Universidade Federal
de São João del-Rei

Departamento de Ciência da Computação

Vítor Augusto Niess Soares Fonseca
Vítor Rezende Silva

Relatório: Algoritmo de Otimização da Baleia (WOA)

São João del-Rei, Setembro de 2024

Sumário

1 Introdução

O algoritmo de otimização da baleia (*Whale Optimization Algorithm* - WOA) é um algoritmo bioinspirado desenvolvido por Seyedali Mirjalili em 2016. Ele é baseado no comportamento de caça das baleias jubarte, que criam padrões de espiral ao cercar suas presas. O WOA tem sido amplamente utilizado em problemas de otimização devido à sua simplicidade e eficácia. Neste relatório, será apresentada a implementação do WOA para resolver problemas de otimização com foco em suavizar o movimento das baleias através de quatro variações da função de espiral de bolhas.

2 Metodologia

2.1 Inicialização

O algoritmo inicia com a criação de um grupo de baleias (*baleal*), onde cada baleia (*baleia*) é posicionada aleatoriamente em um espaço multidimensional delimitado pelos valores $X_{\min}$ e $X_{\max}$. A função objetivo de cada baleia é avaliada utilizando a função Ackley, uma função comumente usada para testar algoritmos de otimização.

Listing 1: Inicialização de uma baleia

```
class Baleia:
    def __init__(self, Xmin, Xmax, n, func_espiral):
        self.Xmin = Xmin
        self.Xmax = Xmax
        self.n = n
        self.func_espiral = func_espiral
        self.X = np.random.uniform(self.Xmin, self.Xmax, n)
        self.fitness = self.funcao_objetivo()
```

2.2 Função Objetivo

A função objetivo utilizada neste exemplo é a função de Ackley, uma função com várias variáveis que cria um cenário complexo de otimização com muitos ótimos locais. A função Ackley é dada pela equação:

$$f(\mathbf{x}) = -20\exp\left(-0,2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$$

Esta função avalia a qualidade da solução representada pela posição de uma baleia no espaço de busca.

2.3 Cercamento e Espiral de Bolhas

No WOA, as baleias cercam a melhor solução conhecida. Quando uma baleia está próxima de sua presa, ela utiliza um padrão de espiral para se aproximar ainda mais. Para isso, foi implementada a função *espiral_de_bolhas*, com quatro variações para suavizar o movimento, representadas por diferentes formas de calcular o deslocamento espiral:

- **Variação 1**: Usa a função seno para suavizar o movimento do espiral.

- **Variação 2**: Usa uma função logarítmica para controlar o movimento ao longo do tempo.
- **Variação 3**: Cria uma espiral elíptica com escalas diferentes para cada dimensão.
- **Variação 4**: Introduz um fator de amortecimento exponencial.

Listing 2: Funções de espiral de bolhas

```
def espiral_1(X, melhor_baleia, D):
    l = np.random.uniform(-1, 1)
    return D * np.exp(l) * np.sin(2 * np.pi * l) + melhor_baleia

def espiral_2(X, melhor_baleia, D):
    l = np.random.uniform(-1, 1)
    return D * np.exp(l) * np.cos(2 * np.pi * l) * np.log(1 + np.abs(l)
```

2.4 Atualização da Posição

A cada iteração, as baleias atualizam suas posições. Dependendo do parâmetro p, que controla a probabilidade de explorar novas regiões ou refinar a solução atual, as baleias escolhem entre cercar a melhor baleia conhecida ou seguir o padrão de espiral de bolhas.

Listing 3: Atualização da posição da baleia

```
def atualizar_posicao(self, melhor_baleia, A, C, p, Xrand=None):
    D = np.abs(C * melhor_baleia - self.X)
    if p < 0.5:
        if np.all(np.abs(A) < 1):
            self.cercamento(melhor_baleia, A, D)
        else:
            if Xrand is not None:
                self.buscar_novas_presas(Xrand, A, D)
    else:
        self.espiral_de_bolhas(melhor_baleia, D)
```

2.5 Execução do Algoritmo

O algoritmo segue um ciclo iterativo no qual todas as baleias atualizam suas posições, e a melhor baleia é monitorada a cada iteração. Ao final, o algoritmo retorna a melhor solução encontrada e gera um GIF mostrando a evolução das posições das baleias com cores distintas para cada variação da função de espiral.

3 Resultados

O algoritmo foi executado com 500 baleias e 25 iterações, utilizando as quatro variações da função de espiral. O GIF gerado mostra a evolução das baleias no espaço de busca, com cada grupo de baleias colorido de maneira distinta, representando as diferentes variações. O GIF mencionado está juntamente com o código do algoritmo, no diretório enviado.

4 Considerações Finais

O WOA é um algoritmo eficaz para problemas de otimização contínua, sendo capaz de encontrar boas soluções rapidamente. As diferentes variações da função de espiral de bolhas contribuem para suavizar o movimento das baleias e melhorar a visualização da convergência no espaço de busca.