Planejamento e Análise de Experimentos (EEE933)
Estudo de Caso 4

Pedro Vinícius, Samara Silva e Savio Vieira

5 de Outubro de 2020

Introdução

O algoritmo de Evolução Diferencial (DE, do inglês *Differential Evolution*) é uma estratégia baseada em população comumente utilizada para resolver problemas de otimização cujos métodos baseados em gradiente são inviáveis ou apresentam déficit de desempenho. Na linguagem R, sua implementação está disponível no pacote `ExpDE` [2] com o seguinte protótipo:

```
ExpDE(popsize, mutpars = list(name = "mutation_rand", f = 0.2),
      recpars = list(name = "recombination_bin", cr = 0.8, nvecs = 1),
      selpars = list(name = "standard"), stopcrit, probpars, seed = NULL,
      showpars = list(show.iters = "none"))
```

onde `popsize` é o tamanho da população, `mutpars`, `recpars` e `selpars` são as listas com as definições dos parâmetros de mutação, recombinação e seleção, respectivamente, `stopcrit` é a lista com as definições dos critérios de parada, `probpars` é a lista com os parâmetros relacionados a instância do problema, `seed` é a semente do gerador de números aleatórios e, por fim, `showpars` é a lista que controla o histórico que será impresso no terminal durante a execução do algoritmo.

No presente estudo serão investigadas duas configurações diferentes deste algoritmo:

Algoritmo 1:
Recombinação *Blend Alpha Beta* com $\alpha = 0,4$ e $\beta = 0,4$
`recpars` <- list(name = "recombination_blxAlphaBeta", alpha = 0.4, beta = 0.4)
Mutação nos indivíduos aleatórios com um fator de escala $f = 4$ para o vetor de diferenças
`mutpars` <- list(name = "mutation_rand", f = 4)

Algoritmo 2:
Recombinação binomial com probabilidade $c_r = 0,9$
`recpars` <- list(name = "recombination_eigen", othername = "recombination_bin", cr = 0.9)
Mutação nos melhores indivíduos com um fator de escala $f = 2,8$ para o vetor de diferenças
`mutpars` <- list(name = "mutation_best", f = 2.8)

A função de teste escolhida foi a Rosenbrock, cuja equação é dada por (1):

$$f(\mathbf{x}) = \sum_{i=1}^{D-1} (100(x_i^2 + x_{i+1})^2) + (x_i - 1)^2) \tag{1}$$

onde D é a dimensão do problema e x_i é a i-ésima variável do indivíduo $\mathbf{x}$, tal que $x_i \in$ [-5, 10] para $i \in [1, D]$. A Figura 1 apresenta a superfície bidimensional da função descrita e ressalta sua principal propriedade que é ter um vale estreito do ótimo local ao ótimo global.

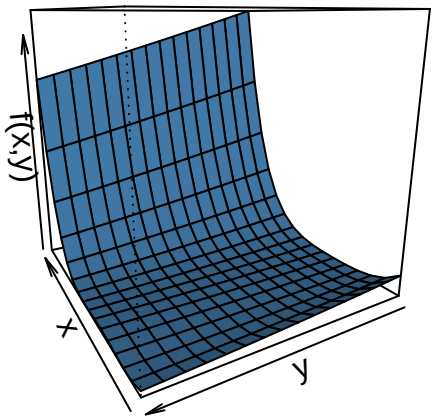

Figura 1: Função Rosenbrock ($D = 2$).

Os demais parâmetros foram definidos igualmente para ambas configurações durante toda a experimentação, sendo tamanho da população `popsize` $= 5 \times D$, número máximo de avaliações na função objetivo `maxevals` $= 5000 \times D$ e número máximo de iterações `maxiter` $= 100 \times D$.

Planejamento dos Experimentos

Dado que o propósito do estudo é analisar o desempenho de duas configurações de algoritmos de otimização, os valores da função objetivo (*fitness*) serão utilizados como métrica de qualidade das soluções encontradas. É interessante ressaltar que em problemas de minimização, como é o caso da função Rosenbrock, soluções boas apresentam valores pequenos de $f(\mathbf{x})$ e soluções ruins apresentam valores maiores de $f(\mathbf{x})$. Assim, ao final de cada execução de um algoritmo para uma determinada instância do problema, apenas o valor de fitness do melhor indivíduo da população será tomado como referência.

