessarias e o numero de repetições de cada algoritmo por instancia para que consigamos chegar na significancia e potencia desejadas

Calculo do numero de instancias (artigo <https://link.springer.com/article/10.1007%2Fs10732-018-9396-7>)

```
# dados do trabalho
alpha <- 0.05
delta <- 0.5
beta <- 0.2

n <- 2
while (qt(1 - alpha/2, n-1) > qt(beta, n - 1, delta*sqrt(n))) n <- n + 1

print(n)
```

```
## [1] 34
```

34 é o numero minimo de blocos para a comparação dos dois algoritmos.

falta: - coleta e tabulação dos dados - testes das hipoteses - estimacao da magnitude da diferença entre os metodos - verificação das premissas dos testes - conclusoes - possiveis limitações do estudo e sugestoes de melhoria

Atividades dos membros

Diego

Elias

Matheus

Todos