

CEFET-MG
Curso de Engenharia de Computação
Disciplina: Programação em Python

Otimização de Layout de Turbinas Eólicas

Autores: João Francisco Teles da Silva e Matheus
Emanuel da Silva
4º Período

Sumário

1	Introdução	2
2	Heurísticas e Meta-Heurísticas	2
2.1	O que são Heurísticas?	2
2.2	O que são Meta-Heurísticas?	2
3	O Algoritmo Genético	2
4	Instalação	3
4.1	Requisitos	3
5	Guia de Uso	3
6	Arquitetura do Código	3
6.1	Estrutura de Diretórios e Arquivos	3
7	Detalhe das Funções	3
7.1	Funções do Algoritmo Genético	4
8	Casos de Teste	4
9	Casos de Teste	4
9.1	Caso de Teste 1	4
9.2	Caso de Teste 2	5
10	Conclusão	5
11	Autores e Contato	5

1 Introdução

A energia eólica tem se tornado uma das fontes renováveis mais promissoras devido ao seu baixo impacto ambiental e grande disponibilidade. No entanto, a eficiência de um parque eólico depende da disposição das turbinas dentro da área disponível, pois a proximidade entre elas pode gerar interferências aerodinâmicas, reduzindo a produção de energia.

O problema abordado neste trabalho consiste em encontrar a melhor distribuição de um conjunto de turbinas dentro de um espaço delimitado, de forma a minimizar a interferência entre elas. Para isso, utilizamos um Algoritmo Genético, uma técnica de otimização inspirada na teoria da evolução.

2 Heurísticas e Meta-Heurísticas

2.1 O que são Heurísticas?

Uma **heurística** é um método de busca por soluções aproximadas para problemas complexos. Ela não garante a solução ótima, mas encontra boas soluções em tempo viável. Heurísticas são utilizadas quando a busca exata é inviável devido ao tempo computacional elevado.

2.2 O que são Meta-Heurísticas?

Meta-heurísticas são estratégias genéricas para resolver problemas de otimização utilizando heurísticas. Diferente das heurísticas convencionais, as meta-heurísticas são projetadas para escapar de mínimos locais e explorar melhor o espaço de soluções. Exemplos incluem Algoritmos Genéticos, Simulated Annealing e Otimização por Enxame de Partículas.

3 O Algoritmo Genético

O Algoritmo Genético (AG) é um método de otimização baseado na teoria da evolução de Darwin. Ele funciona através dos seguintes passos:

- **Inicialização:** Criação de uma população inicial de soluções aleatórias.
- **Avaliação:** Cálculo do desempenho de cada solução com base em uma função objetivo.
- **Seleção:** Escolha das melhores soluções para reprodução.
- **Cruzamento:** Combinação de soluções existentes para criar novas soluções.
- **Mutação:** Pequenas alterações nas soluções para manter a diversidade.
- **Substituição:** Atualização da população para a próxima geração.

O objetivo do AG é minimizar o impacto entre as turbinas, encontrado através da equação:

$$\text{impacto}(x_1, y_1, x_2, y_2) = \frac{1}{d^2}$$

onde d é a distância entre duas turbinas.

4 Instalação

4.1 Requisitos

Para rodar o código, é necessário:

- Python 3 instalado (<https://www.python.org/downloads/>).
- As bibliotecas padrão do Python: `math`, `random` e `copy`.

5 Guia de Uso

O código recebe como entrada uma matriz de interferência das turbinas e retorna:

- Um arquivo com o layout otimizado.
- A matriz de interferência final.
- A melhoria percentual obtida.

6 Arquitetura do Código

6.1 Estrutura de Diretórios e Arquivos

- **main.py** - Implementação do Algoritmo Genético.
- **randomMap.py** - Gera matrizes de interferência aleatórias.
- **input/** - Contém os arquivos de entrada.
- **output/** - Contém os arquivos de saída.