As hipóteses estatísticas foram definidas com o objetivo de verificar as seguintes proposições:

- Há alguma diferença no desempenho médio das duas configurações propostas do algoritmo de Evolução Diferencial para a classe de problemas de interesse?
- Caso haja, qual a magnitude dessa diferença encontrada?
- Se possível ainda, qual configuração é superior em termos da qualidade média das soluções encontradas no conjunto de instâncias definido?

Considerando as questões propostas, foram estabelecidas as seguintes hipóteses de teste sobre o *fitness* médio das duas configurações de algoritmo [4]:

$$\begin{cases} H_0 : \mu_{\text{algo}_1} \geq \mu_{\text{algo}_2} \\ H_1 : \mu_{\text{algo}_1} < \mu_{\text{algo}_2} \end{cases} \tag{2}$$

Os parâmetros experimentais considerados para realização dos testes foram nível de significância $\alpha = 0,05$, mínima diferença de importância prática (padronizada) $d^* = \delta^*/\sigma = 0,5$ e potência mínima desejada $\pi = 1 - \beta = 0,80$ para tamanho de efeito $d = d^*$.

Coleta de Dados

A classe de funções de interesse para o presente experimento é composta por instâncias de dimensão no domínio $2 \leq D \leq 150$. De modo a permitir que o Teorema Central do Limite (TCL) seja evocado, se necessário, cada algoritmo foi executado 33 vezes para cada instância do problema. Supondo todas as instâncias no intervalo desejado, ao todo foram cerca de $n_{\text{algoritmos}} \times n_{\text{instâncias}} \times n_{\text{execuções por instância}} = 2 \times 149 \times 33 = 9834$ execuções. Essa experimentação exaustiva foi realizada por uma máquina Intel(R) Xeon(R) Gold 6140 de 18 núcleos e 36 threads operando a 2.30 GHz, 256 GB de RAM, RAID 5 com SSDs de 1TB e Placa de Vídeo Nvidia Quadro P5000 com 2560 cores CUDA e 16GB GDDR5X.

Embora para este estudo de caso o orçamento computacional não tenha sido um gargalo insuperável, problemas de engenharia, cuja solução emprega heurísticas para otimizar modelos numéricos, podem requerer uma limitação no número de instâncias. Assim, diferenças no desempenho médio que excedem algum limite mínimo de relevância prática d^* ainda podem ser alcançados em um subconjunto das instâncias disponíveis, a um custo computacional muito menor do que o que seria necessário para a investigação completa [4].

O método `calc_instances` do pacote `CAISEr` em linguagem R permite estimar o número de instâncias mínimas necessárias para se comparar múltiplos algoritmos, de modo que os requisitos experimentais α, d^* e π sejam atingidos [5]. O parâmetro número de comparações `ncomparisons` é dado pela Equação (3):

$$\texttt{ncomparisons} = \frac{K \times (K-1)}{2} \tag{3}$$

onde K é o número de algoritmos que se deseja comparar. Para o estudo em questão, $K = 2$ e, portanto, $\frac{2\times(2-1)}{2} = 1$.

```
out <- calc_instances(ncomparisons = 1,
                      d = 0.5,
                      power = 0.80,
                      sig.level = 0.05,
                      alternative.side = "one.sided",
                      power.target = "mean")
cat('Número de instâncias necessárias:', out$ninstances)
```

```
## Número de instâncias necessárias: 27
```

O número de instâncias necessárias para se realizar o experimento descrito é de apenas 27 instâncias, ou seja, 1782 execuções no total, o que demanda cerca de $81,88\%$ a menos de processamento computacional que a experimentação integral. Desse modo, as instâncias seriam amostras igualmente espaçadas no domínio $D \in [2, 150]$.

```
options(width = 90)
round(linspace(2, 150, out$ninstances), digits = 0)
```

```
##   [1]   2   8  13  19  25  30  36  42  48  53  59  65  70  76  82  87  93  99 104 110 116
## [22] 122 127 133 139 144 150
```

Análise Exploratória de Dados

Diante do demasiado número de instâncias do problema, ainda que sejam consideradas apenas as instâncias mínimas definidas anteriormente, analisar estatísticas amostrais, como média, mediana e desvio, par a par, com o intuito de gerar algumas suposições sobre potenciais diferenças nos algoritmos se torna impraticável. Portanto, a fim de compreender melhor os dados em estudo e, posteriormente, inferir sobre as populações de onde as amostras provêm, serão analisadas algumas representações gráficas.

