

# Minimização do peso de um redutor

## Avaliação do desempenho da heurística *Firefly* num problema *benchmark*

Francisco Power · Pedro Rolo

Submetido: 18/12/2020

**Resumo** Neste projeto, avaliamos o desempenho da heurística *Firefly*, um algoritmo de otimização inspirado na natureza, baseado em *Swarm Intelligence*, na minimização do peso de um redutor, um problema *benchmark*.

**Palavras-Chave** Otimização · Benchmark · PSO · Firefly · Heurísticas

### 1 Introdução

O problema da minimização do peso de um redutor de engrenagens paralelas é um problema *benchmark* conhecido em otimização não-linear. Sendo um problema não linear de elevado grau de complexidade e com restrições, e sendo conhecida uma solução ótima, é ideal para testar o desempenho de novos algoritmos de otimização.

Neste projeto, pretende-se estudar o desempenho do algoritmo *Firefly*, um algoritmo inspirado na natureza, baseado no conhecido *Particle Swarm Optimization*.

### 2 Minimização do peso de um redutor

O problema em questão tem como objetivo minimizar o peso de um redutor de engrenagens paralelas de um andar, variando sete variáveis [1], estando estas sujeitas a determinadas restrições.

---

Francisco Power  
n.º 84706  
E-mail: franciscopower66@ua.pt

Pedro Rolo  
n.º 84803  
E-mail: pmrolo@ua.pt

### 3 Formulação dos problemas de otimização

O problema de otimização em questão pode ser formulado como apresentado seguidamente:

Procurar  $\mathbf{x}$  de modo a :

$$\begin{aligned} \text{minimizar } f(\mathbf{x}) = & 0.7854x_1x_2^2 \\ & \times (3.3333x_3^2 + 14.9334x_3 - 43.0934) \\ & - 1.508x_1(x_6^2 + x_7^2) \\ & + 7.477(x_6^3 + x_7^3) \\ & + 0.7854 * (x_4x_6^2 + x_5x_7^2), \quad (1) \\ \text{sujeito a } g_k(\mathbf{x}) \leq & \dots, \quad k = 1, 11 \\ & \dots \geq x_i \geq \dots, \quad i = 1, 7 \end{aligned}$$

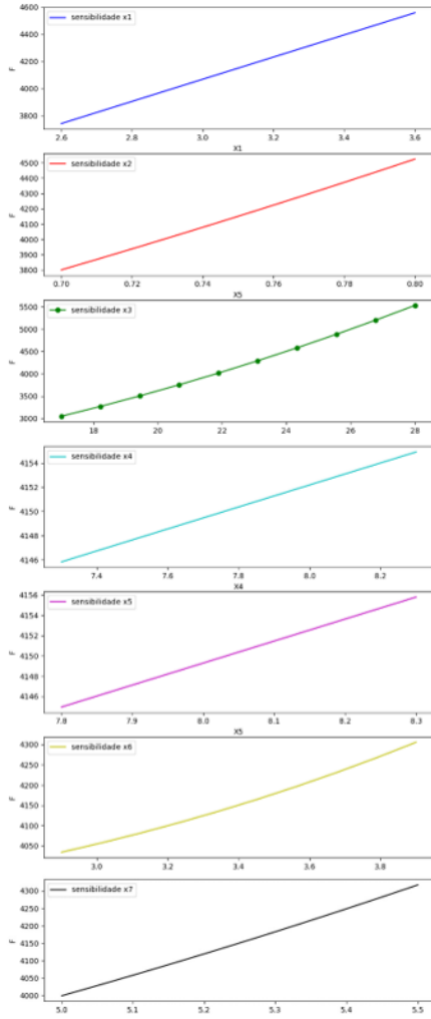
As 11 restrições e os limites das variáveis podem ser encontrados no documento [1].

Tanto a função objetivo como as restrições, são funções não lineares. Assim sendo, este problema é classificado com um problema de otimização não linear, de escala intermédia (uma vez que há mais de 5 variáveis de otimização), como variáveis de natureza determinística, a maioria continuas e uma inteira.

### 4 Análise de sensibilidade

A análise de sensibilidade tem como principal objectivo verificar o efeito da alteração de uma variável na função objectivo, mostrando assim o quanto sensível é a função objectivo a mudanças dessa variável.

