Universidade Federal de Minas Gerais
Escola de Engenharia
Departamento de Engenharia Elétrica
ELE077 Otimização Não Linear

Trabalho Computacional I

1 Problema 1

Imagine que você está controlando a temperatura de um forno industrial. O objetivo é que a temperatura atinja um valor de referência (T_{ref}) e permaneça estável nesse ponto, independentemente de variações externas. Para isso, usamos um controlador PID que ajusta o sinal de controle $u(t)$ com base no erro entre a temperatura atual $T(t)$ e a temperatura de referência. O controlador PID é definido pela seguinte equação:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \tag{1}$$

onde:

- $e(t) = T_{\text{ref}} - T(t)$ é o erro entre a temperatura de referência e a temperatura atual.
- K_p, K_i, e K_d são os ganhos proporcional, integral e derivativo, respectivamente.
- $u(t)$ é o sinal de controle que será aplicado para ajustar o sistema (por exemplo, potência aplicada ao aquecedor).

O objetivo é ajustar os parâmetros K_p, K_i, e K_d de forma a minimizar uma métrica de desempenho, como o tempo de estabilização, o erro quadrático médio (MSE), ou o tempo de resposta. Um dos critérios comuns é minimizar o erro quadrático médio ao longo de um período de tempo T_{sim}, que pode ser expresso como:

$$J(K_p, K_i, K_d) = \int_0^{T_{\text{sim}}} e(t)^2 dt \tag{2}$$

A função-objetivo para o problema de otimização seria minimizar o erro quadrático médio $J(K_p, K_i, K_d)$, o que pode ser formulado como:

$$\min_{K_p, K_i, K_d} \int_0^{T_{\text{sim}}} \left(T_{\text{ref}} - T(t, K_p, K_i, K_d)\right)^2 dt \tag{3}$$

sendo que a temperatura $T(t, K_p, K_i, K_d)$ é obtida a partir da solução de um modelo matemático que descreve a dinâmica do sistema. Para simplificar, considere que o modelo é dado por uma equação diferencial ordinária (EDO) de primeira ordem:

$$\frac{dT}{dt} = -\frac{T}{\tau} + \frac{K}{\tau} u(t) \tag{4}$$

onde τ é a constante de tempo do sistema e K é uma constante de proporcionalidade. Portanto, o objetivo é encontrar os parâmetros K_p, K_i, e K_d que minimizam o erro quadrático médio $J(K_p, K_i, K_d)$ ao longo de um período de simulação T_{sim}.

Para resolver o problema, considere os seguintes parâmetros:

- $T_{\text{ref}} = 100^o$C (temperatura de referência).
- $T(0) = 25^o$C (temperatura inicial).
- $K_p = 1$, $K_i = 0.1$, $K_d = 0.01$ (valores iniciais dos ganhos).
- $\tau = 1$ (constante de tempo do sistema).
- $T_{\text{sim}} = 10$ segundos (tempo de simulação).
- $K = 1$ (constante de proporcionalidade).

Para resolver o problema, você pode aproveitar a implementação da função-objetivo disponível no arquivo `questao1.py`. Você vai ter que implementar a chamada do algoritmo de otimização. Escolha o algoritmo mais adequado para resolver o problema baseado em suas características, justifique a sua escolha e apresente os resultados da otimização.

2 Problema 2

A regressão logística é um modelo de classificação binária que estima a probabilidade de um evento ocorrer. Ela pode ser usada em diferentes aplicações, como diagnóstico médico, previsão de vendas, ou análise de risco. Por exemplo, suponha você tenha um conjunto de m músicas e cada música é descrita por um conjunto de n características (por exemplo, gênero, duração, popularidade, etc.). Para cada uma dessas músicas, você tem um rótulo binário indicando se a música foi um sucesso ou não. O objetivo é prever se uma música será um sucesso ou não com base nessas características.

De uma maneira básica, o modelo recebe um conjunto de dados $(\mathbf{x}_i, y_i)$ onde:

- $\mathbf{x}_i \in \mathbb{R}^n$ é o vetor de entrada com cada uma das n características da i-ésima música.
- $y_i \in \{0, 1\}$ é o rótulo da i-ésima música, i.e., se ela foi um sucesso (1) ou não (0).

O modelo, ao receber esse conjunto de dados, determina um conjunto de pesos $\mathbf{w} \in \mathbb{R}^n$ que melhor se ajusta aos dados. Determinar esses pesos é um problema de otimização, onde o objetivo pode ser definido como minimizar a função de custo de entropia cruzada regularizada:

$$J(\mathbf{w}) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(h(\mathbf{x}_i; \mathbf{w})) + (1 - y_i) \log(1 - h(\mathbf{x}_i; \mathbf{w}))] + \frac{\lambda}{2} \|\mathbf{w}\|^2 \qquad (5)$$

onde:

- $h(\mathbf{x}_i; \mathbf{w}) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}_i}}$ é a saída do modelo logístico (função sigmoide).
- m é o número de amostras.
- λ é o parâmetro de regularização para evitar *overfitting*[1].
- $\|\mathbf{w}\|^2$ é o termo de regularização $L2$ que penaliza grandes valores de $\mathbf{w}$.