No que se refere ao gráfico de *fitness* médio por instância do problema, é possível evidenciar que, para um intervalo visualmente estimado de $10 < D < 60$, a média de desempenho do Algoritmo 1 é substancialmente pior que a média de desempenho do Algoritmo 2. À medida que a dimensão do problema cresce, as diferenças

entre os valores médios já não são mais perceptíveis. Essa verificação ressalta ainda mais a importância de se realizar uma amostragem uniformemente distribuída de instâncias no intervalo de interesse. Caso contrário, as análises podem ser completamente enviesadas. No presente estudo, o Algoritmo 2, ao que tudo indica, apresenta uma melhor performance para dimensões relativamente menores. Se porventura o subconjunto de instâncias abrangesse apenas problemas de baixa dimensão, por exemplo, as conclusões tiradas seriam polarizadas.

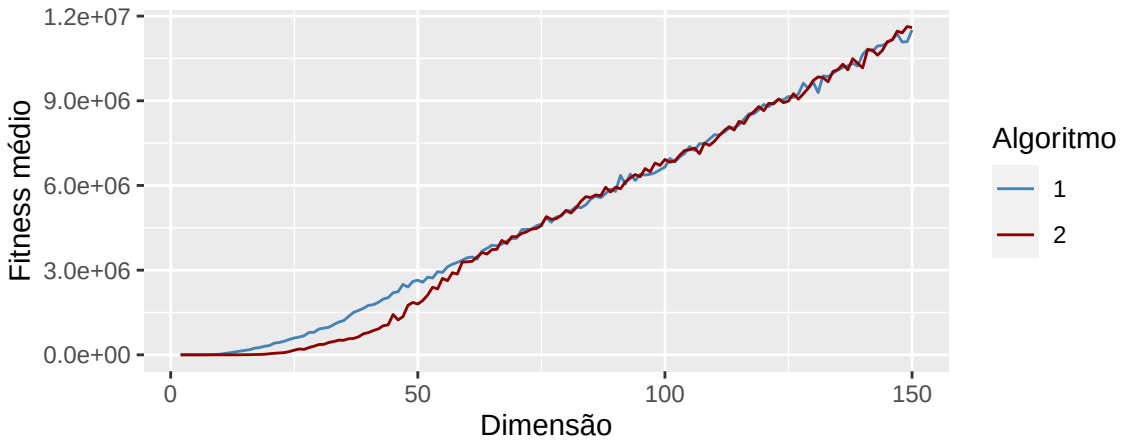

Figura 2: Média de desempenho dos algoritmos em todas as 149 instâncias.

Com a finalidade de verificar o comportamento dos algoritmos nas primeiras instâncias, foram gerados diagramas de caixa par a par no intervalo $2 \leq D \leq 11$. As diferenças evidentes no gráfico anterior têm origem na instância $D = 8$, onde já se pode perceber uma mediana maior do Algoritmo 1 em relação ao Algoritmo 2, bem como uma tendência de crescimento apenas do Algoritmo 1 nas instâncias seguintes. Além disso, a dispersão da primeira configuração apresenta um comportamento com princípio exponencial ($D \geq 8$), enquanto que para a segunda configuração não é possível visualizar o intervalo interquatilico entre o terceiro e o primeiro quartil para o mesmo domínio.

