Estudo 3

Jonatan Almeida e Helbert Paulino

2023-11-16

Resumo

Este estudo de caso tem por objetivo realizar comparações estatísticas entre dois algoritmos de otimização, baseados em evolução diferencial (algoritmo de Storn and Prince, 1997). Basicamente, a análise estatística tem como objetivo responder às seguintes perguntas:

> Há alguma diferença no desempenho médio do algoritmo quando equipado com estas diferentes configurações, para a classe de problemas de interesse? Caso haja, qual a melhor configuração em termos de desempenho médio, e qual a magnitude das diferenças encontradas? Há alguma configuração que deva ser recomendada em relação à outra?

Para um algoritmo de otimização, quanto menor o valor retornado (tempo de convergência), melhor o algoritmo. Esses parâmetros serão suficientes para determinar se houve ou não alguma melhoria de um algoritmo em relação ao outro, independentemente das entradas do algoritmo.

Análise exploratória dos dados

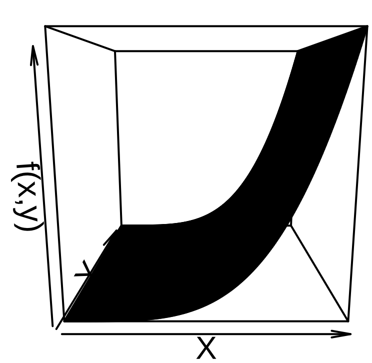
A análise será feita em funções de Rosenbrock (Rosenbrock, 1960), que é comumente utilizada para avaliar o desempenho de algoritmos de otimização. Essas funções são do tipo não convexa, unimodal e garantem convergência para um mínimo global (algumas vezes, esse mínimo global pode ser de difícil convergência (Picheny et al., 2012)). Uma função de Rosenbrock é definida como:

$$f(x, y) = (a - x)^2 + b(x - y^2)^2$$

Em que há um mínimo global em $f(a, a^2) = 0$

A figura a seguir ilustra um exemplo desse tipo de função:

Example of Rosenbrock function



Na análise, serão avaliadas algumas classes de funções de Rosenbrock, de dimensões de 2 a 150. Para isso, um algoritmo de Pohlheim (Pohlheim, 2005) será utilizado para gerar uma função de dimentão dim.

Se considerarmos o modelo estatístico baseado no efeito do fator experimental, RCBD (randomized complete block design), então temos o seguinte modelo:

$$y_{ij} = \mu + \tau_i + \beta_j + \epsilon_{ij} \begin{cases} i = 1, ..., a \\ j = 1, ..., b \\ \epsilon_{ij} \in N(0, \sigma^2) \end{cases}$$

em que i = 1,. . . ,a (número de níveis) e j = 1,. . . ,n (número de observações). Em que μ é a média global, τ_i é o efeito do nível i, β_j é o efeito para o bloco j e ϵ_{ij} é o resíduo.

Dessa forma, a pergunta de interesse nos leva a definir as seguintes hipoteses:

$$\begin{cases} H_0 : \tau_i = 0, \forall i \in [1, a] \\ H_1 : \exists \tau_i \neq 0 \end{cases}$$

Assim, devemos realizar testes para sabermos se há efeitos de nível significativamente diferentes de zero, permitindo a rejeição da hipótese nula. Para isso, foi definido um grupo de instâncias homogêneas de valor 30 e foram aplicadas 64 replicações de testes.

Análise Estatística

Para poder determinar qual dos dois algoritmos seria o melhor, utilizaremos as seguintes métricas:

- Mínima diferença de importância prática (padronizada): (d = δ / σ) = 0.5
- Significância desejada: $\alpha = 0.05$
- Potência mínima desejada para o caso: 1 - β = 0.8

Para as funções de Rosenbrock de dimensão dim, usaremos as variáveis x dentro do intervalo [-5, 10]. Com isso, teremos as seguintes dimensões:

```
##  [1]  86   7 137  18  94 126  42  76 132  51 116  56  64 148  34  55 106 113  39
## [20] 134 111  41  17  46 130  81 149  84  10 120  73 127 124  95   3  83  68   4
## [39]   2  88  49 144 119  80 136  19   9  37 117  38  99  22  78  72 138  45  21
## [58]  29  69 122  48 141  71 135
```

