

Relatório do Laboratório 3 - Otimização com Métodos de Busca Local

1 Breve Explicação em Alto Nível da Implementação

O laboratório realizado teve por objetivo obter parâmetros físicos relativos ao movimento de uma bola através de métodos de otimização em busca local. Como é um modelo simples, o método dos mínimos quadrados pode ser aplicado para resolver o sistema, o que permite comparar com os métodos mais complexos. O modelo é apresentado na equação 1. Logo, a função custo J de é dada pela equação 2 para cada uma das observações k , de forma que o vetor de parâmetros a ser otimizado é o vetor $\theta = [v_0 \ f]^T$.

$$v(t) = v_0 - ft \quad (1)$$

$$J([v_0, f]) = \sum_{k=1}^n (v_0 + ft[k] - v[k])^2 \quad (2)$$

1.1 Descida do Gradiente

O primeiro método utilizado para otimizar os parâmetros foi a Descida do Gradiente. Esse método, muito utilizado para treinar redes neurais, funciona bastante bem caso seja possível calcular o gradiente do custo analiticamente, que é o caso do modelo que foi trabalhado nesse relatório.

O método se resume em, a cada iteração, atualizar o vetor de parâmetros diminuindo de uma taxa que equivale ao gradiente da função custo, demonstrado na equação 3. Assim, a lógica do método é seguir na direção de máximo decrescimento da função custo.

$$\theta_{k+1} = \theta_k - \alpha \frac{\partial J(\theta)}{\partial \theta} \quad (3)$$

A implementação do método é feita de forma a diminuir do custo a cada iteração, ou seja, fazer com que o valor de $|J_{k+1} - J_k|$ tenda a zero. Logo, a cada iteração, um valor novo de θ é computado através da equação 3. O programa irá iterar essa lógica enquanto o valor de $|J_{k+1} - J_k|$ seja menor que um valor *threshold* pré-definido ϵ ou enquanto o número de iterações for menor que um valor pré-definido.

1.2 Hill Climbing

O segundo método utilizado foi o *Hill Climbing*. Esse método é bastante utilizado para espaços discretos ou para modelos que não são computáveis, ou seja, em que os gradientes não podem ser calculados facilmente.

O método funciona iterando continuamente até que uma condição de parada seja satisfeita, que nesse caso é uma condição de número de iterações máximo pré-definido. Enquanto ela não for satisfeita, o código explora, a partir de um estado inicial, estados vizinhos ao redor dentro de um raio pré-definido Δ . Para a implementação aqui realizada, foram explorados os vizinhos em uma estratégia 8-conectada. Para todos os vizinhos é então computado sua função custo e comparada com o nó com menor custo até o momento. Se algum dos vizinhos tiver um custo menor que o melhor até o momento, esse passa a ser o melhor e se explora então os vizinhos desse, repetindo o ciclo.

1.3 *Simulated Annealing*

Esse método, baseado no processo de têmpera da metalurgia, tem objetivo solucionar um problema comum dos métodos de busca local - os mínimos locais. Para atacar esse problema, esse método opera como o *Hill Climbing* mas permite a transição para estados piores com o objetivo de fugir de caminhos enviesados de mínimos locais. O método permite que iterações mais novas possuam maior probabilidade de ir para estados piores.

Para modelar isso, utiliza-se uma equação que define o grau de proximidade com o objetivo. Para o problema simulado aqui, utilizou-se a equação 4, em que T_0 e β são hiperparâmetros e i é a iteração atual. Com essa equação, calcula-se T , que podemos utilizar para então dar um limite para a nossa escolha aleatória r (um valor aleatório de 0 a 1). Se o vizinho de um nó explorado tiver um custo menor nós o exploramos, assim como no *Hill Climbing*. Entretanto, se for pior e o valor de r satisfazer a equação 5, nós também o exploramos.

$$T = \frac{T_0}{1 + \beta i^2} \quad (4)$$

$$r \geq e^{\frac{(J_{k+1} - J_k)}{T}} \quad (5)$$

2 Figuras Comprovando Funcionamento do Código

2.1 Descida do Gradiente

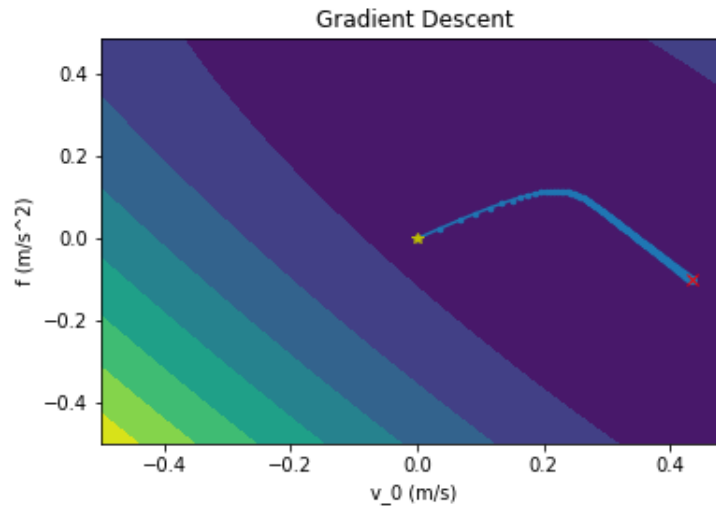


Figura 1: Resolução utilizando o método de Descida do Gradiente.