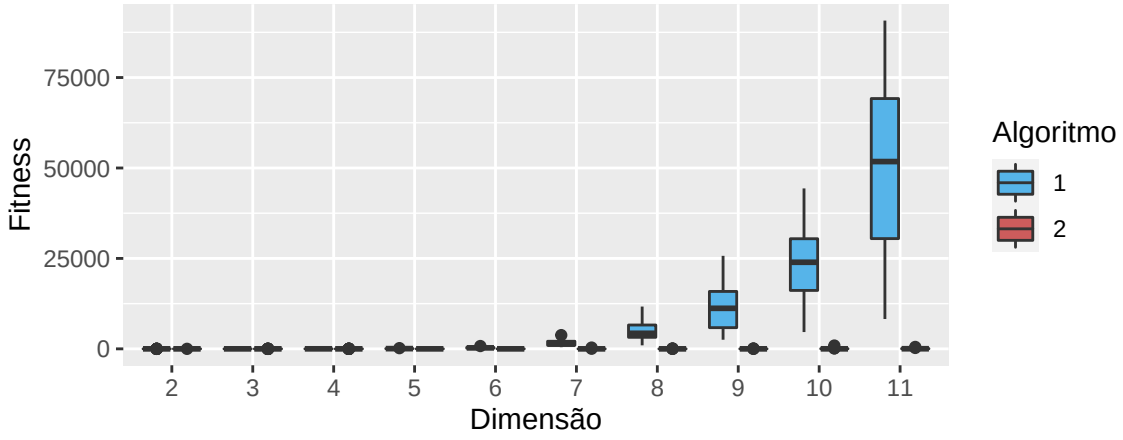

Figura 3: Boxplot das dez primeiras instâncias.

Analogamente, os diagramas de caixa par a par das últimas instâncias também foram produzidos. Conforme mencionado anteriormente para o gráfico de *fitness* médio por instância, as diferenças nos desempenhos dos algoritmos se tornam imperceptíveis em dimensões maiores. Nas instâncias $D = 141, 142, 148$ e 149, por exemplo, a mediana do Algoritmo 1 se encontra inferior a mediana do Algoritmo 2. Em contrapartida, nas

instâncias $D = 143$ e 144, se observa o inverso. Por fim, não se pode concluir nada a respeito dos segundos quartis nas instâncias $D = 145$ e 146. Essa incerteza acerca das comparações entre as duas configurações propostas do algoritmo de Evolução Diferencial torna ainda mais clara a necessidade da análise estatística para o estudo de caso que se segue.

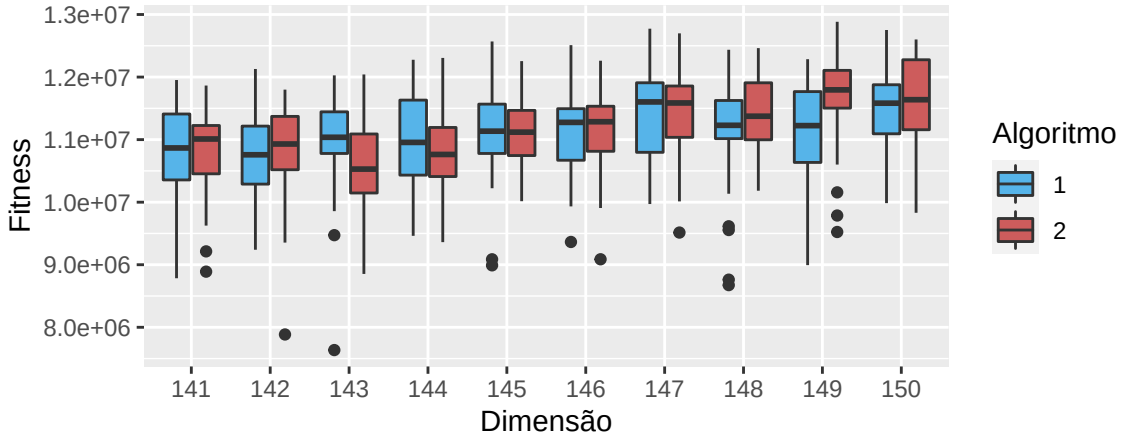

Figura 4: Boxplot das dez últimas instâncias.