Para o problema proposto, iremos analisar as seguintes configurações parar o problema de dimensões:

Configuração 1:

```r
recpars1 <- list(name = "recombination_exp", cr = 0.6)
mutpars1 <- list(name = "mutation_best", f = 2)
```

Configuração 2:

```r
recpars2 <- list(name = "recombination_geo", alpha = 0.6)
mutpars2 <- list(name = "mutation_rand", f = 1.2)
```

Dessa forma, executamos o algoritmo de otimização nas diferentes instâncias, obtendo o arquivo data.csv com o resultado da execução dos dados. A partir dele, faremos a análise estatística.

Análise dos resultados

A partir dos dados gerados, temos os tempos de execução e, com isso, um resumo sobre eles:

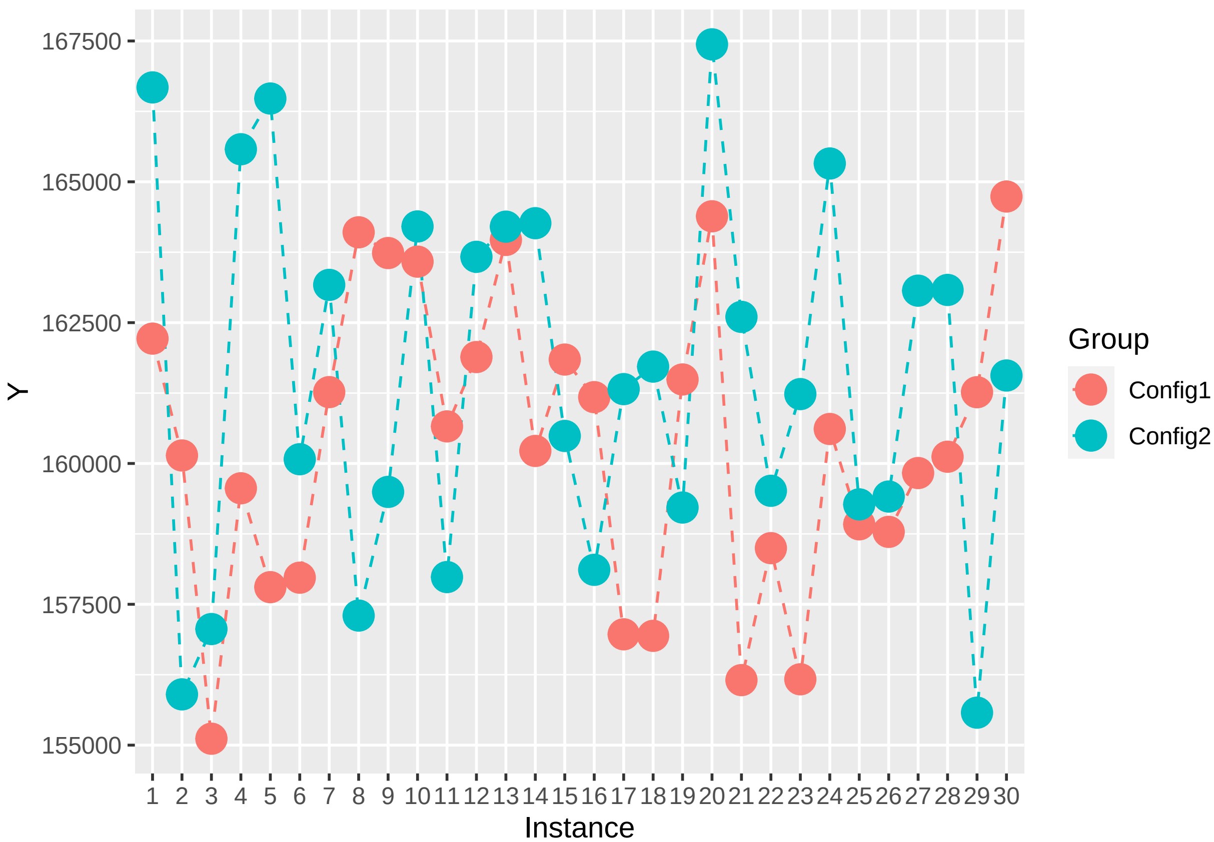
```
##    replicate        instance        group          result       
##  Min.   : 1.00   Min.   : 1.0   Min.   :1.0   Min.   :     0.0  
##  1st Qu.:16.75   1st Qu.: 8.0   1st Qu.:1.0   1st Qu.:   653.5  
##  Median :32.50   Median :15.5   Median :1.5   Median : 29102.9  
##  Mean   :32.50   Mean   :15.5   Mean   :1.5   Mean   :160919.0  
```

```
##  3rd Qu.:48.25   3rd Qu.:23.0   3rd Qu.:2.0   3rd Qu.:307640.7
##  Max.   :64.00   Max.   :30.0   Max.   :2.0   Max.   :832577.4
```

