

# EEE933 - Estudo de Caso 03

Felipe Campelo<sup>a</sup>

<sup>a</sup>Programa de Pós-Graduação em Engenharia Elétrica, UFMG.

This version was compiled on October 24, 2017

## Comparação de desempenho de configurações de um algoritmo de otimização

**Apresentação.** Algoritmos baseados em populações são uma alternativa comum para a solução de problemas de otimização em engenharia. Tais algoritmos normalmente consistem de um ciclo iterativo, no qual um conjunto de soluções-candidatas ao problema são repetidamente sujeitas a operadores de variação e seleção, de forma a promover uma exploração do espaço de variáveis do problema em busca de um ponto de ótimo (máximo ou mínimo) de uma dada função-objetivo.

Dentre estes algoritmos, um método que tem sido bastante utilizado nos últimos anos é conhecido como *evolução diferencial* (DE, do inglês *differential evolution*) (Storn and Price, 1997). De forma simplificada, este método é composto pelos seguintes passos:

0. Entrada:  $N$ ,  $n_{iter}$ ,  $recpars$ ,  $mutpars$ 
  1.  $t \leftarrow 0$
  2.  $X_t \leftarrow \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$  (população inicial)
  3.  $\vec{f}_t \leftarrow f(X_t)$
  4. Enquanto ( $t < n_{iter}$ )
    1.  $V_t \leftarrow \text{mutação}(X_t, mutpars)$
    2.  $U_t \leftarrow \text{recombinação}(X_t, V_t, recpars)$
    3.  $\vec{j}_t \leftarrow f(U_t)$
    4.  $(X_{t+1}, \vec{f}_{t+1}) \leftarrow \text{seleção}(X_t, U_t, \vec{f}_t, \vec{j}_t)$
    5.  $t \leftarrow t + 1$
5. Saída:  $(X_t, \vec{f}_t)$

Suponha que um pesquisador está interessado em investigar o efeito de diferentes configurações deste algoritmo em seu desempenho para um dado problema. Para isso, o mesmo implementou o algoritmo e diversos operadores de forma padronizada, na forma do pacote ExpDE (<https://cran.r-project.org/web/packages/ExpDE/>) (Campelo and Botelho, 2016), cuja versão de desenvolvimento encontra-se disponível no Github (<https://github.com/fcampelo/ExpDE/tree/devel>).

**Atividades.** Como forma de análise preliminar deste problema, cada equipe terá como tarefa a comparação experimental de algumas configurações em um único problema de teste. O objetivo deste estudo é responder às perguntas:

*Há alguma diferença no desempenho médio do algoritmo quando equipado com estas diferentes configurações, para o problema de teste utilizado? Caso haja, qual a melhor configuração em termos de desempenho médio (atenção: quanto menor o valor retornado, melhor o algoritmo), e qual a magnitude das diferenças encontradas? Há alguma configuração que deva ser recomendada em relação aos demais?*

Os seguintes parâmetros experimentais são dados para este estudo:

- Mínima diferença de importância prática entre qualquer par de algoritmos (padronizada, em termos do coeficiente  $d$  de Cohen):  $(d^* = \delta^* / \sigma) = 0.25$
- Significância desejada:  $\alpha = 0.05$
- Potência mínima desejada (para o caso  $d = d^*$ ):  $\pi = 1 - \beta = 0.85$

**Informações operacionais.** Para instalar e carregar o pacote no computador, os seguintes comandos podem ser utilizados:

```
install.packages("ExpDE")  
library(ExpDE)
```

Os parâmetros fixos do algoritmo (ao longo de toda a experimentação) são dados por:

```
selpars <- list(name = "selection_standard")  
stopcrit <- list(names = "stop_maxeval", maxevals = 50000, maxiter = 1000)  
probpars <- list(name = "sphere", xmin = -seq(1,20), xmax = 20 + 5 * seq(5, 24))
```

As configurações que deverão ser comparadas por cada grupo passaram por uma etapa anterior de ajuste de parâmetros. Suas definições são dadas por:

```
# Equipe Iuri, Henrique, Rodrigo  
  
## Config 1  
recpars1 <- list(name = "recombination_arith")  
mutpars1 <- list(name = "mutation_rand", f = 4)  
popsize1 <- 300  
  
## Config 2  
recpars2 <- list(name = "recombination_bin", cr = 0.7)  
mutpars2 <- list(name = "mutation_best", f = 3)  
popsize2 <- 300  
  
## Config 3  
recpars3 <- list(name = "recombination_blxAlphaBeta", alpha = 0.4, beta = 0.4)  
mutpars3 <- list(name = "mutation_rand", f = 4)  
popsize3 <- 230  
  
## Config 4  
recpars4 <- list(name = "recombination_eigen",  
                othertype = "recombination_bin", cr = 0.9)  
mutpars4 <- list(name = "mutation_best", f = 2.8)  
popsize4 <- 85
```

