

# Princípio do Mínimo da Energia Potencial Total aplicado a deformação de uma estrutura treliçada do tipo cantilever

Luis Vinicius Costa Silva

26 de Junho de 2019

Disciplina: Otimização Clássica  
Professor: Romes Antônio Borges

## 1 Introdução

Este trabalho tem como objetivo criar um modelo matemático que permita representar a deformação de uma estrutura dada sua configuração inicial. Foi utilizada uma abordagem da otimização clássica que se baseia no Princípio do mínimo da Energia Potencial.

Para tanto, fez-se a modelagem de uma viga treliçada do tipo cantilever, e a obtenção das equações constitutivas do sistema e a escrita do problema em forma de problema de otimização. Posteriormente foi escrito um código computacional (em linguagem Python) para a resolução computacional do problema, isto é: dada uma configuração inicial da estrutura, minimizar a Função de Energia Potencial Total do sistema, obtendo os valores das variáveis de estados (configuração final da estrutura) associada a energia potencial mínima do sistema.

Foram utilizados três algoritmos clássicos para a resolução do problema de otimização: L-BFGS-B, SLSQP e Newton Truncado. Além disso foi utilizado a Evolução Diferencial como algoritmo evolutivo para a resolução do problema. Os resultados obtidos por tais algoritmos foram comparados em um estudo de caso, avaliando as vantagens e desvantagens de cada um deles para o problema em questão.

## 2 Modelagem

### 2.1 Princípio do mínimo da Energia Potencial Total

É sabido que a Energia Potencial Total de um sistema é dado pela diferença entre as forças internas (strain Energy/deformação) e o trabalho das Forças

Externas.

$$\pi = U - W \quad (1)$$

Expandindo os termos da equação anterior, temos que:

$$\pi = \sum_{e=1}^n \Lambda^{(e)} - \sum_{i=1}^m F_i u_i \quad (2)$$

A energia potencial mínima do sistema pode ser obtida igualando a derivada da energia potencial pelo deslocamento a zero:

$$\frac{\partial}{\partial u_i} \sum_{e=1}^n \Lambda^{(e)} - \frac{\partial}{\partial u_i} \sum_{i=1}^m F_i u_i = 0, \quad i = 1, 2, \dots, n \quad (3)$$

Visto que tal diferenciação torna-se muito complicada para sistemas com vários  $u_i$  (i.e: pontos de deslocamento), faz-se necessário a formulação do problema no formato de minimização, a fim de que algoritmos de otimização atinjam o resultado esperado.

## 2.2 Modelagem da treliça simplesmente engastada

Energia potencial de uma mola entre dois pontos:

$$\frac{1}{2} k (|p_1 - p_2| - r)^2 \quad (4)$$

Energia potencial total de uma massa:

$$E_{pg} = mgh \quad (5)$$

Logo, têm-se a seguinte função objetivo:

$$\begin{aligned} \min \quad U &= \sum_{i=1}^n mgy_i + \frac{1}{2} \sum_{i,j=1}^n ka_{ij} (|p_i - p_j| - l_{ij})^2 \quad s.t : \\ p_1 &= (0.0, 0.0) \\ p_2 &= (0.0, -0.1) \\ p_i - 1.0 &\leq p_i \leq p_i + 1.0, \quad i = 3, \dots, n, \quad (opcional) \end{aligned} \quad (6)$$

- $n$  – quantidade de nós da malha;
- $m$  – massa sobre a estrutura ( $kg$ );
- $g$  – constante gravitacional ( $m/s$ );
- $y_i$  – coordenada  $y$  do nó;
- $k$  – constante elástica das molas ( $kg/m$ );

- $p_i$  – i-ésima posição (x,y) do nó;
- $a_{ij}$  – elemento da matriz de adjacência (1 caso exista mola, 0 senão);
- $l_{ij}$  – comprimento relaxado da mola (i,j) (0 caso não exista mola entre nós);