Potência do Teste

Tendo em vista que a análise estatística do presente trabalho será realizada sobre a experimentação completa, isto é, as 149 instâncias do problema ao invés apenas da quantidade mínima calculada, a potência obtida no teste sofrerá uma mudança e, portanto, deve ser calculada para o novo tamanho amostral. Dado que deseja-se manter o tamanho de efeito constante ($d^* = 0,5$), tem-se a seguinte relação entre potência do teste e número de instâncias para um nível de significância igual a $\alpha = 0,05$.

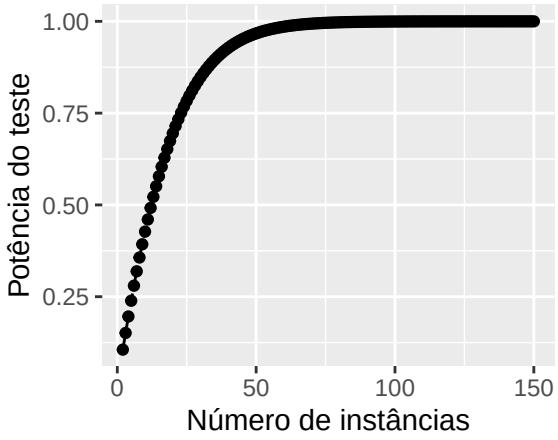

Figura 5: Relação entre o poder do teste e o número de instâncias.

Aumentar o número de instâncias torna o teste de hipótese mais sensível, isto é, mais provável de rejeitar a hipótese nula quando ela é, de fato, falsa. Em outras palavras, a probabilidade de cometer um erro do tipo II (β) diminui à medida que o tamanho amostral aumenta. Como a potência do teste é o complemento do erro do tipo II ($\pi = 1 - \beta$), então consequentemente ela aumenta. Desse modo, tem-se uma potência de $99,99\%$ ao se considerar 149 instâncias do problema com os demais parâmetros experimentais fixos.

```r
out <- calc_instances(ncomparisons = 1,
                      d = 0.5,
                      ninstances = 149,
                      sig.level = 0.05,
                      alternative.side = "one.sided",
                      power.target = "mean")
cat('Potência alcançada:', out$power)
```

```
## Potência alcançada: 0.9999953
```

Validação de Premissas

Uma possibilidade de conduzir a análise deste experimento é por meio do Planejamento por Blocagem Completamente Randomizado, do inglês *Randomized Complete Block Design* (RCBD). Este método, também conhecido como Blocagem (*Blocking*), permite isolar os fatores indesejados, conhecidos e desconhecidos, e consequentemente conduzir a uma análise comparativa mais fidedígna, uma vez que o poder estatístico é impulsionado ao excluir a variabilidade entre as repetições dos resíduos [3]. Neste contexto, embora a aplicação do algoritmo em diferentes dimensões possa causar um efeito indesejado, a separação das instâncias por blocos exclui o efeito da instância e permite o estudo da média dos algoritmos.

Este experimento assume que há uma observação por bloco, blocos independentes e independência da aleatorização dentro do bloco [3]. Dessa forma, cada uma das dimensões dentro do intervalo proposto foi considerada como uma instância do problema, e por isso, a média de cada uma das 33 execuções por instância é uma observação do bloco referente à respectiva instância, e somente desta instância.

Para reduzir possíveis viéses causados por outliers, e melhorar o ajuste dos dados, facilitando a análise destes, foi aplicado uma transformação logarítmica sobre a variável de saída (*fitness* médio).

```r
model <- aov(formula = log(f)~Algoritmo+Instancia_Grupo, data = aggdata)
```

Assim como no Anova, as premissas deste teste são normalidade dos resíduos, homocedasticidade e independência amostral. Tendo isso em vista, e com intuito de garantir a normalidade dos dados, cada uma das instâncias foi executada 33 vezes, pois assim pode-se evocar o Teorema do Limite Central.

