

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

André Boechat (Checker), Augusto (Checker), Mateus Pongelupe(Coordinator), Samuel Leite(Recorder)

30 de Outubro de 2018

Resumo

Este relatório é o terceiro dos Estudos de Casos na disciplina de Planejamento e Análise de Experimentos. O problema inicial consiste em duas variações do mesmo algoritmo avaliando a mesma população. Deseja-se analisar o efeito dessa variação afim de verificar o desempenho das alterações no algoritmo.

O método da Evolução Diferencial (DE) é uma técnica que procura solucionar problemas de otimização contínua. É um método simples, porém poderoso, que avalia soluções de forma iterativa, buscando otimização global. Ele é dotado de 3 passos, sendo ele: mutação, cruzamento e seleção. No primeiro, é feito a soma de um indivíduo base com a diferença de dois outros indivíduos aleatórios multiplicados por uma ponderação F . O vetor mutado é chamado de V . O cruzamento é uma intercalagem entre o vetor original e o vetor mutado, de forma que aumenta-se a variabilidade da sua população. O vetor intercalado é chamado de U . A seleção avalia qual o melhor indivíduo a ser escolhido, seja ele da população original ou da população cruzada.

Planejamento do Experimento

Será realizado um experimento, nos quais parâmetros diferentes são submetidos ao realizar o método DE. Para o teste de hipóteses, avaliada a diferença entre as médias, denominada pelo parâmetro μ_d .

$$\begin{cases} H_0 : \mu_D = 0 \\ H_1 : \mu_D <> 0 \end{cases}$$

Para esse teste, definiu-se um nível de significância de $\alpha = 0.05$, uma mínima diferença de importância prática $d^* = \delta^*/\sigma = 0.5$ e uma potência desejada de $\pi = 1 - \beta = 0.8$.

```
alpha = 0.05
d = 0.5
beta = 0.2
power = 1 - beta
```

O tamanho da amostra é calculado utilizando-se dos parâmetros de teste requeridos e a função `power.t.test`.

```
n <- power.t.test(n=NULL,
                  delta = d,
                  sd = 1,
                  sig.level = alpha,
                  power = power,
                  type="paired",
                  alternative="two.sided")$n

N <- ceiling(n)
cat("N: ", N)
```

```
## N: 34
```

Coleta de Dados

Para geração dos dados foi criada a rotina **generateAllSamples** que recebe o número de execuções de cada algoritmo por amostra e o número de amostras N calculado para o problema. Nessa função, são selecionadas aleatoriamente as dimensões das funções de Rosenbrock a serem avaliadas e os resultados são escritos em arquivos no formato *.csv*. Dessa forma, essa função é executada apenas uma vez para um dado tamanho amostral. O valor para o número de avaliações por software será setado no valor padrão de 30.

```
library("smoof")

suppressPackageStartupMessages(library(smoof))

getSamples <- function(n,dim) {
  # This assingment makes fn a global variable
  # This function is created dinamically for each dimension
  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)
  }

  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_lbga")
  mutpars1 <- list(name = "mutation_rand", f = 4.5)

  ## Config 2
  recpars2 <- list(name = "recombination_blxAlphaBeta", alpha = 0.1, beta = 0.4)
  mutpars2 <- list(name = "mutation_rand", f = 3)

  library("ExpDE")

  suppressPackageStartupMessages(library(ExpDE))

  # Extracted one execution of the DE algorithm
  generate_sample <- function(mutpars, recpars, popsize, selpars, stopcrit, probpars) {
    return(ExpDE(mutpars = mutpars,
                  recpars = recpars,
                  popsize = popsize,
                  selpars = selpars,
                  stopcrit = stopcrit,
                  probpars = probpars,
                  showpars = list(show.iters = "dots", showevery = 20))$Fbest);
  }

  # Executes n times a given configuration of the DE algorithm
  generate_n_samples <- function(n, mutpars, recpars) {
    return(replicate(n, generate_sample(mutpars, recpars, popsize, selpars, stopcrit, probpars)))
  }
```

```

# Runs n times both DE algorithms
return(data.frame("A1" = generate_n_samples(n, mutpars1, recpars1), "A2" = generate_n_samples(n, mutp
})

generateAllSamples <- function(n_samples, N) {
  rosenbrok_dim_interval <- 2:150
  dimensions <- sort(sample(rosenbrok_dim_interval, N))

  result <- lapply(dimensions, FUN = function(x) {
    write.csv(getSamples(n_samples, x), paste('samples-dim-', x, '.csv', sep = ''), row.names=FALSE)
  })
  write.csv(data.frame("dim" = dimensions), 'dimensions.csv', row.names=FALSE)
}

dimensionsFile <- 'dimensions.csv'
nTestsPerSample <- 30

# if dimension files does not exist, gets all samples
if(!file.exists(dimensionsFile)) {
  generateAllSamples(nTestsPerSample, N)
}

