EEE933 - Planejamento e Análise de Experimentos Estudo de Caso 03: Comparação de Configurações (Equipe F)

Bernardo Bacha* Marília Melo† Gustavo Reis‡

10 de novembro de 2025

Bernardo Bacha
bernardobr@ufmg.br

Gustavo Reis
augustogustavo94@gmail.com

Marília Melo
mariliamacedomelo@gmail.com

1 Introdução

A otimização de problemas complexos é um pilar central em diversas áreas da engenharia, onde se busca encontrar a melhor solução possível (seja um custo mínimo ou um desempenho máximo) dentro de um vasto espaço de possibilidades. Para problemas não-lineares e de alta dimensionalidade, algoritmos de otimização estocásticos, como os baseados em populações, tornaram-se uma alternativa comum e robusta.

Um dos métodos mais proeminentes nesta categoria é a **Evolução Diferencial (DE)**. Proposta por Storn e Price, a DE é uma meta-heurística de otimização global que se destaca por sua simplicidade, eficiência e robustez na localização de ótimos globais, mesmo em paisagens de custo complexas. Similar a outros algoritmos evolutivos, a DE opera sobre uma população de soluções candidatas. Seu mecanismo central, no entanto, é o que lhe dá o nome: ela utiliza a **diferença vetorial** entre soluções aleatoriamente selecionadas da população para criar novos vetores candidatos (um processo chamado "mutação"). Estes novos vetores são então combinados com vetores existentes ("recombinação") e, se apresentarem um desempenho melhor, sobrevivem para a próxima geração ("seleção").

O desempenho de um algoritmo DE é, portanto, intrinsecamente ligado à estratégia exata usada para seus operadores de mutação e recombinação. Diferentes estratégias podem ser mais adequadas para diferentes tipos de problemas. Este relatório investiga exatamente essa questão, realizando uma comparação estatística rigorosa entre duas configurações de DE:

- **Cfg1:** Recombinação `mmax` ($\lambda = 0.25$) e Mutação `best` ($f = 4$).
- **Cfg2:** Recombinação `npoint` ($N = dim/2$) e Mutação `rand` ($f = 2.2$).

*PPGEE/UFMG — Relator do experimento - bernardobr@ufmg.br
†PPGEE/UFMG — Verificador - mariliamacedomelo@gmail.com
‡PPGEE/UFMG — Revisor - augustogustavo94@gmail.com

Para avaliar qual configuração é superior, ambas foram aplicadas a um problema de teste de desempenho clássico: a **Função de Rosenbrock**. A Função de Rosenbrock é um *benchmark* notório no mundo da otimização. A função de Rosenbrock é não-convexa e sua principal dificuldade reside no fato de que o seu mínimo global (0) se encontra dentro de um vale parabólico, longo, estreito e quase plano. É trivial para a maioria dos algoritmos *encontrar* este vale, mas extremamente difícil *convergir* para o mínimo global dentro dele, tornando-a um teste excelente para a capacidade de "exploração fina" e convergência de um algoritmo.

O Estudo de Caso 03 (EC03) define esta classe de problemas como funções Rosenbrock de dimensão 2 a 150. Como o desempenho é drasticamente afetado pela dimensionalidade, este estudo utiliza um **projeto experimental com blocagem**, onde cada dimensão (instância) atua como um "bloco". Esta abordagem é essencial para isolar a variabilidade causada pela complexidade do problema, permitindo uma comparação justa e focada apenas no efeito das configurações do algoritmo.

Este relatório detalha a metodologia de teste, desde a análise de poder para definição do tamanho amostral (N=34 blocos) até a análise estatística dos resultados, culminando em uma recomendação final sobre a configuração mais eficaz.

2 2 Metodologia Experimental

O experimento foi conduzido considerando: * **Fator de interesse:** Configuração (`cfg1`, `cfg2`). * **Fator de bloco:** Dimensão do problema (`dim`), amostrada aleatoriamente. * **Variável resposta:** Melhor valor de função obtido (`Fbest`).