*# Equipe Hugo, Lucas, Rafael, Victor*

## Config 1

```
recpars1 <- list(name = "recombination_exp", cr = 0.6)
```

```
mutpars1 <- list(name = "mutation_best", f = 2)
```

```
popsizel <- 130
```

## Config 2

```
recpars2 <- list(name = "recombination_geo", alpha = 0.6)
```

```
mutpars2 <- list(name = "mutation_rand", f = 1.2)
```

```
popsizel <- 70
```

## Config 3

```
recpars3 <- list(name = "recombination_lbga")
```

```
mutpars3 <- list(name = "mutation_rand", f = 4.5)
```

```
popsizel <- 300
```

## Config 4

```
recpars4 <- list(name = "recombination_blxAlphaBeta", alpha = 0.1, beta = 0.4)
```

```
mutpars4 <- list(name = "mutation_rand", f = 3)
```

```
popsizel <- 80
```

*# Equipe Monica, Leticia Mayra, Mateus*

## Config 1

```
recpars1 <- list(name = "recombination_blxAlphaBeta", alpha = 0, beta = 0)
```

```
mutpars1 <- list(name = "mutation_rand", f = 4)
```

```
popsizel <- 200
```

## Config 2

```
recpars2 <- list(name = "recombination_linear")
```

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

```
popsizel <- 250
```

## Config 3

```
recpars3 <- list(name = "recombination_mmax", lambda = 0.25)
```

```
mutpars3 <- list(name = "mutation_best", f = 4)
```

```
popsizel <- 375
```

## Config 4

```
recpars4 <- list(name = "recombination_npoint", N = 17)
```

```
mutpars4 <- list(name = "mutation_rand", f = 2.2)
```

```
popsizel <- 225
```

*# Equipe Ezequiel, Lucas, Ricardo*

```
## Config 1
recpars1 <- list(name = "recombination_onepoint", K = 17)
mutpars1 <- list(name = "mutation_best", f = 2.4)
popsize1 <- 225

## Config 2
recpars2 <- list(name = "recombination_pbest", cr = 0.25)
mutpars2 <- list(name = "mutation_rand", f = 3.5)
popsize2 <- 325

## Config 3
recpars3 <- list(name = "recombination_sbx", eta = 90)
mutpars3 <- list(name = "mutation_best", f = 4.5)
popsize3 <- 200

## Config 4
recpars4 <- list(name = "recombination_wright")
mutpars4 <- list(name = "mutation_best", f = 4.8)
popsize4 <- 113
```

*# Equipe Eduardo, Guilherme, Paulo, Franklin*

```
## Config 1
recpars1 <- list(name = "recombination_eigen",
                 othername = "recombination_bin", cr = 0.9)
mutpars1 <- list(name = "mutation_best", f = 2.8)
popsize1 <- 85

## Config 2
recpars2 <- list(name = "recombination_sbx", eta = 90)
mutpars2 <- list(name = "mutation_best", f = 4.5)
popsize2 <- 200

## Config 3
recpars3 <- list(name = "recombination_linear")
mutpars3 <- list(name = "mutation_rand", f = 1.5)
popsize3 <- 250

## Config 4
recpars4 <- list(name = "recombination_blxAlphaBeta", alpha = 0.1, beta = 0.4)
mutpars4 <- list(name = "mutation_rand", f = 3)
popsize4 <- 80
```

*# Equipe Rafael, Marcus, Paulo, Elany*

```
## Config 1
recpars1 <- list(name = "recombination_onepoint", K = 17)
mutpars1 <- list(name = "mutation_best", f = 2.4)
popsize1 <- 225

## Config 2
recpars2 <- list(name = "recombination_mmax", lambda = 0.25)
mutpars2 <- list(name = "mutation_best", f = 4)
popsize2 <- 375

## Config 3
recpars3 <- list(name = "recombination_pbest", cr = 0.25)
mutpars3 <- list(name = "mutation_rand", f = 3.5)
popsize3 <- 325

## Config 4
recpars4 <- list(name = "recombination_geo", alpha = 0.6)
mutpars4 <- list(name = "mutation_rand", f = 1.2)
popsize4 <- 70
```