```

Como os dados foram gerados apenas uma vez, devido ao tempo requerido para tal, o processamento dos dados foi feito a partir da leitura dos arquivos *.csv* produzidos. Assim, todos os dados salvos para experimento, estão presentes na variável *dataFrame*.

```

dimensions <- read.csv(file = dimensionsFile, header = T)
dataFrame <- data.frame('A1' = character(), 'A2' = character(), 'dim' = character(), stringsAsFactors =
lapply(dimensions$dim, FUN = function(x) {
  fileName <- paste('samples-dim-', x, '.csv', sep = '')
  data <- read.csv(file = fileName, header = T)
  dataFrame <-- rbind(dataFrame, data)
})

```

Análise Estatística

Teste da Média

Dados os parâmetros definidos na seção *Planejamento do Experimento* para o teste, foram analisadas $N = 34$ amostras e o teste foi executado nas linhas abaixo. Foram utilizados valores diferentes para a análise DE, de acordo com o fornecido para o Grupo E.

Foi analisado se há alguma parametrização recomendada para um melhor desempenho, considerando que o algoritmo comparado seria de otimização. Também foi verificado a diferença em amplitude nos resultados encontrados.

O tamanho da amostra a considerar nesta situação refere-se ao número de instâncias de problemas, e não necessariamente ao número de medidas repetidas dentro de problemas.

Observa-se que o método de pareamento, permite eliminar problemas de ruído (distúrbio) entre as amostras, uma vez que a análise ocorre sobre a diferença entre elas, portanto o distúrbio em relação a um ponto médio, para dados gerados por uma mesma distribuição aleatória, tendem a ser parecidos.

```

matriz.media <-matrix(nrow=35, ncol=3)
matriz.media[1,1]<-"A1"
matriz.media[1,2]<-"A2"
matriz.media[1,3]<-"Dim"
matriz.media

dim<-read.csv("samples-dim-2.csv")

matriz.media[2,1]<-mean(dim$A1)
matriz.media[2,2]<-mean(dim$A2)
matriz.media[2,3]<-"2"

dim<-read.csv("samples-dim-5.csv")

matriz.media[3,1]<-mean(dim$A1)
matriz.media[3,2]<-mean(dim$A2)
matriz.media[3,3]<-"5"

dim<-read.csv("samples-dim-15.csv")

matriz.media[4,1]<-mean(dim$A1)
matriz.media[4,2]<-mean(dim$A2)
matriz.media[4,3]<-"15"

dim<-read.csv("samples-dim-21.csv")

matriz.media[5,1]<-mean(dim$A1)
matriz.media[5,2]<-mean(dim$A2)
matriz.media[5,3]<-"21"

dim<-read.csv("samples-dim-24.csv")

matriz.media[6,1]<-mean(dim$A1)
matriz.media[6,2]<-mean(dim$A2)
matriz.media[6,3]<-"24"

dim<-read.csv("samples-dim-26.csv")

matriz.media[7,1]<-mean(dim$A1)
matriz.media[7,2]<-mean(dim$A2)
matriz.media[7,3]<-"26"

dim<-read.csv("samples-dim-46.csv")

matriz.media[8,1]<-mean(dim$A1)
matriz.media[8,2]<-mean(dim$A2)
matriz.media[8,3]<-"46"

dim<-read.csv("samples-dim-50.csv")

matriz.media[9,1]<-mean(dim$A1)
matriz.media[9,2]<-mean(dim$A2)
matriz.media[9,3]<-"50"