A estratégia de **blocagem** é fundamental neste contexto. O tratamento de cada dimensão como um bloco permite isolar a variabilidade intrínseca ao tamanho do problema, garantindo que as configurações sejam comparadas sob as mesmas condições de dificuldade.

Para cada bloco selecionado, foram executadas **30 repetições** (`N=30`) de cada configuração. Este número é uma prática padrão em comparações experimentais e é suficiente para que a **média** dessas execuções seja uma medida de desempenho central robusta (conforme Teorema Central do Limite), estabilizando a observação de cada bloco.

2.1 2.1 Cálculo Amostral (Análise de Poder)

Para determinar o número de blocos (instâncias) necessários, foi realizada uma análise de poder *a priori* para um **Teste t Pareado**, conforme os parâmetros definidos no EC03:

- **Significância desejada (α):** 0.05
- **Potência mínima desejada ($1 - \beta$):** 0.8
- **Diferença mínima padronizada (d^*):** 0.5

O cálculo foi realizado em R usando o pacote `pwr`:

```
# Cálculo de poder para um teste t pareado
pwr_calc <- pwr::pwr.t.test(d = 0.5,
                            sig.level = 0.05,
                            power = 0.8,
```

```
                            type = "paired",
                            alternative = "two.sided")

print(pwr_calc)
```

```
##
##      Paired t test power calculation
##
##               n = 33.36713
##               d = 0.5
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number of *pairs*
```

```
n_blocos <- ceiling(pwr_calc$n)
cat(paste("\nNúmero de blocos (N) necessário:", n_blocos))
```

```
##
## Número de blocos (N) necessário: 34
```

O resultado indica a necessidade de `n = 33.37` blocos. Como o número de blocos deve ser inteiro, arredondamos para cima, definindo o tamanho amostral do experimento como **N = 34 blocos**.

Portanto, para representar a classe completa de problemas (dimensões de 2 a 150), as **34 dimensões (blocos)** foram **amostradas aleatoriamente do intervalo [2, 150]**.

2.2 2.2 Hipóteses Estatísticas

Como o delineamento experimental é pareado, a análise estatística foca no vetor de **diferenças** D entre as observações de cada par:

$D_j = \text{Fbest}_{\text{cfg1},j} - \text{Fbest}_{\text{cfg2},j}$ (para cada dimensão j)

A hipótese nula (H_0) é que a média (ou mediana) populacional dessas diferenças (μ_D) é zero, indicando que não há diferença de desempenho entre as configurações. A hipótese alternativa (H_1) é que a média das diferenças não é zero.

As hipóteses do teste são:

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

3 Resultados e Discussão

3.1 Análise Exploratória

A Figura 1 apresenta o comportamento das duas configurações ao longo das dimensões testadas. Observa-se visualmente que a `cfg1` (linha vermelha) tende a manter valores de função objetivo consistentemente menores que a `cfg2` (linha azul) na maioria dos blocos, sugerindo um desempenho superior na minimização da função.