Assim, com uma análise de sensibilidade "um-decada-vez", podemos observar que a função objetivo é contínua, o que nos pode ajudar a seleccionar o método de otimização mais apropriado. Neste caso, seria possível a utilização de métodos baseados no gradiente da



**Figura 1.** gráfico dos resultados da análise de sensibilidade para diferentes variáveis da função objetivo

função, uma vez que a análise de sensibilidade mostra que a função é diferenciável.

## 5 Algoritmo de otimização

### 5.1 O algoritmo *Firefly*

A otimização deste problema pode ser feita com recurso a muitos algoritmos de otimização diferentes. O algoritmo em questão foi escolhido tendo em consideração outros dois problemas, cuja solução é desconhecida, que se quer otimizar. Estes problemas são também problemas de otimização não linear de escala intermédia, com variáveis contínuas e discretas, mas cuja função objetivo apresenta degraus, não sendo possível calcular o seu gradiente.

Assim sendo, optou-se por não utilizar métodos baseados no gradiente da função. Métodos de procura di-

reta também não são uma possibilidade, uma vez que são muito pouco eficientes em problemas com elevado grau de complexidade.

Métodos heurísticos, apesar de nem sempre convergirem com exatidão no mínimo da função, permitem encontrar o ótimo global mesmo em funções com degraus, não contínuas. Optou-se, portanto, por investigar um destes métodos.

O algoritmo *Firefly* é um método heurístico baseado na ideia de inteligência de enxames (*Swarm Intelligence*) [3], em particular, inspirado pelo comportamento de pirilampos, que, na altura de reprodução, são atraídos por pirilampos com maior brilho. O método assume que todos os pirilampos são unissexuais, de modo a que um pirilampo seja atraído por todos os outros, sendo essa atração proporcional ao seu brilho e inversamente proporcional à distância entre os pirilampos. No contexto de otimização, o brilho do pirilampo é determinado pela função objetivo [?].

As posições iniciais dos pirilampos são geradas aleatoriamente.

### 5.2 Pseudo-código do algoritmo

---

```
# Inicializacao das posicoes dos pirilampos
I = f(x)
for t in range(0, iteracao_max):
    for i in range(1, populacao):
        for j in range(1, populacao):
            if I[j] < I[i]:
                # mover pirilampo i na direcao j
                # segundo a equacao abaixo
                # Avaliar novas solucoes f(x)
                # Ordenar os pirilampos de acordo com a sua
                # luminosidade e guardar o otimo global
                # Atualizar valor de alpha
                alpha = alpha * damp
```

---

[4] [5]

$$x_i^{novo} = x_i^{ant} + \beta_0(x_j - x_i)e^{-\gamma d^2} + \alpha L \cdot \text{rand}(-1,1) \quad (2)$$

onde,

- $\beta_0$  - Atração entre pirilampos à distancia 0;
- $\gamma$  - Coeficiente de absorção;
- $d = \|x_j - x_i\|$  - Distancia entre os pirilampos;
- $\alpha$  - Fator de aleatoriedade;
- $L$  - Escala do problema;

[4] [2]

### 5.3 Penalizações externas

O algoritmo *Firefly* assume que o problema não tem restrições. Uma vez que o problema em estudo tem restri-

ções de desigualdade, é necessário transformá-lo num problema sem restrições. Isto pode ser feito aplicando penalizações à função objetivo, a qual passa a ser:

Procurar  $\mathbf{x}$  de modo a :

$$\text{minimizar } F(\mathbf{x}, r_g) = f(\mathbf{x}) + P(\mathbf{x}, r_g), \quad (3)$$

$$\text{sujeito a } x_i^{\min} \leq x_i \leq x_i^{\max}, \quad (4)$$

sendo

$$P(\mathbf{x}, r_g) = r_g \cdot \left[ \sum_{j=1}^m [\max\{0, g_j(\mathbf{x})\}]^\gamma \right] \quad (5)$$

Esta nova função aplica-se a este caso específico, em que apenas há restrições de desigualdade. Caso houvessem restrições de igualdade, teria que ser acrescentada outra parcela à função  $P$ .