```

```

dim<-read.csv("samples-dim-54.csv")

matriz.media[10,1]<-mean(dim$A1)
matriz.media[10,2]<-mean(dim$A2)
matriz.media[10,3]<-"54"

dim<-read.csv("samples-dim-55.csv")

matriz.media[11,1]<-mean(dim$A1)
matriz.media[11,2]<-mean(dim$A2)
matriz.media[11,3]<-"55"

dim<-read.csv("samples-dim-56.csv")

matriz.media[12,1]<-mean(dim$A1)
matriz.media[12,2]<-mean(dim$A2)
matriz.media[12,3]<-"56"

dim<-read.csv("samples-dim-57.csv")

matriz.media[13,1]<-mean(dim$A1)
matriz.media[13,2]<-mean(dim$A2)
matriz.media[13,3]<-"57"

dim<-read.csv("samples-dim-58.csv")

matriz.media[14,1]<-mean(dim$A1)
matriz.media[14,2]<-mean(dim$A2)
matriz.media[14,3]<-"58"

dim<-read.csv("samples-dim-59.csv")

matriz.media[15,1]<-mean(dim$A1)
matriz.media[15,2]<-mean(dim$A2)
matriz.media[15,3]<-"59"

dim<-read.csv("samples-dim-64.csv")

matriz.media[16,1]<-mean(dim$A1)
matriz.media[16,2]<-mean(dim$A2)
matriz.media[16,3]<-"64"

dim<-read.csv("samples-dim-66.csv")

matriz.media[17,1]<-mean(dim$A1)
matriz.media[17,2]<-mean(dim$A2)
matriz.media[17,3]<-"66"

dim<-read.csv("samples-dim-69.csv")

matriz.media[18,1]<-mean(dim$A1)
matriz.media[18,2]<-mean(dim$A2)
matriz.media[18,3]<-"69"

```

```

dim<-read.csv("samples-dim-70.csv")

matriz.media[19,1]<-mean(dim$A1)
matriz.media[19,2]<-mean(dim$A2)
matriz.media[19,3]<-"70"

dim<-read.csv("samples-dim-73.csv")

matriz.media[20,1]<-mean(dim$A1)
matriz.media[20,2]<-mean(dim$A2)
matriz.media[20,3]<-"73"

dim<-read.csv("samples-dim-78.csv")

matriz.media[21,1]<-mean(dim$A1)
matriz.media[21,2]<-mean(dim$A2)
matriz.media[21,3]<-"78"

dim<-read.csv("samples-dim-79.csv")

matriz.media[22,1]<-mean(dim$A1)
matriz.media[22,2]<-mean(dim$A2)
matriz.media[22,3]<-"79"

dim<-read.csv("samples-dim-82.csv")

matriz.media[23,1]<-mean(dim$A1)
matriz.media[23,2]<-mean(dim$A2)
matriz.media[23,3]<-"82"

dim<-read.csv("samples-dim-83.csv")

matriz.media[24,1]<-mean(dim$A1)
matriz.media[24,2]<-mean(dim$A2)
matriz.media[24,3]<-"83"

dim<-read.csv("samples-dim-91.csv")

matriz.media[25,1]<-mean(dim$A1)
matriz.media[25,2]<-mean(dim$A2)
matriz.media[25,3]<-"91"

dim<-read.csv("samples-dim-92.csv")

matriz.media[26,1]<-mean(dim$A1)
matriz.media[26,2]<-mean(dim$A2)
matriz.media[26,3]<-"92"

dim<-read.csv("samples-dim-94.csv")

matriz.media[27,1]<-mean(dim$A1)
matriz.media[27,2]<-mean(dim$A2)
matriz.media[27,3]<-"94"

