

# Estudo de Caso 3: Planejamento e Análise de Experimentos

*Matheus Marzochi, Mayra Mota, Rafael Ramos e Victor Magalhães*

*11 de Novembro de 2019*

## Resumo

O objetivo deste estudo de caso é investigar como modificações de hiperparâmetros de um algoritmo de otimização baseado em evolução diferencial influenciam em seu desempenho, em diferentes cenários de execução. Os algoritmos serão testados partir de funções de Rosenbrock, de dimensões entre 2 e 150.

## Papéis Desempenhados

A divisão de tarefas no grupo segue a descrição da *Declaração de Políticas de Equipe*. Estando aqui organizada da seguinte forma:

- Matheus: Verificador
- Mayra: Monitora
- Rafael: Coordenador
- Victor: Revisor

## Planejamento do Experimento

### Descrição do Problema

O problema analisado tem por objetivo comparar o desempenho de duas configurações de um algoritmo de otimização baseada em evolução diferencial, na resolução do problema de ótimo em funções de Rosenbrock. As funções podem possuir dimensões que variam entre 2 e 150. A hipótese nula é a de que o desempenho dos dois algoritmos permanece a mesma independente da configuração, e a hipótese que está sendo testada é a de que existe uma diferença de desempenho entre elas.

Para avaliar o desempenho de cada um dos algoritmos, seus desempenhos sob diferentes dimensões da função de Rosenbrock são levados em consideração, a partir de testes pareados. Testes pareados constituem uma parte de testes blocados, onde distribuições são comparadas caso a caso, agrupadas em blocos, com o objetivo de diminuir efeitos que não estejam relacionados com os parâmetros em teste. O valor de performance  $y$  para cada algoritmo  $i$  em cada instância  $j$  é dado a partir da fórmula:

$$y_{ij} = \mu + \tau_i + \beta_j + \epsilon_{ij}$$

onde  $\mu$  corresponde à média geral de todas as amostras,  $\tau_i$  corresponde o efeito do  $i$ -ésimo algoritmo,  $\beta_j$  representa , e  $\epsilon_{ij}$  representa um erro randômico com média nula e independentemente distribuídos, de variância  $\sigma^2$ .

O que queremos testar é a equivalência dos parâmetros sob teste para cada algoritmo, ou seja,  $y_i$ . Seja  $y_i$  definido como:

$$y_i = \sum_{j=1}^n y_{ij} \overline{y_i} = \frac{y_i}{n}$$

onde  $n$  equivale ao total de instâncias (blocos). A hipótese nula é:

$$H_0 : y_1 = y_2 = \dots = y_a$$

onde a equivale ao total de algoritmos sendo comparados, o que no problema em questão são 2. Como o que difere cada parâmetro  $y_i$  é o valor da influência do algoritmo  $i$ , dizer que a hipótese nula é que os parâmetros  $y_i$  são iguais equivale a dizer que

$$H_0 : \tau_1 = \tau_2 = \dots = \tau_a = 0$$

o que indica não haver influência do algoritmo nos parâmetros das execuções, e a definição do mesmo passa a ser portanto apenas a média global acrescida de um erro  $\epsilon_{ij}$ . Isto equivale a dizer que todas as observações foram retiradas de uma distribuição normal com média  $\mu$  e variância  $\sigma^2$ . A hipótese em teste, é:

$$H_1 : \tau_i = 0$$

para pelo menos algum valor de  $i$ .

### Execução do Experimento

Como deseja-se avaliar o desempenho das duas configurações de algoritmos sob diferentes dimensões de problema, serão realizados testes pareados. Cada observação do teste corresponde a uma dimensão do problema de Rosenbrook. Considerando as métricas para o teste estabelecidas anteriormente, o número de instâncias para o teste pareado deverá ser, no mínimo, 34 amostras, como o cálculo a seguir demonstra.

```
result <- power.t.test(delta=0.5,
  sig.level=0.05,
  power=0.8,
  type='paired',
  alternative='two.sided')
print(result)

##
##      Paired t test power calculation
##
##              n = 33.3672
##              delta = 0.5
##              sd = 1
##      sig.level = 0.05
##              power = 0.8
##      alternative = two.sided
##
## NOTE: n is number of *pairs*, sd is std.dev. of *differences* within pairs
```

### Coleta dos Dados

```
n_instancias <- 34

dimensions <- linspace(2, 150, n_instancias)

out_dim.conf1 = data.frame(matrix(ncol = 2, nrow = 0))
colnames(out_dim.conf1) <- c("dim", "best")
out_dim.conf2 = data.frame(matrix(ncol = 2, nrow = 0))
colnames(out_dim.conf2) <- c("dim", "best")
```

```

count.dim <- 1
for (d in dimensions){

  dim <- ceil(d)

  cat("\nBuilding dimension ", dim)

  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 = smooof::makeRosenbrockFunction(dimensions = dim))
    return(Y)
  }
  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

  ## Config 1
  recpars1 <- list(name = "recombination_arith")
  mutpars1 <- list(name = "mutation_rand", f = 4)
  ## Config 2
  recpars2 <- list(name = "recombination_bin", cr = 0.7)
  mutpars2 <- list(name = "mutation_best", f = 3)

  out <- ExpDE(mutpars = mutpars1,
               recpars = recpars1,
               popsize = popsize,
               selpars = selpars,
               stopcrit = stopcrit,
               probpars = probpars,
               showpars = list(show.iters = "dots", showevery = 20))
  de <- list("dim"=dim, "best"=out$Fbest)
  out_dim.conf1 <- rbind(out_dim.conf1,de, stringsAsFactors=FALSE)

  out <- ExpDE(mutpars = mutpars2,
               recpars = recpars2,
               popsize = popsize,
               selpars = selpars,
               stopcrit = stopcrit,
               probpars = probpars,
               showpars = list(show.iters = "dots", showevery = 20))
  de <- list("dim"=dim, "best"=out$Fbest)
  out_dim.conf2 <- rbind(out_dim.conf1,de, stringsAsFactors=FALSE)

  count.dim <- count.dim + 1
}

write.csv(out_dim.conf1, 'conf_1.csv', row.names=FALSE)
write.csv(out_dim.conf2, 'conf_2.csv', row.names=FALSE)

```

**Análise Exploratória dos Dados**

**Análise dos Dados Experimentais Femininos**

**Teste de Hipótese**

**Discussão para Melhoria do Experimento**

**Conclusões**

**Referências**