7 Detalhe das Funções

- **distancia(x1, y1, x2, y2)**: Calcula a distância entre duas turbinas usando a fórmula Euclidiana.
- **impacto(x1, y1, x2, y2)**: Calcula o impacto entre turbinas com base na distância.
- **impacto_total(positions)**: Calcula o impacto total do layout.

7.1 Funções do Algoritmo Genético

- **gerar_populacao()**: Gera uma população inicial de layouts aleatórios.
- **selecionar_torneio()**: Escolhe os melhores indivíduos para reprodução.
- **cruzamento()**: Gera novos layouts combinando dois layouts existentes.
- **mutacao()**: Aplica pequenas mudanças no layout para evitar estagnação.
- **algoritmo_genetico()**: Controla todo o processo evolutivo.

8 Casos de Teste

Abaixo estão os resultados de dois casos de teste, contendo a matriz inicial, o layout otimizado e a matriz de interferência final.

9 Casos de Teste

Abaixo estão os resultados de dois casos de teste, contendo a matriz de interferência inicial, o layout otimizado das turbinas após a execução do algoritmo genético e a matriz de interferência final.

9.1 Caso de Teste 1

Entrada - Matriz de Interferência Inicial:

```
1 5
2 0.0 0.1 0.3 0.5 0.7
3 0.1 0.0 0.6 0.3 0.4
4 0.3 0.6 0.0 0.3 0.2
5 0.5 0.3 0.3 0.0 0.1
6 0.7 0.4 0.2 0.1 0.0
```

Saída - Layout Otimizado:

```
1 Turbina 1: (x=0.62, y=0.13)
2 Turbina 2: (x=1.90, y=98.41)
3 Turbina 3: (x=98.68, y=98.76)
4 Turbina 4: (x=48.43, y=49.09)
5 Turbina 5: (x=99.58, y=1.46)
```

Saída - Matriz de Interferência Final:

```
1 0.0000 0.0001 0.0001 0.0002 0.0001
2 0.0001 0.0000 0.0001 0.0002 0.0001
3 0.0001 0.0001 0.0000 0.0002 0.0001
4 0.0002 0.0002 0.0002 0.0000 0.0002
5 0.0001 0.0001 0.0001 0.0002 0.0000
```

Melhoria percentual: 68.88%

9.2 Caso de Teste 2

Entrada - Matriz de Interferência Inicial:

```
1 5
2 0.0 1.0 0.6 0.6 0.6
3 1.0 0.0 0.6 0.8 0.5
4 0.6 0.6 0.0 0.4 0.5
5 0.6 0.8 0.4 0.0 0.6
6 0.6 0.5 0.5 0.6 0.0
```

Saída - Layout Otimizado:

```
1 Turbina 1: (x=0.48, y=3.76)
2 Turbina 2: (x=99.25, y=98.86)
3 Turbina 3: (x=50.33, y=54.08)
4 Turbina 4: (x=2.28, y=99.66)
5 Turbina 5: (x=95.58, y=1.88)
```

Saída - Matriz de Interferência Final:

```
1 0.0000 0.0001 0.0002 0.0001 0.0001
2 0.0001 0.0000 0.0002 0.0001 0.0001
3 0.0002 0.0002 0.0000 0.0002 0.0002
4 0.0001 0.0001 0.0002 0.0000 0.0001
5 0.0001 0.0001 0.0002 0.0001 0.0000
```

Melhoria percentual: 111.99%

10 Conclusão

Os resultados demonstram que o Algoritmo Genético foi capaz de reduzir significativamente a interferência entre as turbinas eólicas. A análise das matrizes de interferência finais comprova que os impactos entre as turbinas diminuíram em relação às matrizes iniciais.

A melhoria percentual alcançada foi de 68.88% no primeiro caso e 111.99% no segundo caso, indicando que a técnica aplicada foi eficaz na otimização do layout.

11 Autores e Contato

- João Francisco Teles da Silva - joaoteles0505@gmail.com
- Matheus Emanuel da Silva - memmanuel643@gmail.com