```

```

dim<-read.csv("samples-dim-97.csv")

matriz.media[28,1]<-mean(dim$A1)
matriz.media[28,2]<-mean(dim$A2)
matriz.media[28,3]<-"97"

dim<-read.csv("samples-dim-113.csv")

matriz.media[29,1]<-mean(dim$A1)
matriz.media[29,2]<-mean(dim$A2)
matriz.media[29,3]<-"113"

dim<-read.csv("samples-dim-117.csv")

matriz.media[30,1]<-mean(dim$A1)
matriz.media[30,2]<-mean(dim$A2)
matriz.media[30,3]<-"117"

dim<-read.csv("samples-dim-128.csv")

matriz.media[31,1]<-mean(dim$A1)
matriz.media[31,2]<-mean(dim$A2)
matriz.media[31,3]<-"128"

dim<-read.csv("samples-dim-131.csv")

matriz.media[32,1]<-mean(dim$A1)
matriz.media[32,2]<-mean(dim$A2)
matriz.media[32,3]<-"131"

dim<-read.csv("samples-dim-136.csv")

matriz.media[33,1]<-mean(dim$A1)
matriz.media[33,2]<-mean(dim$A2)
matriz.media[33,3]<-"136"

dim<-read.csv("samples-dim-145.csv")

matriz.media[34,1]<-mean(dim$A1)
matriz.media[34,2]<-mean(dim$A2)
matriz.media[34,3]<-"145"

dim<-read.csv("samples-dim-149.csv")

matriz.media[35,1]<-mean(dim$A1)
matriz.media[35,2]<-mean(dim$A2)
matriz.media[35,3]<-"149"

write.table(matriz.media, file="media.csv", row.names=FALSE, col.names = FALSE, sep=",")

medias <- read.csv("media.csv", sep=",", dec=".")
t.test(medias$A1, medias$A2, paired = TRUE, conf.level = alpha, alternative = "two.sided")

```

```
##
## Paired t-test
##
## data: medias$A1 and medias$A2
## t = -13.475, df = 33, p-value = 5.724e-15
## alternative hypothesis: true difference in means is not equal to 0
## 5 percent confidence interval:
## -1829291 -1812216
## sample estimates:
## mean of the differences
## -1820753
```

O histograma abaixo mostra a comparação das médias calculadas pelos algoritmos.

```
#Plot de Diferencas
Diferenca = medias$A1 - medias$A2
plot(Diferenca, pch = 34, ylab="Diferença Algoritmo A1-A2")
abline(0,0, col="blue", lwd=2)
```