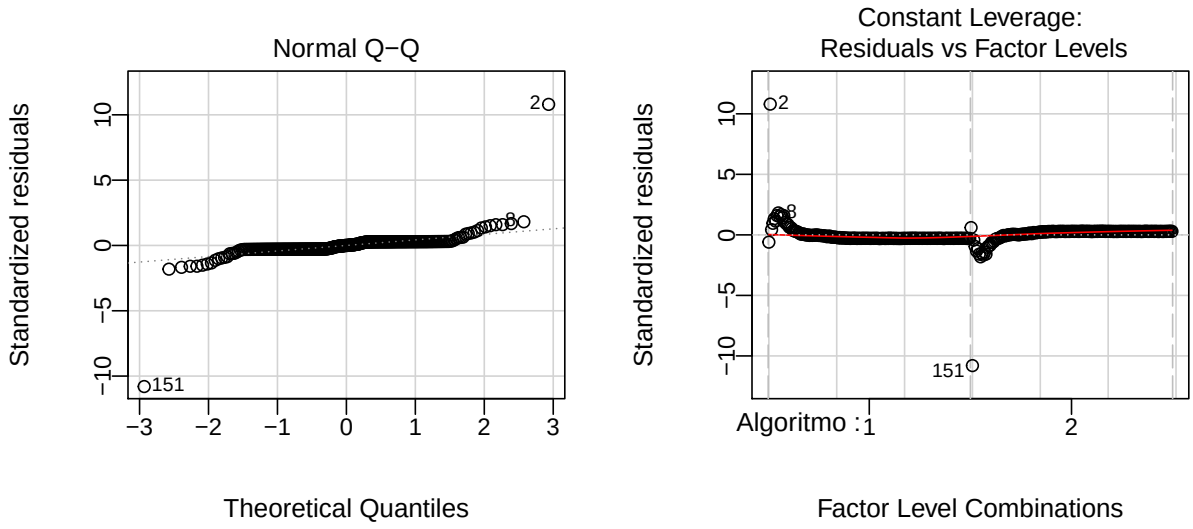

Figura 6: QQ-Plot e Resíduos por Níveis de Fatores.

Ainda sim, é importante notar que graficamente, os resíduos se ajustam bem à reta do QQ-plot, com a presença de somente dois outliers, referentes à dimensão igual a 2 de cada um dos algoritmos. Quanto à homocedasticidade, observa-se que os resíduos comportam-se de maneira semelhante, com variabilidade constante, independente do algoritmo. A fim de garantir a independência das amostras, todas as execuções ocorreram de forma independente na mesma máquina, conforme relatado na coleta de dados. Diante da validação das premissas, é possível prosseguir para a análise estatística inicialmente proposta.

Análise Estatística

No R é possível ajustar um modelo de análise de variância (ANOVA) por meio da função `aov` [1], passando como parâmetro tanto o fator experimental a ser analisado (os algoritmos), como o fator a ser bloqueado, que neste contexto são as instâncias. Pode-se verificar que o somatório dos quadrados das instâncias foi muito superior ao observado nos algoritmos. Com isso, percebe-se que a blocagem deste elemento mostra-se importante para análise correta dos dados. Observa-se ainda que tanto em relação aos algoritmos, como se tratando das instâncias, os p valores foram substancialmente pequenos, $p = 0,00061$ e $p = 2 \times 10^{-16}$, respectivamente, o que permite inferir que a hipótese nula deste teste é rejeitada ao nível de confiança de 95%. Assim, é possível relatar que as diferenças observadas nas médias dos algoritmos de fato existem, além de se estimar quão grande pode ser o efeito das instâncias sobre resultado do algoritmo, caso não seja bloqueado.

```
summary(model)
```

```
##                  Df Sum Sq Mean Sq F value  Pr(>F)
## Algoritmo         1     68   68.01  12.267 0.00061 ***
## Instancia_Grupo 148   5016   33.89   6.113 < 2e-16 ***
## Residuals       148    820    5.54
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Esta função também tem como saída o coeficiente de determinação, também conhecido como R^2. Trata-se de uma medida de ajuste que determina o quanto a variância dos dados pode ser explicada pelo modelo proposto. O R^2 pode assumir valores entre 0 e 1, e quanto mais próximo de 1, mais o modelo é uma boa representação das diferenças observadas. Neste contexto, R^2 resultou em $0,86$ que evidencia que as conclusões obtidas definem bem o experimento proposto.

