

**INTELIGÊNCIA ARTIFICIAL**

**2024/2025**

**Professor**

**Paulo Oliveira**

**Eduardo Pires**

**RELATÓRIO**

Métodos de pesquisa

trabalho pratico 2 (Parte a)

**79881 - David Fidalgo**

**78800 - Tiago Carvalho**

**2024/2025**

Indíce

[Indíce 2](#_Toc183723251)

[Indíce de Figuras 3](#_Toc183723252)

[1. Resumo 4](#_Toc183723253)

[2. Introdução 5](#_Toc183723254)

[3. Enquadramento Teórico 6](#_Toc183723255)

[3.1. Algoritmo da Subida da Colina 6](#_Toc183723256)

[3.2. Algoritmo Simulated Annealing 8](#_Toc183723257)

Indíce de Figuras

[Figura 1. Hill Climb Algorithm: Gráfico exemplo 7](#_Toc183723244)

[Figura 2-> Simulated Annealing Algorithm: Gráfico Exemplo 8](#_Toc183723245)

# Resumo

# Introdução

# Enquadramento Teórico

## 3.1. Algoritmo da Subida da Colina

O algoritmo da **subida da colina** (ou **Hill Climbing**) é um método heurístico de pesquisa usado em problemas de otimização matemática na Inteligência Artificial. O objetivo passa por melhorar continuamente uma única solução por meio de pequenas alterações em cada iteração. Em cada passo, o algoritmo verifica se a nova solução é melhor que a anterior.

O processo começa com uma solução inicial, que é aperfeiçoada a cada iteração. Cada mudança é avaliada por uma função heurística que determina a qualidade da solução. O algoritmo continua até encontrar um máximo local, ou seja, um ponto em que não haja mais melhorias possíveis.

Este algoritmo também é conhecido como um **Algoritmo Ganancioso** (ou **Greedy Local Search**) devido ao facto de escolher sempre as soluções que parecem ser as melhores no momento, sem considerar o impacto futuro dessas escolhas.

**Tipos de Hill Climbing Algorithm**:

* Hill Climbing Simple – Faz alterações na solução atual e aceita a alteração, mas pode ficar preso em máximos locais.
* Hill Climbing Multiple Restart – Faz múltiplas reinicializações em busca de uma solução melhor, aumentado a probabilidade de encontrar o máximo global.

**Variantes a ter em conta no Hill Climbing Algorithm**:

* Solução inicial – Indica o ponto de partida da busca, escolhida de uma forma aleatória ou baseado em critérios heurísticos;
* Função de Avaliação – É o que avalia se uma solução está mais próxima de uma solução “ótima”;
* Vizinhança – Conjunto de soluções próximas à solução atual que podem ser alcançadas através de pequenas alterações;
* Critério de Aceitação – Define quando uma solução vizinha deve ser aceita. Em um problema de maximização, é aceite o vizinho se tiver a solução maior que a solução atual;
* Número de Iterações – Número de vezes que o algoritmo examina as soluções vizinhas;

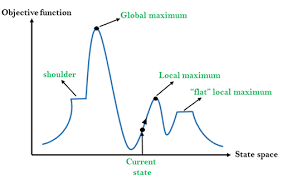


Figura 1. Hill Climb Algorithm: Gráfico exemplo

**Explicação passo-a-passo:**

* Inicialmente, é gerado uma solução aleatória ou então uma solução escolhida de forma heurística;
* Inicia um ciclo que será executado até que a condição imposta seja contrariada;
* Escolhe um dos vizinhos da solução na posição atual de forma aleatória;
* Verifica se o valor do vizinho é maior, menor ou igual à solução atual. Se tivermos perante um problema de maximização, o algoritmo vai verificar se o valor é maior que a solução atual, aceitando o vizinho se a condição se verificar;

## 3.2. Algoritmo Simulated Annealing

O **Algoritmo Simulated Annealing** (SA) é um método heurístico inspirado no processo de arrefecimento de metais. Na otimização, este método tenta encontrar soluções globais, evitando que o algoritmo fique preso em máximos/mínimos locais.

Diferente do **Hill Climbing Algorithm**, o **Simulated Annealing** poderá aceitar soluções “piores” durante a sua execução, permitindo assim uma exploração mais ampla do espaço de pesquisa. "Embora a simplicidade muitas vezes leve a soluções elegantes, a complexidade pode ser necessária para capturar a totalidade de um problema." (Knuth, D. 1997).

**Variantes a ter em conta no Simulated Annealing**:

* Solução Atual – Solução corrente na busca;
* Função de Custo – Mede a qualidade da solução;
* Temperatura – Controla a probabilidade de aceitar soluções piores;
* Fator de Arrefecimento (FA, 0<FA<1) – Taxa usada para reduzir a temperatura (T\_new=T\_atual\*FA);
* Vizinhança – Soluções próximas à atual, geradas por ligeiras alterações;
* Diferença de Energia (ΔE) – Diferença entre o valor do custo da solução vizinha e da solução atual;
* Critério de Paragem – Número máximo de iterações ou temperatura mínima atingida.

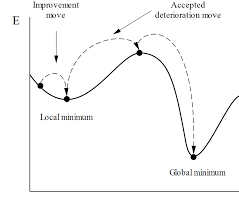


Figura 2-> Simulated Annealing Algorithm: Gráfico Exemplo

**Explicação passo-a-passo:**

* É gerado uma solução inicial aleatória ou heurística;
* É definida uma Temperatura inicial elevada para aceitar as soluções piores inicialmente;
* É gerado uma nova solução vizinha, realizando uma pequena perturbação na solução atual;
* Critério de aceitação:
  + Se a solução vizinha é melhor, ela é aceita;
  + Se é pior, poderá ser aceita pela seguinte função probabilística:

**P=e^-(ΔE/T\_atual)​**

* Arrefecimento – Reduz a temperatura de acordo com a taxa aplicada para o efeito;
* Critério de paragem – O SA para quando a temperatura atingiu um valor próximo de zero ou um número máximo de iterações é atingido.