Para cada instância homogênea, calculamos a média dos tempos de execução. Os resultados são:

```
##      Group       Instance        Y
##  Config1:30   1      : 2   Min.   :155114
##  Config2:30   2      : 2   1st Qu.:158886
##               3      : 2   Median :160918
##               4      : 2   Mean   :160919
##               5      : 2   3rd Qu.:163273
##               6      : 2   Max.   :167440
##               (Other):48
```

A partir disso, realizamos alguns plots sobre os dados, para comparação:



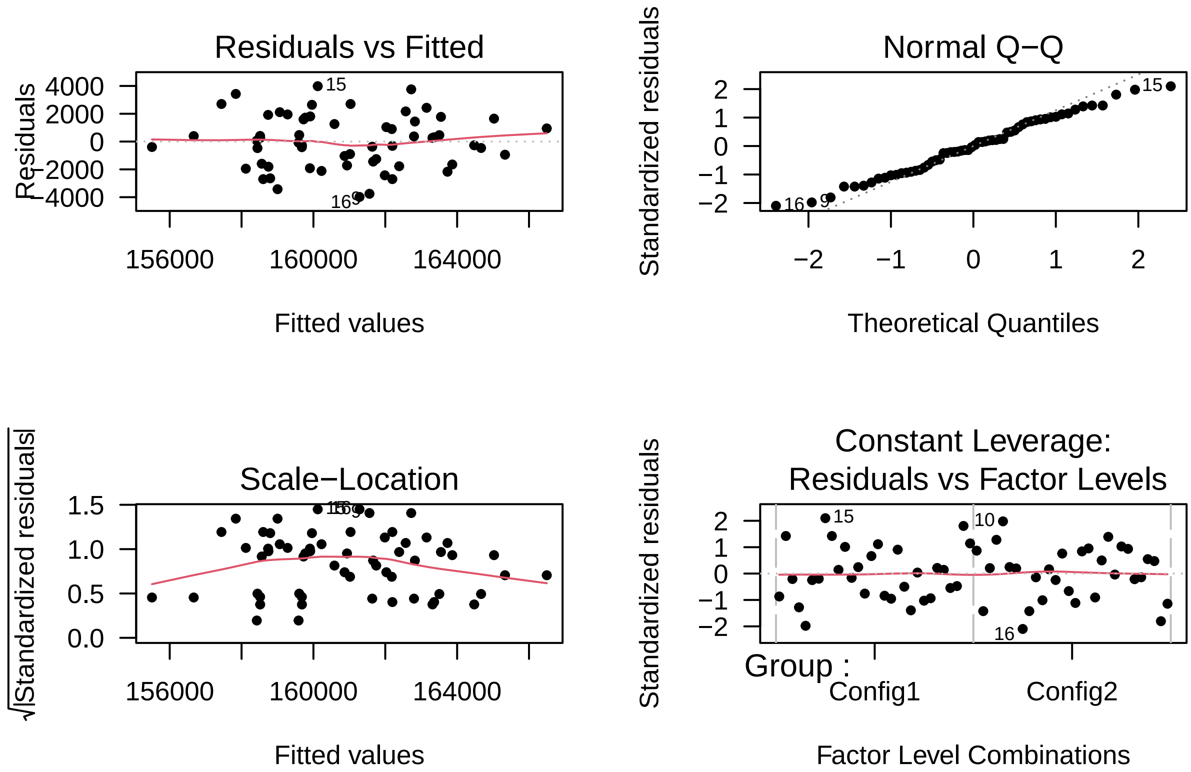
Para a análise da variabilidade do das observações, temos:

```
##             Df    Sum Sq  Mean Sq F value Pr(>F)
## Group        1  20267416 20267416   2.720  0.110
## Instance    29 294499249 10155147   1.363  0.205
## Residuals   29 216106158  7451936
```

```
## [1] 0.5929229
```