Portanto, o objetivo é encontrar o vetor de pesos $\mathbf{w}$ que minimiza a função de custo $J(\mathbf{w})$. Para esta questão, é fornecido um conjunto de dados de músicas com $m = 10000$ amostras e $n = 500$ características. Você pode aproveitar a implementação da função-objetivo disponível no arquivo `questao2.py`. Você vai ter que implementar a chamada do algoritmo de otimização. Escolha o algoritmo mais adequado para resolver o problema baseado em suas características, justifique a sua escolha e apresente os resultados da otimização.

3 Problema 3

Uma empresa deseja ajustar os parâmetros de uma campanha promocional para maximizar o lucro. Os parâmetros da campanha são: o desconto (d), o tempo de duração da promoção (t) e o orçamento em marketing (m). O lucro da empresa ($L(d, t, m)$) é igual a receita ($R(d, t, m)$) menos os custos ($C(d, t, m)$). A receita é dada por:

$$R(d, t, b) = V_B \cdot (1 + f_1(d) + f_2(t)) \cdot \log(1 + b) \tag{6}$$

onde:

- V_B é o número de vendas sem promoção.
- $f_1(d) = -0.005d^2 + 0.2d$ é o incremento percentual nas vendas devido ao desconto.
- $f_2(t) = 0.05t$ é o incremento nas vendas proporcional ao tempo da promoção.

O custo é dado por:

$$C(d, t, b) = C_B + C_M(b) + P(d, t, m) \tag{7}$$

onde:

- C_B é o custo fixo inicial.
- $C_M(m) = m$ é o custo de marketing.
- $P(d, t)$ é uma penalização não contínua:
  - Penalização de R\$5000 se $d > 30\%$ (desconto muito agressivo).

[1] *Overfitting* é um efeito que ocorre quando um modelo aprende os dados de treinamento tão bem, incluindo ruídos e variações aleatórias presentes neles, que ele perde a capacidade de generalizar para novos dados, ou seja, dados que não foram usados durante o treinamento.

- Penalização de R$2000 se $t > 15$ dias.

É importante frisar que:

- O desconto não pode ser menor que 0% nem maior que 50% ($0 \leq d \leq 50$).
- O tempo de duração da promoção não pode ser menor que 1 dia nem maior que 30 dias ($1 \leq t \leq 30$).
- O orçamento de marketing não pode ser menor que R$1000 nem maior que R$50000 ($1000 \leq b \leq 50000$).

Para implementar essas restrições você pode acrescentá-las às penalizações. Por exemplo, se qualquer uma dessas variáveis ultrapassar os limites, você pode adicionar somar um valor muito alto, por exemplo, 1.000.000.

Portanto, o problema geral de otimização pode ser definido como:

$$\max_{d,t,b} L(d,t,b) = R(d,t,b) - C(d,t,b) \tag{8}$$

Para este problema, você pode aproveitar a pré-implementação da função-objetivo disponível no arquivo `questao3.py`. Além de ter que implementar a chamada do algoritmo de otimização, você vai precisar implementar os cálculos das receitas, custos, $f_1(d)$, $f_2(t)$ e $P(d,t)$. Escolha o algoritmo mais adequado para resolver o problema, justifique a sua escolha e apresente os resultados da otimização.

4 Problema 4

Considere a aleta unidimensional mostrada na Fig. 1. As variáveis x_1 e x_2 são as temperaturas em graus Celsius dos pontos 1 e 2, respectivamente. Em regime permanente, as temperaturas x_1 e x_2 são tais que minimizam a seguinte função:

$$f(x_1, x_2) = 0.6382x_1^2 + 0.3191x_2^2 - 0.2809x_1x_2 - 67.906x_1 - 14.29x_2 \tag{9}$$

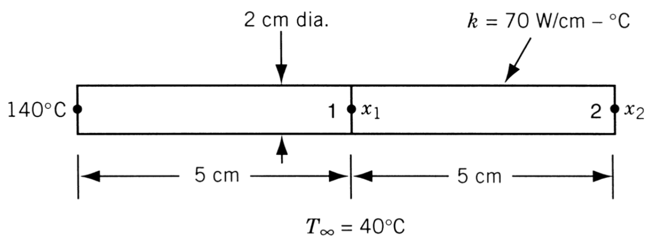

Figura 1: Aleta unidimensional.

Determine as temperaturas de regime permanente (x_1, x_2) minimizando a função da equação (9). Para este problema, você terá que implementar a função-objetivo e a chamada do algoritmo de otimização. Escolha o algoritmo mais adequado para resolver o problema, justifique a sua escolha e apresente os resultados da otimização.

5 Observações finais

- O relatório com a discussão dos experimentos e conclusões, bem como os códigos fonte dos métodos implementados, deverão ser enviados ao professor via Moodle (trabalhos enviados por e-mail não serão considerados). Templates para a escrita do relatório estão disponíveis na página da disciplina, mas não são obrigatórios.

- Espera-se um relatório discutindo as abordagens estudadas e sua aplicação a problemas de otimização. Relatórios contendo apenas códigos e/ou figuras não serão avaliados.

- Vocês podem utilizar a linguagem de programação que preferir. Minha sugestão: Python ou MATLAB (preferencialmente Python).

- Se vocês optarem pelo Python, vocês podem utilizar a biblioteca *scipy.optimize* que já conta com as implementações dos métodos vistos em sala de aula. Também é possível utilizar a biblioteca do professor, chamada `otimo.py`.

- Se vocês optarem pelo MATLAB, vocês podem utilizar a *Optimization toolbox*.