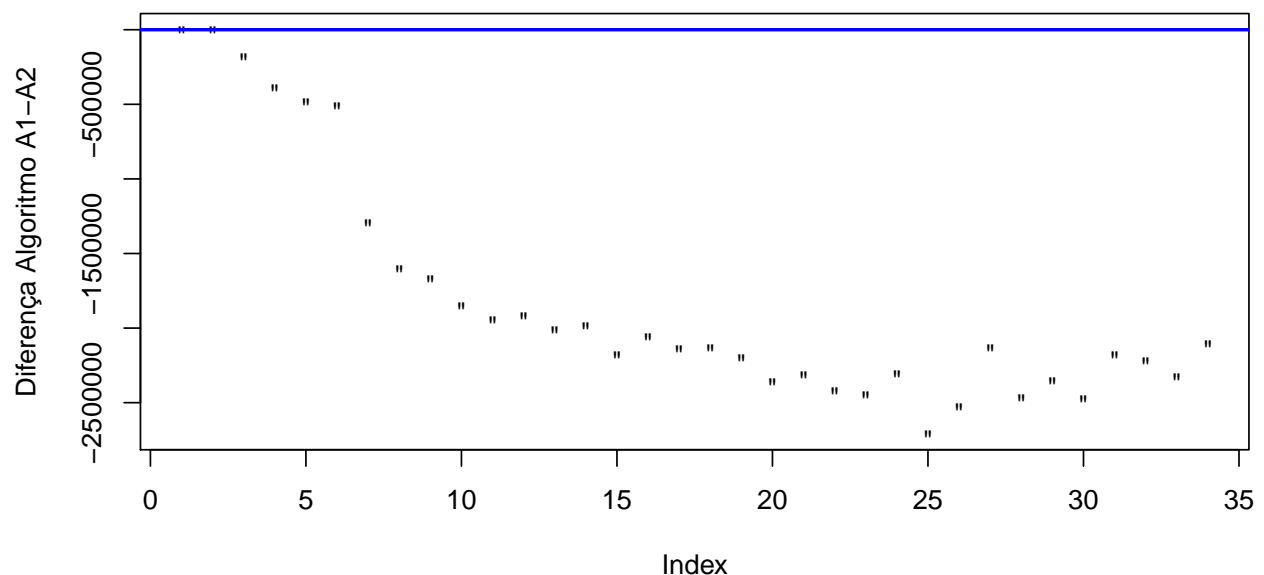


Figura 1: Histograma da diferença de performance

```
plot(medias$A1, medias$A2, pch=34, xlab="Algoritmo A1", ylab="Algoritmo A2")
abline(0,1,col="blue", lwd=2)

hist(Diferenca, col="gray",main="Histograma da diferença de performance", xlab="Diferença")
```

Conclusões e Recomendações

O teste realizado neste relatório serviu para avaliar o desempenho de dois algoritmos diferentes avaliando uma mesma população. O método da Evolução Diferencial é um método de otimização, mas como ele apresenta parâmetros numéricos de entrada, deve também ter esses otimizados. Este relatório teve como objetivo analisar dois conjuntos parâmetros para realizar o método de convergência DE para um mesmo conjunto de dados gerados e observar qual obteve uma maior otimização.

Esse método é fortemente recomendado, quando há uma correlação forte entre as amostras, casos por exemplo de condições experimentais heterogeneas).