*# Equipe Marcelo, Brayan, William, André*

```
## Config 1
recpars1 <- list(name = "recombination_blxAlphaBeta", alpha = 0, beta = 0)
mutpars1 <- list(name = "mutation_rand", f = 4)
popsize1 <- 200

## Config 2
recpars2 <- list(name = "recombination_exp", cr = 0.6)
mutpars2 <- list(name = "mutation_best", f = 2)
popsize2 <- 130

## Config 3
recpars3 <- list(name = "recombination_blxAlphaBeta", alpha = 0.4, beta = 0.4)
mutpars3 <- list(name = "mutation_rand", f = 4)
popsize3 <- 230

## Config 4
recpars4 <- list(name = "recombination_wright")
mutpars4 <- list(name = "mutation_best", f = 4.8)
popsize4 <- 113
```

```

# Equipe Letícia Resende, Marcelo, André

## Config 1
recpars1 <- list(name = "recombination_eigen",
                 othername = "recombination_bin", cr = 0.9)
mutpars1 <- list(name = "mutation_best", f = 2.8)
popsize1 <- 85

## Config 2
recpars2 <- list(name = "recombination_sbx", eta = 90)
mutpars2 <- list(name = "mutation_best", f = 4.5)
popsize2 <- 200

## Config 3
recpars3 <- list(name = "recombination_blxAlphaBeta", alpha = 0.4, beta = 0.4)
mutpars3 <- list(name = "mutation_rand", f = 4)
popsize3 <- 230

## Config 4
recpars4 <- list(name = "recombination_geo", alpha = 0.6)
mutpars4 <- list(name = "mutation_rand", f = 1.2)
popsize4 <- 70

```

Cada observação individual do desempenho do algoritmo com um dado operador pode ser obtida através dos comandos abaixo:

```

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

# Extract observation:
out$Fbest

```

onde *popsizeX*, *mutparsX* a *recparsX* devem ser substituídos pelas variáveis apropriadas (e.g., *popsize1*, *popsize2* etc.).

**Outras definições.** Este estudo de caso consiste das seguintes etapas:

1. Formulação das hipóteses de teste;
2. Cálculo do tamanho amostral;
3. Coleta e tabulação dos dados;
4. Teste das hipóteses;
5. Estimação dos tamanhos de efeito e dos intervalos de confiança;
6. Verificação das premissas dos testes;

7. Derivação de conclusões;
8. Discussão sobre possíveis limitações do estudo e sugestões de melhoria.

Lembre-se que as conclusões devem ser colocadas no contexto das perguntas técnicas de interesse.

**Relatório.** Cada equipe deverá entregar um relatório detalhando o experimento e a análise dos dados. O relatório será avaliado de acordo com os seguintes critérios:

- Obediência ao formato determinado (ver abaixo);
- Reproducibilidade dos resultados;
- Qualidade técnica;
- Estrutura da argumentação;
- Correto uso da linguagem (gramática, ortografia, etc.);

O relatório deve *obrigatoriamente* ser produzido utilizando [R Markdown](#) (opcionalmente utilizando estilos distintos, como o do presente documento), e deve conter todo o código necessário para a reprodução da análise obtida, embutido na forma de blocos de código no documento. Os grupos devem enviar:

- O arquivo **.Rmd** para geração do relatório.
- O arquivo **.pdf** compilado a partir do **.Rmd**.
- O arquivo de dados utilizado, em formato **.csv**.

O arquivo **.Rmd** deve ser capaz de ser compilado em um pdf sem erros, e deve assumir que o arquivo de dados se encontra no mesmo diretório do arquivo do relatório. Modelos de estudos de caso estão disponíveis no repositório da disciplina no github. Caso a equipe deseje utilizar o estilo do presente documento, pode consultar seu código-fonte no repositório (note que o mesmo requer a instalação do pacote *pinp*).

**Importante:** Salve seu arquivo **.Rmd** em UTF-8 (para evitar erros na compilação em outros sistemas).  
**Importante:** Inclua no relatório os papéis desempenhados por cada membro da equipe (Relator, Verificador etc.)  
Relatórios serão aceitos em português, inglês ou espanhol.

**Entrega.** Os arquivos deverão ser enviados via *e-mail* para o endereço [fcampelo@ufmg.br](mailto:fcampelo@ufmg.br). O título do e-mail deve seguir o padrão “[EEE933\_2017-2\_EC03] Nome\_da\_equipe” (sem as aspas). A data-limite para o recebimento dos arquivos é **terça-feira (07/11) às 11:00h**

## References

- Campelo F, Botelho M (2016). “Experimental Investigation of Recombination Operators for Differential Evolution.” In *Proc. Genetic and Evolutionary Computation Conference - GECCO'2016*.
- Storn R, Price K (1997). “Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces.” *J. of Global Optimization*, **11**(4), 341–359.