2.2 *Hill Climbing*

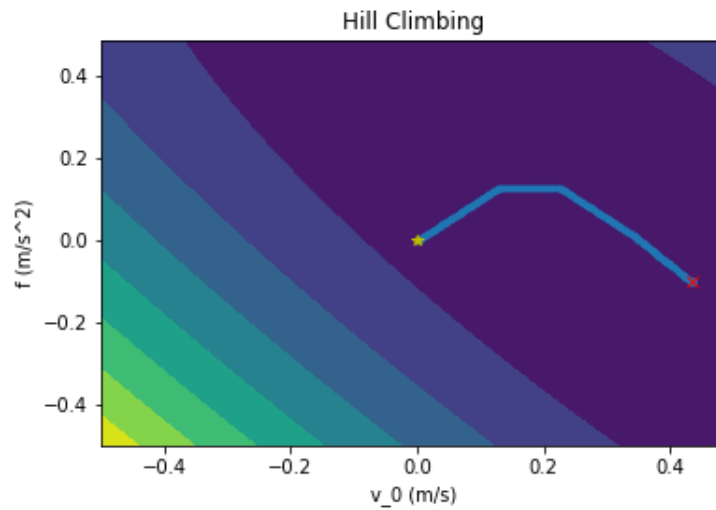


Figura 2: Resolução utilizando o método de *Hill Climbing*.

2.3 *Simulated Annealing*

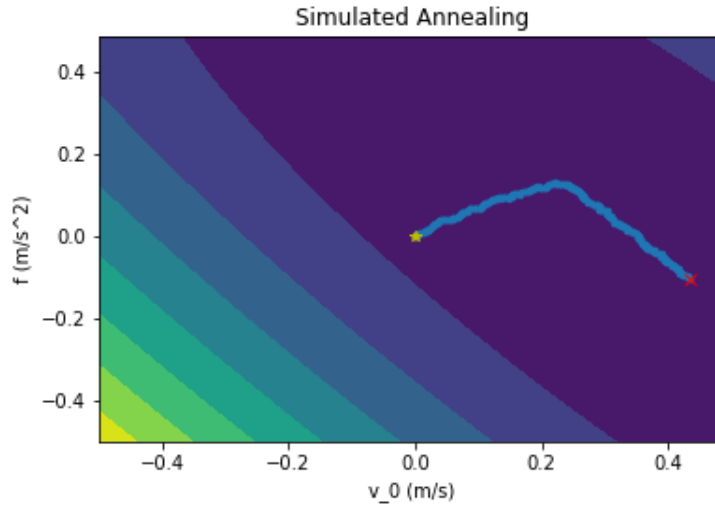


Figura 3: Resolução utilizando o método de *Simulated Annealing*.

3 Comparação entre os métodos

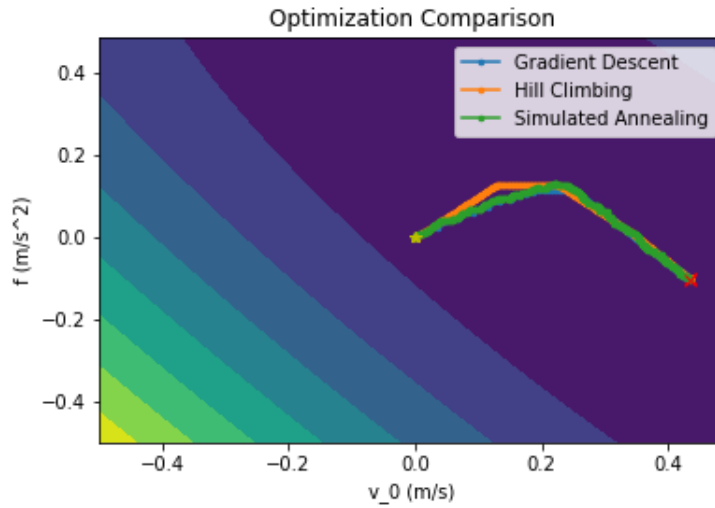


Figura 4: Comparação dos métodos iterativos utilizados.

Tabela 1 com a comparação dos parâmetros da regressão linear obtidos pelos métodos de otimização.

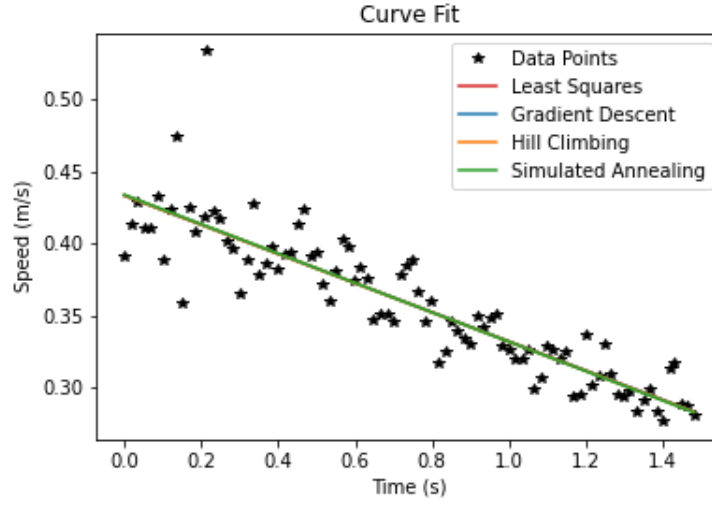


Figura 5: Comparação do *fit* da curva dos dados a partir dos diferentes métodos.

Tabela 1: parâmetros da regressão linear obtidos pelos métodos de otimização.

Método	v_0	f
MMQ	0.433373	-0.101021
Descida do gradiente	0.433371	-0.101019
<i>Hill climbing</i>	0.433411	-0.101196
<i>Simulated annealing</i>	0.433939	-0.101821