A partir dela, podemos ver que os efeitos ao longo das instâncias (efeito horizontal β), são menos significativos que os efeitos verticais (τ), ou seja, há uma variação maior entre os algoritmos do que entre as instâncias. O teste R^2 obtido mostra um valor de 0.59 aproximadamente, o que faz com que o modelo adotado não explique bem toda a variação que ocorre nas variáveis independentes.

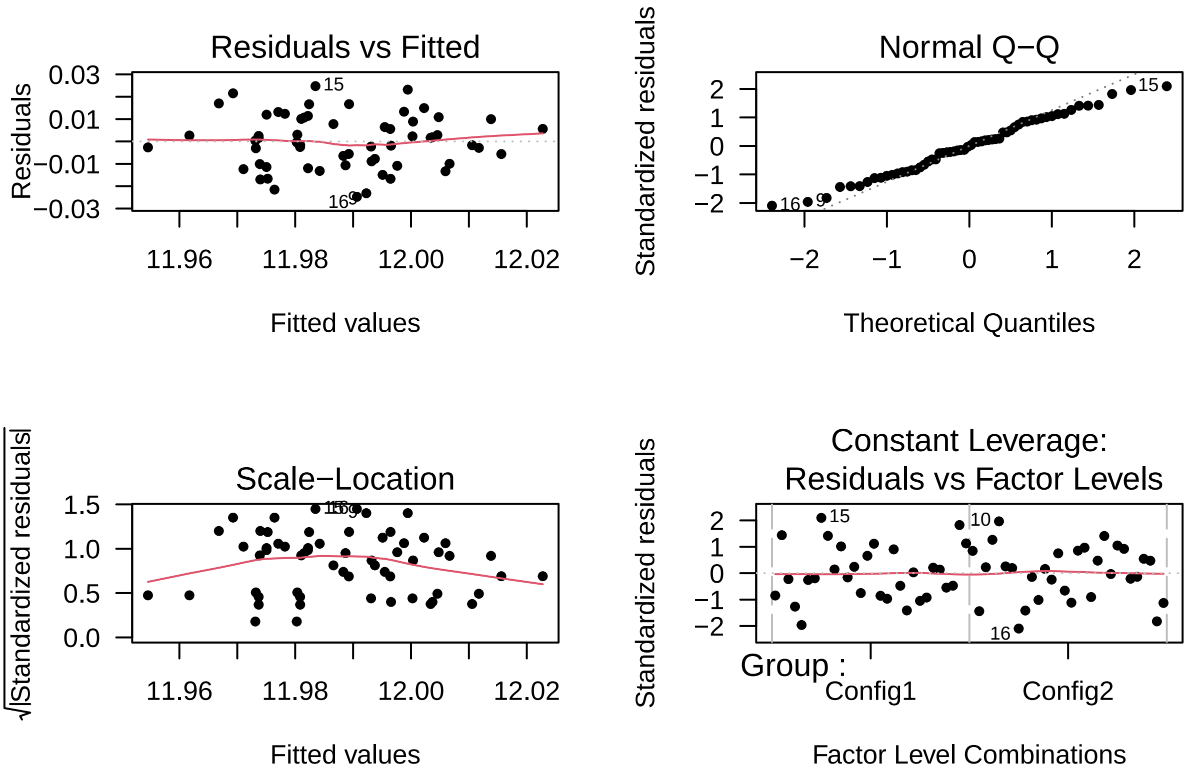
Os seguintes plots mostram a respeito do resíduos:



Como forma complementar de análise, podemos analisar os resíduos em escala logaritimica:

```
##              Df   Sum Sq   Mean Sq F value Pr(>F)
## Group         1 0.000769 0.0007688   2.665  0.113
## Instance     29 0.011353 0.0003915   1.357  0.208
## Residuals    29 0.008364 0.0002884
```

```
## [1] 0.591709
```