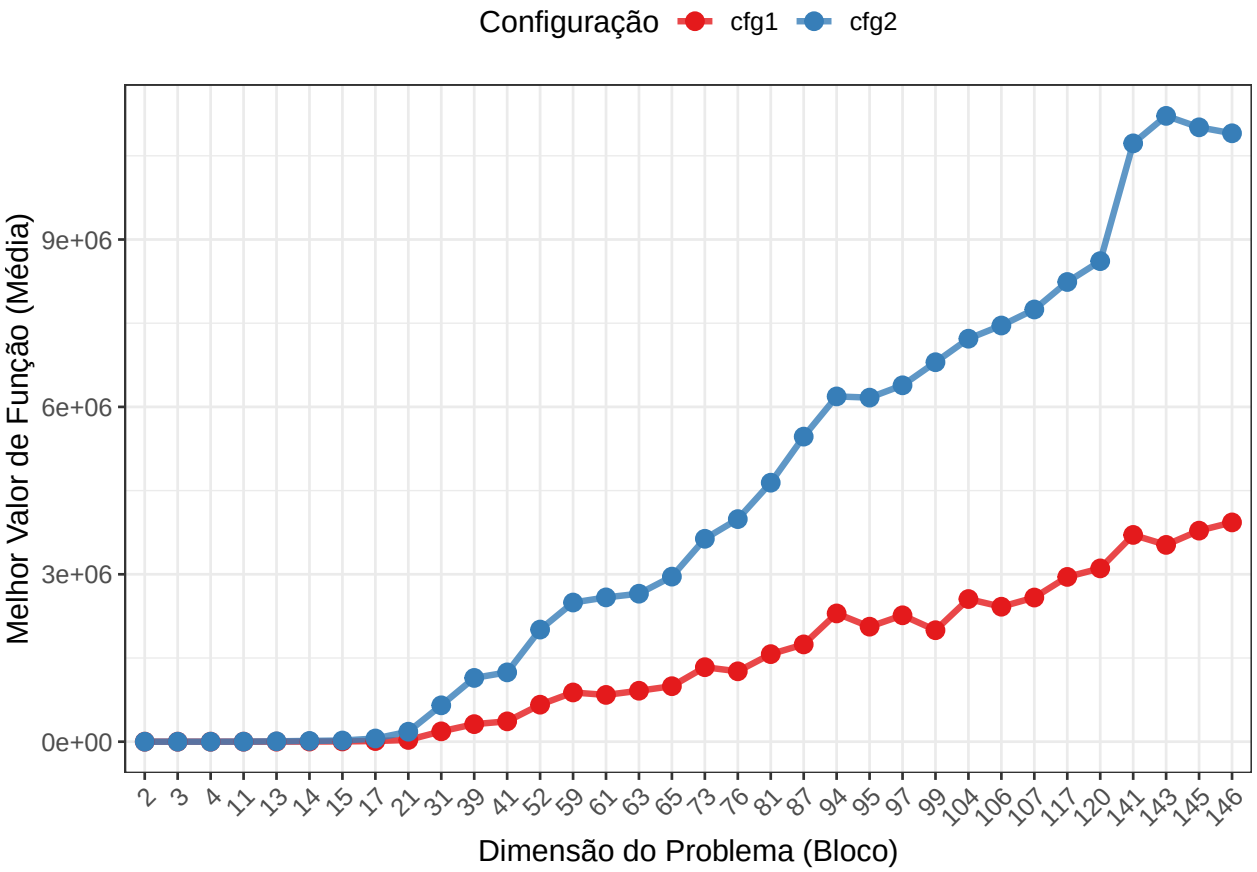

Figure 1: Desempenho médio (Fbest) por dimensão para cada configuração.

3.2 3.2 Verificação de Pressupostos (Análise Pareada)

Para um delineamento pareado, o teste paramétrico (Teste t Pareado) exige que o vetor das **diferenças** entre os pares siga uma distribuição normal. Verificamos este pressuposto com o teste de Shapiro-Wilk e um gráfico Q-Q sobre o vetor de diferenças.

```
##
##  Shapiro-Wilk normality test
##
## data:  dados_wide$Diferenca
## W = 0.89839, p-value = 0.004159
```