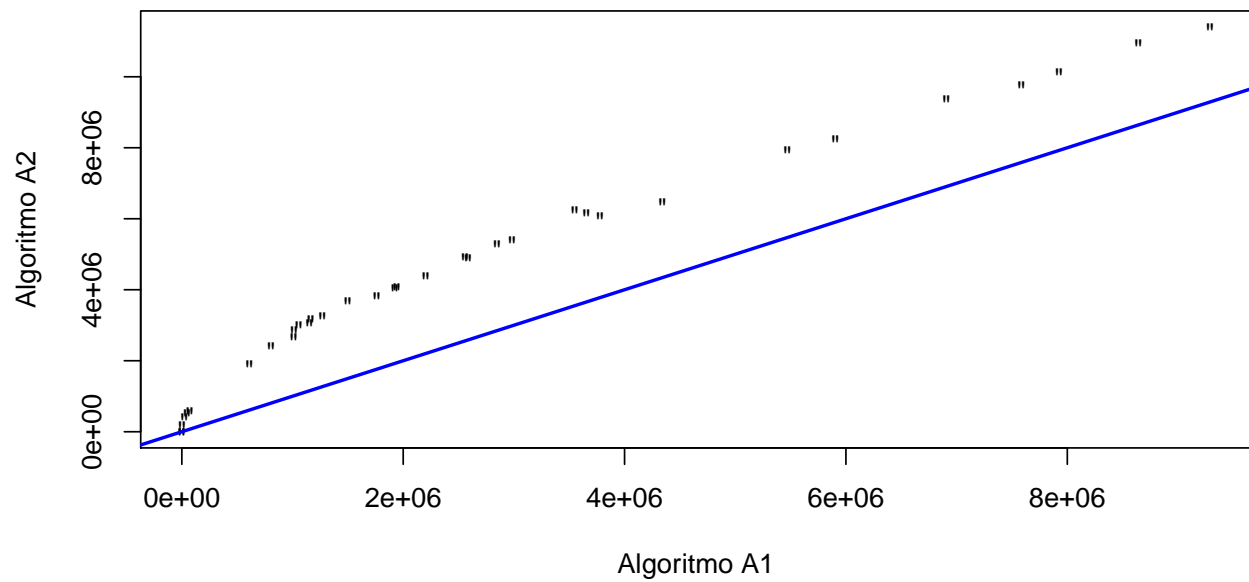


Figura 2: Histograma da diferença de performance

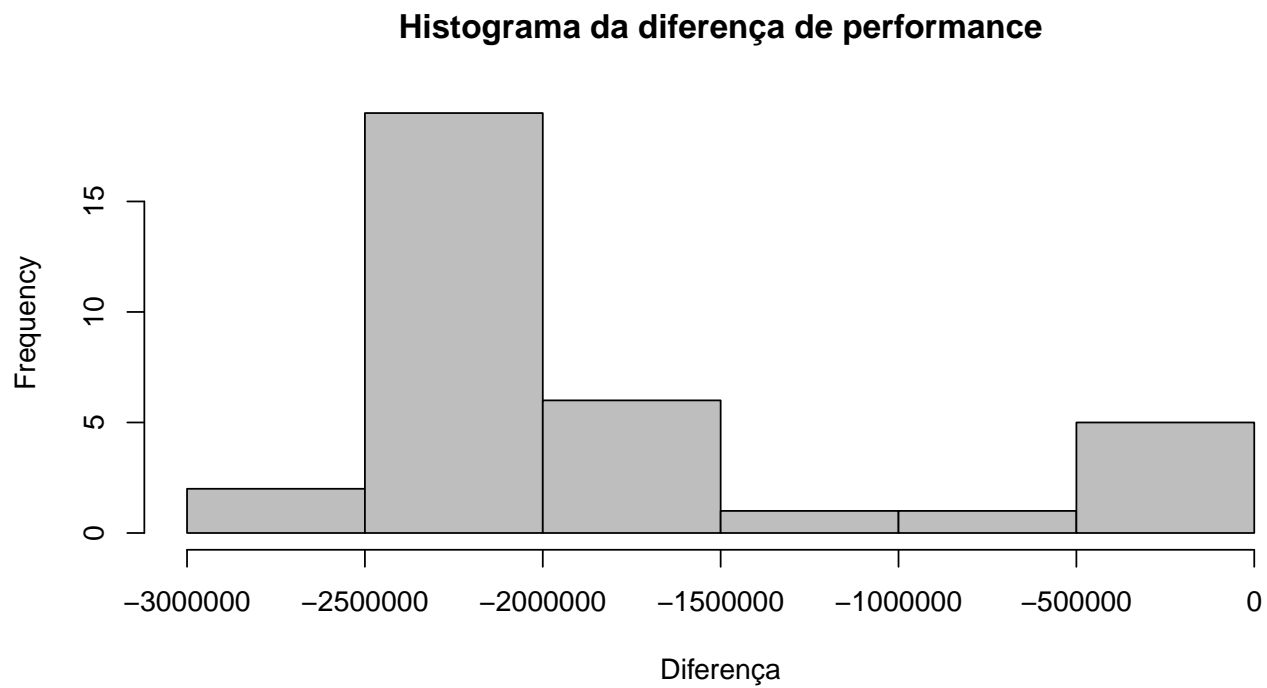


Figura 3: Histograma da diferença de performance

Através dos testes realizados, é possível perceber que ambas as configurações utilizadas no método DE apresentaram resultados que, à partir do Teste T de médias, não são similares. À partir dos gráficos é possível perceber que há uma tendência de redução dos valores calculados à medida que há um aumento da dimensão testada. Portanto, as alterações foram efetivas.

O tempo computacional para realizar a geração de dados e o processamento no “DE” foram um impecilho que geraram um certo atraso na realização do teste. Tornou-se assim inviável propor novos parâmetros para realizar a otimização.

Referências

- Statistics R Tutorial - <https://www.cyclismo.org/tutorial/R/confidence.html>
- Montgomery, Douglas C. - Applied statistics and probabiliy for engineers (3rd Edition) - Cap?tulos 8,9
- Notas de Aula - <https://github.com/fcampelo/Design-and-Analysis-of-Experiments>
- Notas - https://edisciplinas.usp.br/pluginfile.php/2063723/mod_resource/content/0/Aula11-2016.pdf