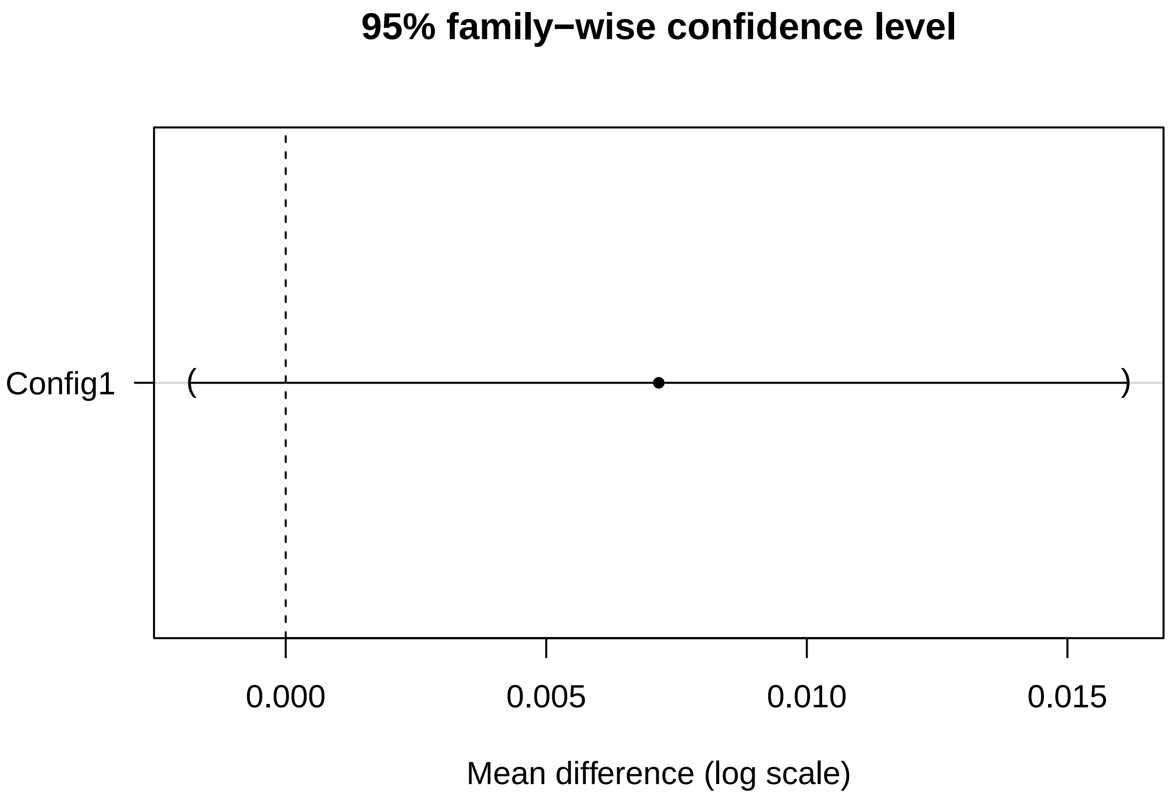


Como se pode perceber, não houve uma melhora significativa nos dados, o que pode ser um indicativo de que um modelo mais apromorado pode ser aplicado.