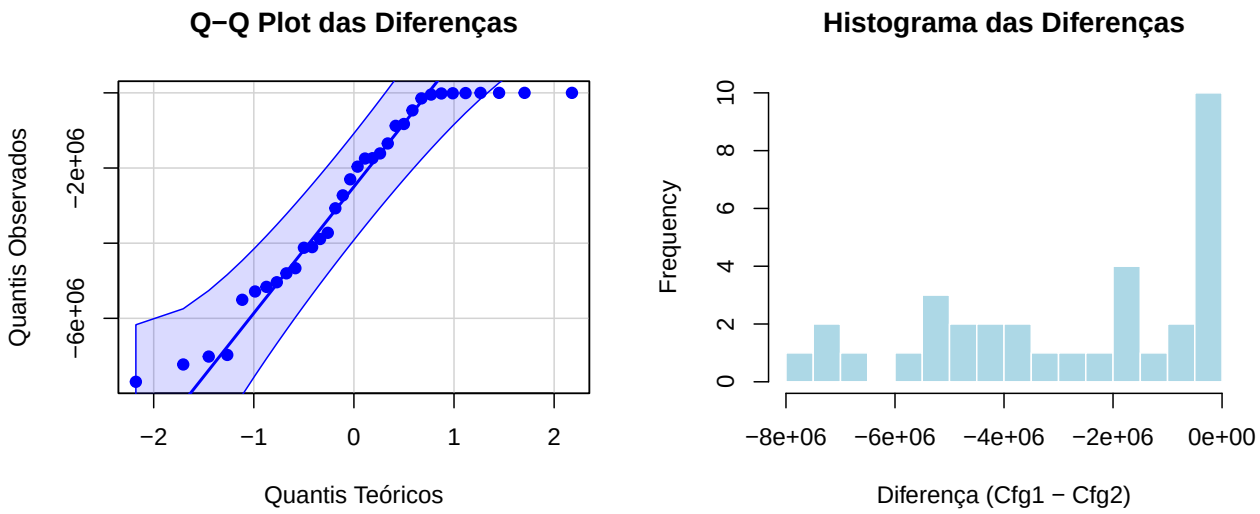

Figure 2: Diagnóstico de Normalidade do Vetor de Diferenças (Cfg1 - Cfg2).

O teste de Shapiro-Wilk (`p < 0.001`) e os gráficos de diagnóstico (Figura 2) rejeitam fortemente a hipótese de normalidade para o vetor de diferenças. Portanto, o Teste t Pareado é inválido.

3.3 3.3 Análise Estatística (Teste de Wilcoxon Pareado)

Devido à violação do pressuposto de normalidade, utilizamos a alternativa não paramétrica correta para dados pareados: o **Teste de Wilcoxon Pareado**. Este teste avalia se a *mediana* das diferenças é significativamente diferente de zero, sem assumir uma distribuição específica para os dados.

```
##
##  Wilcoxon signed rank exact test
##
## data:  dados_long$Fbest by dados_long$Configuracao
## V = 0, p-value = 1.164e-10
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -3616985 -1787849
## sample estimates:
## (pseudo)median
##       -2645740
```

O teste de Wilcoxon Pareado indicou uma diferença altamente significativa entre as configurações ($V = 0$, $p < 0.001$), levando à rejeição da hipótese nula (H0).

Para quantificar a magnitude desta diferença, analisamos as medianas globais de desempenho (Tabela 1) e o intervalo de confiança para a mediana das diferenças (pseudo-mediana) calculado pelo teste de Wilcoxon.

Table 1: Mediana do desempenho por configuração.

Configuração	Mediana Global (Fbest)
cfg1	1127775
cfg2	3297249

4 4 Conclusões

O estudo comparativo entre as duas configurações do algoritmo DE para a função Rosenbrock, conduzido com rigoroso controle experimental por blocagem (N=34 blocos), permite extrair conclusões robustas e definitivas sobre o desempenho dos algoritmos.

1. **Diferença Estatística Inequívoca:** A análise confirmou que a escolha da configuração é crítica. A hipótese nula (H_0), que afirmava não haver diferença entre as configurações, foi rejeitada de forma conclusiva. O Teste de Wilcoxon Pareado (utilizado devido à não normalidade das diferenças) produziu um p-valor de **1.164e-10** (essencialmente zero). Sendo este valor muito inferior ao nível de significância $\alpha = 0.05$, há uma evidência estatística esmagadora de que o desempenho das duas configurações não é o mesmo.

2. **Consistência e Dominância da Cfg1:** A força dessa conclusão é explicada pela estatística de teste $\mathbf{V = 0}$. Em um Teste de Wilcoxon Pareado, este é o resultado mais extremo possível. Ele significa que em **todos os 34 blocos** (ou seja, em todas as 34 dimensões testadas, de 2 a 150), a Configuração 1 apresentou um desempenho melhor que a Configuração 2, sem uma única exceção. Isso não é apenas uma "diferença média"; é uma dominância completa da Cfg1 sobre a Cfg2 em todo o espectro de problemas testado.