```
cat('Coeficiente de determinação:', summary.lm(model)$r.squared)
```

```
## Coeficiente de determinação: 0.8610401
```

Uma vez identificado através da análise das variâncias (ANOVA) que existe diferença estatística entre as médias das soluções dos algoritmos, deve-se realizar testes comparativos para identificar qual algoritmo tem o melhor desempenho. Quando se deseja realizar múltiplas comparações, dois paradigmas são possíveis: (i) comparação todos contra todos; (ii) e todos contra um. Todos contra um é utilizado quando se tem um tratamento de referência e deseja-se compará-lo com os demais. No caso deste experimento, por se tratar de apenas dois algoritmos ($K = 2$), o teste de comparação de todos contra um coincide com o cenário do caso de todos contra todos. O teste de Dunnett neste caso foi utilizado por se ter maior sensibilidade [3], atuando de forma semelhante ao teste t de Student. Após apresentação do intervalo das diferenças das médias em escala logarítmica, pode-se concluir com 95% de confiança que o Algoritmo 2 apresenta melhor solução que o Algoritmo 1, uma vez que o intervalo de confiança está totalmente contido no domínio negativo ($\mu_{\text{algo}_2} - \mu_{\text{algo}_1} < 0$) e o melhor algoritmo apresenta a menor média (problema de minimização).

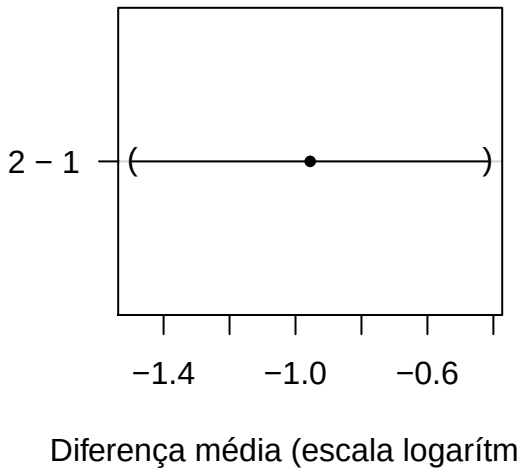

Figura 7: Intervalo de confiança do teste de Dunnet ao nível de confiança de 95%.

Como já foi mencionado anteriormente, os dois algoritmos não apresentam diferenças visualmente significativas nas dimensões extremas. Dessa forma, foi destacado graficamente uma proporção do intervalo das dimensões onde os algoritmos apresentam diferenças na qualidade das soluções $D \in [20, 30]$. Tal evidência corrobora ainda mais o resultado do teste de Dunnet.

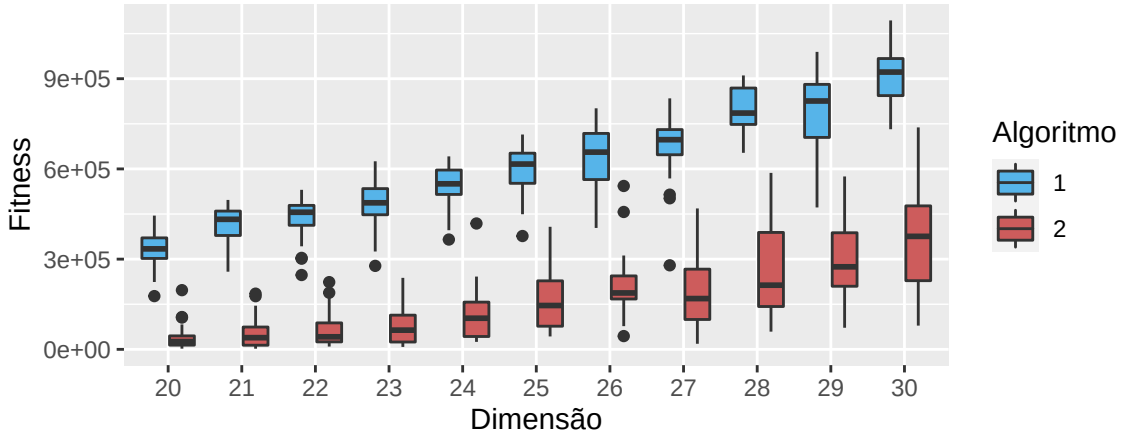

Figura 8: Boxplot das instâncias contidas no domínio $20 \leq D \leq 30$.

Conclusões

O maior desafio do experimento foi definir o número de instâncias mínimas para se obter a potência, o tamanho de efeito e o nível de significância desejados. No entanto, ainda que o número mínimo calculado tenha sido de 27 instâncias, considerou-se todo o domínio do problema, isto é, as 149 instâncias. Para que fosse possível realizar a coleta de dados sem maiores problemas de orçamento computacional, as execuções foram realizadas em uma máquina mais robusta.

Ao comparar dois algoritmos, espera-se que diferenças estatísticas sejam evidenciadas, de tal forma que se tenha um *ranking* de desempenho dos algoritmos. A princípio, graficamente, a diferença se apresentou pouco aparente para a maioria das instâncias, com exceção de um subconjunto de instâncias no domínio $D \in [10, 60]$. Tal indicativo reiterou ainda mais a necessidade de se fazer um estudo estatístico para possíveis ratificações.