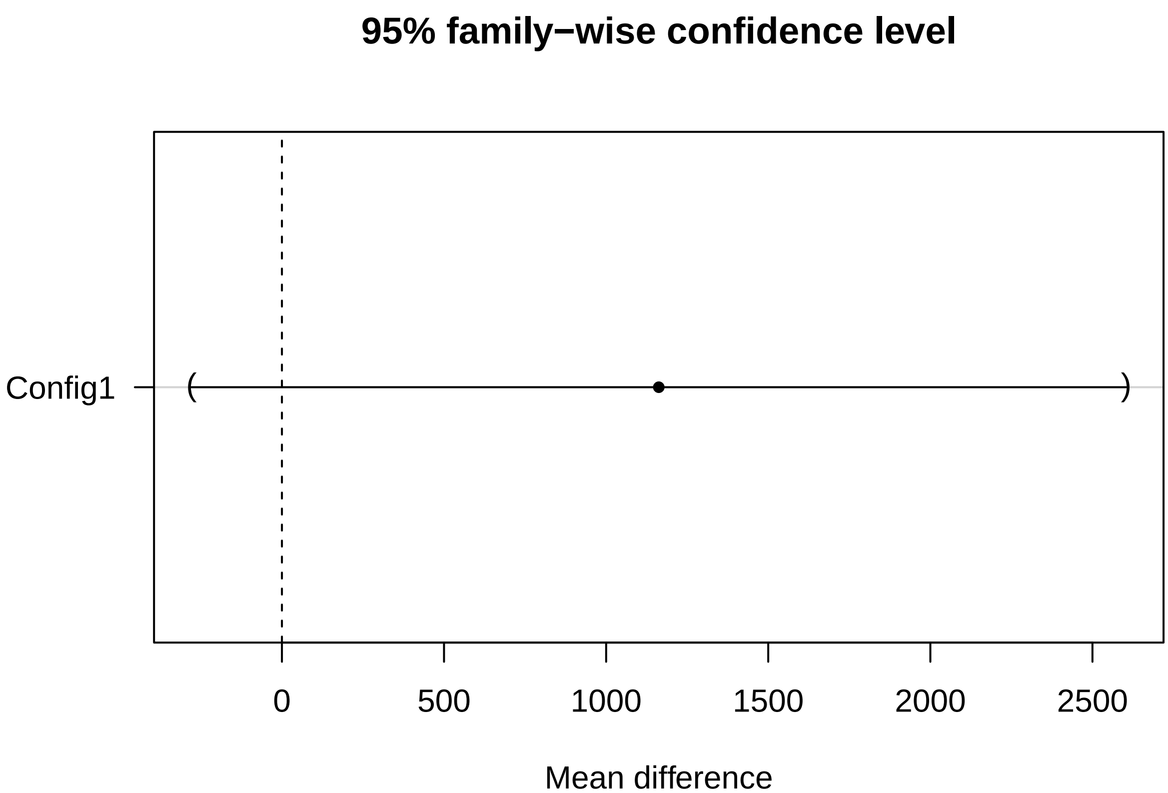
A eficiência relativa da blocagem é um parâmetro que podemos utilizar para comparar observações CRD com a RCBD para sabermos, em porcentagem, quantas observações a mais teríamos que fazer no CRD para ter o mesmo poder estatístico que no RCBD. Para o modelo linear ajustado, obtemos o seguinte valor:

O valor obtido mostra que teríamos que fazer cerca de 17,6% mais observações no estudo aleatorizado (CRD) do que na análise em blocos (RCBD) para ter o mesmo poder estatístico.

Por fim, utilizando o teste de Dunnett para realizar uma comparação entre as configurações, em escala logaritmica, temos:



Para a plotagem em escalar linear, temos:



Atividades específicas

Ambos os autores realizaram a avaliação dos dados estatísticos, pesquisaram sobre a ferramenta utilizada para os cálculos, realizaram correções nos trabalhos, implementações em R e sugestão de testes. Desenvolvemos também uma maneira para paralelizar os testes, os arquivos serão enviados juntamente com o .zip. A diferença entre a forma como usamos o script e o R Markdown é que, ao chamar várias instâncias, passamos para o sistema operacional o gerenciamento das tarefas, para que ele possa realizar o paralelismo e uso de mais cores, quando necessário. Assim, o tempo de execução dos dados foi reduzido drasticamente, em comparação com a execução sequencial.

Conclusões

A partir dos testes performados, os resultados estatísticos permitiram responder às perguntas de interesses.

Existe sim uma tendência de desempenho médio melhor das configurações idenditificadas como 1 em relação as identificadas como 2, isso pode ser observado no gráfico de comparação das Instancias por Y. Apesar de ser uma leve tendência, existem muito mais "outliers" da configuração 2 (denominada de "Config 2" na análise). Além disso, pode-se observar que há maior efeito quando realizamos comparações dentro de uma mesma instância do que ao longo delas, indicando um maior efeito vertical, do que horizontal, conforme mostrado no teste F.

Segundo o teste de Dunnett que foi performado com intuito de dizer qual a melhor configuração do algoritmo, podemos confirmar a afirmação anterior. O algoritmo com configurações 1 (Config) é melhor que a configuração 2 (Config 1), uma vez que este apresenta tempo de execução médio menor que a outra configuração, permitindo realizar otimizações mais rapidamente.