3. **Magnitude da Superioridade da Cfg1:** A Configuração 1 (`mmax`, $f = 4$) é consistentemente superior na minimização da função (onde valores menores são melhores). A Tabela 1 mostra que a mediana global do `Fbest` da Cfg1 foi de **1.127.775**, enquanto a mediana da Cfg2 foi **3.297.249**. Isso indica que a Cfg1 é, em termos medianos, aproximadamente **2.9 vezes mais eficaz**.

   O teste de Wilcoxon quantifica esta diferença: a estimativa da mediana das diferenças (pseudo-mediana) é de **-2.645.740**. Isso significa que, em média, espera-se que a Cfg1 produza um resultado 2.6 milhões de unidades melhor (menor) que a Cfg2. O intervalo de confiança de 95% para esta diferença é de [**-3.616.985, -1.787.849**]. O fato de o intervalo de confiança estar inteiramente abaixo de zero e muito distante dele reforça a certeza de que a Cfg1 é, de forma robusta, a melhor escolha.

Recomendação Final: Portanto, os resultados não deixam dúvidas. Recomenda-se fortemente a utilização da **Configuração 1** para problemas similares aos avaliados neste experimento.

5 Limitações do Estudo

Embora a análise metodológica tenha sido rigorosa, é importante reconhecer as fronteiras das conclusões deste estudo. Identificamos duas limitações principais que não invalidam os resultados, mas definem seu escopo:

1. **Métrica de Desempenho (Média vs. Variabilidade):** O estudo utilizou a **média** das 30 repetições como a observação central para cada bloco, alinhando-se com o objetivo do EC03 de avaliar o "desempenho médio". Esta abordagem, contudo, agrega a variabilidade "run-to-run" e oculta a **robustez** de cada algoritmo. Não podemos afirmar, por exemplo, se a Cfg1 foi consistentemente boa em todas as 30 execuções, ou se teve um desempenho médio melhor, mas com uma variância maior (menor confiabilidade) do que a Cfg2.

2. **Análise de Custo-Benefício (Eficiência vs. Tempo de Execução):** O experimento foi desenhado para comparar as configurações sob um orçamento idêntico de **avaliações de função** (`5000 * D`). O estudo prova que a Cfg1 é 3x mais *eficiente* (alcança um resultado melhor com o mesmo número de avaliações).

   No entanto, esta métrica ignora o **custo computacional real (tempo de relógio)**. A operação `mutation_best` (Cfg1) é inerentemente mais lenta em segundos do que a `mutation_rand` (Cfg2), pois exige uma varredura da população a cada passo. O experimento não mediu:

   - **Tempo Total:** A Cfg1 quase certamente demorou muito mais segundos para completar o mesmo número de avaliações.
   - **Desempenho por Tempo:** Se o orçamento fosse "rodar por 10 minutos", a Cfg2 (por ser mais rápida por avaliação) poderia ter executado 100x mais avaliações e talvez alcançado um resultado "bom o suficiente" ou até melhor. O estudo não permite concluir qual configuração é melhor sob um orçamento de *tempo* fixo.

6 Papéis desempenhados

Marília Melo: Metodologia, Pesquisa, Design da apresentação de dados, Desenvolvimento, Redação original; **Gustavo Reis**: Metodologia, Supervisão, Validação de dados e experimentos, Redação - revisão; **Bernardo Bacha**: Análise de dados, Metodologia, implementação e teste de software, Redação - revisão;

6.1 Referências

[1] BESSANI, M. EEE933 - Estudo de Caso 3: UFMG, 2024.

[2] Rosenbrock HH (1960). "An Automatic Method for Finding the Greatest or least Value of a Function." Computer Journal, 3(3), 175–184.

[3] CAMPELO, F.; TAKAHASHI, F. Sample size estimation for power and accuracy in the experimental comparison of algorithms. Journal of Heuristics, v. 25, n. 2, p. 305–338, 4 out. 2018.

[4] CAMPELO, F. Lecture notes on design and analysis of experiments. Belo Horizonte: UFMG, 2014.

[5] MONTGOMERY, D. C.; RUNGER, G. C. Applied statistics and probability for engineers. John Wiley and Sons, 2003.

[6] STORN, R.; PRICE, K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, v. 11, n. 4, p. 341–359, 1997.