## 6 Resultados

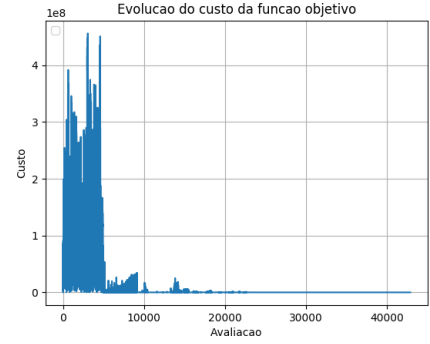
Para obter resultados fiáveis usando o algoritmo escolhido, foi necessário ajustar os parâmetros do algoritmo.

Como este algoritmo converge extremamente rápido, usamos apenas 10 iterações para uma população de 100 indivíduos o que fez um total de aproximadamente 42 000 avaliações, uma vez que em cada iteração há um número de iterações aproximadamente igual a metade do quadrado do tamanho da população.

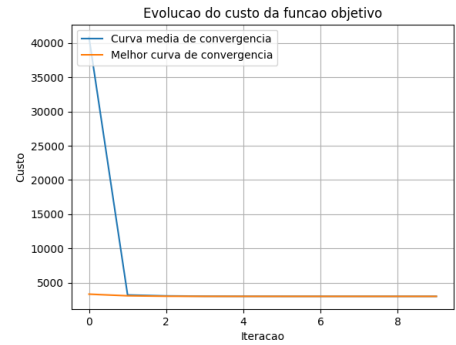
Os parâmetros escolhidos, para o melhor resultado obtido, foram:  $\gamma = 1$  de forma a convergir mais depressa, menos iterações, compensado assim o valor do  $\text{damp} = 0.4$ , cuja redução diminui também a velocidade de convergência,  $\alpha = 0.1$ , de forma a que os valores aleatórios da distribuição normal não sejam demasiado distantes dos valores calculados na iteração anterior e, por fim, temos o valor de  $\beta = 1.8$ , que, como os outros valores, foi obtidos através de vários testes.

Através do gráfico da figura 2 podemos ver que ao fim de 25000 a 30000 avaliações já temos valores relativamente próximos do valor ótimo e pode ser útil ou não fazer mais avaliações, dependendo da precisão requerida.

Com os valores obtidos em 20 ensaios, calculámos a curva média de convergência (figura 3, e desse conjunto de avaliações realizadas extraímos o melhor custo de 2996.349406 com uma posição de  $\mathbf{x} = [3.500001, 0.7, 17, 7.3, 7.8, 3.350216, 5.286684]$ , isto é mais próximo da literatura que é de 2996.348165 com a posição de  $\mathbf{x} = [3.499999, 0.7, 17, 7.3, 7.8, 3.350215, 5.286683]$  [1].



**Figura 2.** gráfico do custo em cada avaliação da função objetivo



**Figura 3.** gráfico com a média dos melhores custos em cada iteração e melhor custo global

## 7 Conclusões

Utilizando o algoritmo *Firefly* na resolução de um problema de otimização *benchmark*, foi possível avaliar o seu desempenho e exatidão de resultados. Em comparação com o melhor valor obtido noutros estudos deste problema, um custo de 2996.348165, o resultado por nós obtido apresenta um erro de apenas 0,0000416%. Trata-se de um algoritmo com uma muito boa exatidão de resultado, para este problema. Além disso, sendo um algoritmo que converge depressa, caso seja necessário reduzir o custo computacional, será possível diminuir um pouco o número de avaliações (diminuindo a população ou o número de iterações) sem comprometer demasiado o resultado final. Com este estudo, foi também possível ganhar uma maior compreensão dos parâmetros dos quais o algoritmo depende.

## Referências

1. A. Andrade-Campos, J. Dias-de-Oliveira, Benchmark 2020: Weight minimisation of a speed reducer, *Non-Linear Optimisation in Engineering / MIEM.2020*, 2020.
2. Yang, Xin-She & Deb, Suash. (2010). Eagle Strategy Using Lévy Walk and Firefly Algorithms for Stochastic Optimization. *Studies in Computational Intelligence*.
3. Chakraborty, Amrita & Kar, Arpan. (2017). Swarm Intelligence: A Review of Algorithms
4. Firefly Algorithm - an overview | ScienceDirect Topics. (2020). Retrieved 14 December 2020, from <https://www.sciencedirect.com/topics/computer-science/firefly-algorithm>
5. WiraDKP/Firefly-Algorithm. (2020). Retrieved 17 December 2020, from <https://github.com/WiraDKP/Firefly-Algorithm>