Sendo assim, através da análise das variâncias conclui-se ao nível de confiança de 95% que existe diferença estatística entre o *fitness* médio das soluções dos algoritmos e também com 95% de confiança, através do teste de comparações múltiplas, que a segunda configuração do DE apresenta melhor solução média que a primeira configuração apresentada para o problema de interesse. Uma vez que se tem comprovado tais evidências estatísticas, pode-se corroborar a hipótese nula H_0 principal do estudo ao nível de significância de 5% ($\mu_{\text{algo}_1} \geq \mu_{\text{algo}_2}$).

Discussão de Melhorias

Dentre as principais melhorias possíveis, algumas já foram consideradas no decorrer do planejamento e da análise dos experimentos. No que se refere ao processo de coleta de dados, por exemplo, um número maior de instâncias foi considerado, de modo a maximizar a potência do teste e de tornar o estudo mais completo. É interessante ressaltar ainda que ao realizar as execuções dos algoritmos em uma mesma máquina foi possível isolar um possível efeito das diferenças das máquinas.

Sob a perspectiva da análise estatística, optou-se ainda pela utilização do modelo de análise de variância pela possibilidade de se verificar além da divergência entre os algoritmos, o impacto que as instâncias poderiam ter, caso não fossem bloqueadas. Embora o ANOVA em conjunto com o teste de Dunnet seja uma abordagem adequada e bastante efetiva, geralmente tal associação é utilizada quando há pelo menos três tratamentos. Portanto, outros métodos mais simples poderiam chegar aos mesmos resultados, como é o caso do teste t de duas amostras para a comparação dos algoritmos.

Atividades Desempenhadas

A hipótese de teste foi definida em concordância com os três autores. A coleta, o pré-processamento e a análise exploratória dos dados foram conduzidas pelo Pedro, bem como o estudo da potência do teste. A validação das premissas foi elaborada pela Samara. A análise estatística, por sua vez, que inclui tanto a verificação da existência de diferenças entre as médias dos dois algoritmos quanto o teste comparativo a posteriori, foi realizada pelos três autores. Por fim, a Samara e o Savio concluíram o trabalho com as considerações finais e discussões de melhoria.

Referências

[1] AOV: Fit An Analysis Of Variance Model. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/aov, 2020. stats version v3.6.2.

[2] Felipe Campelo. Modular Differential Evolution for Experimenting with Operators. https://www.rdocumentation.org/packages/ExpDE/versions/0.1.2, 2016. Version 0.1.2.

[3] Felipe Campelo. Lecture Notes on Design and Analysis of Experiments. http://git.io/v3Kh8, 2018. Version 2.12; Creative Commons BY-NC-SA 4.0.

[4] Felipe Campelo and Fernanda Takahashi. Sample Size Estimation for Power and Accuracy in the Experimental Comparison of Algorithms. *Journal of Heuristics*, 25(2):305–338, 2019.

[5] Felipe Campelo, Fernanda Takahashi, and Elizabeth Wanner. CAISEr: Comparing Algorithms with Iterative Sample-size Estimation in R. https://www.rdocumentation.org/packages/lmtest/versions/0.9-37/topics/dwtest. Documentation reproduced from package CAISEr, version 1.0.16.