### 3 Algoritmos utilizados

#### 3.1 L-BFGS-B

O algoritmo L-BFGS-B (Byrd et al., 1995) é uma modificação do algoritmo BFGS capaz de lidar com limites laterais (i.e:  $a \leq x \leq b$ ) diretamente. Além disso, este algoritmo não armazena uma aproximação densa da inversa da matriz hessiana, ao invés disso, o algoritmo armazena apenas alguns vetores que representam tal termo implicitamente, i.e: o algoritmo armazena as últimas  $m$  atualizações de  $x$  e  $\nabla f(x)$ . Estes termos são utilizados quando algumas operações exigem o produto vetorial da matriz  $H_k$ .

O tratamento dos limites laterais é realizado através do método de gradiente que identifica o conjunto com os limites das variáveis de entrada que estão ativas, e então o modelo quadrático é aproximadamente minimizado, em relação as variáveis livres.

Após isso, o próximo ponto de avaliação é calculado através da busca unidimensional, utilizando um determinado tamanho de passo  $\alpha_{l-bfgs-b}$  (hiperparâmetro).

#### 3.2 SLSQP

O SLSQP (Sequential Least Squares Quadratic Programming) é um algoritmo de programação sequencial de mínimos quadrados. O SLSQP usa o algoritmo BFGS para aproximar a Hessiana e a função de mérito  $L$  é usada para uma busca unidimensional.

As restrições de desigualdade e igualdade são combinadas e expressadas por  $\vec{h}_k$ , o vetor de multiplicadores de Lagrange é dado por  $u_k$ . As restrições de desigualdade são igualadas a zero e tratadas de acordo (entretanto os multiplicadores de Lagrange associados a estas restrições não devem ser negativos), restrições de desigualdade inativas são ignoradas.

Existem 3 critérios de parada para o algoritmo SLSQP, são eles:

$$|\nabla f_k^T d_k| + \sum |u_i| |h_i(x_k)| \leq tol \quad (7)$$

Equation 7: Condição de Parada 1

$$|f(x)_k - f(x)_{k+1}| \leq tol \quad (8)$$

Equation 8: Condição de Parada 2

$$k \geq \max it \quad (9)$$

Equation 9: Condição de Parada 3

Como pode ser visto pela equação ?? O tamanho do primeiro passo da busca unidimensional geralmente é 1, este tamanho de passo é reduzido sucessivamente por um determinado fator de contração. Logo, o primeiro termo  $|\nabla f_k^T d_k|$  representa o quanto de melhoria a função objetivo obteve ao longo da direção de descida mais íngreme com o maior comprimento de passo. O segundo termo representa uma combinação ponderada do total de todas as violações. Portanto, este critério denota que, uma vez que ocorreu uma melhoria significativa da função Lagrangeana  $L$  menor que uma tolerância pré-definida, então considera-se que o algoritmo convergiu para um  $x^*$ .

O critério de parada ?? e ?? são amplamente conhecidos, e denotam respectivamente a parada do algoritmo em função da diferença entre dois  $x_k$  de iterações consecutivas ser menor que uma tolerância pré-definida, e a parada do algoritmo após um número máximo de iterações.

A função de mérito  $L_1$  (equação 10) é utilizada para garantir uma convergência global, o que significa que o  $x$  ótimo será encontrado, independentemente do  $x$  inicial. A função de mérito pode ser considerada como uma função objetivo para a busca unidimensional, esta é decrementada sucessivamente para a geração de um novo  $x_k$  no decorrer do processo. Para problemas irrestritos a função objetivo original pode ser usada como função de mérito, em qualquer outro caso, a função de mérito  $L_1$  assume a seguinte forma:

$$L_1 = f(x_k + \alpha d_k) + \sum_{j=1}^m \rho_j |h_j(x_k + \alpha d_k)| \quad (10)$$

Equation 10: Função de mérito L1

O fluxograma abaixo demonstra o funcionamento básico do algoritmo SLSQP:

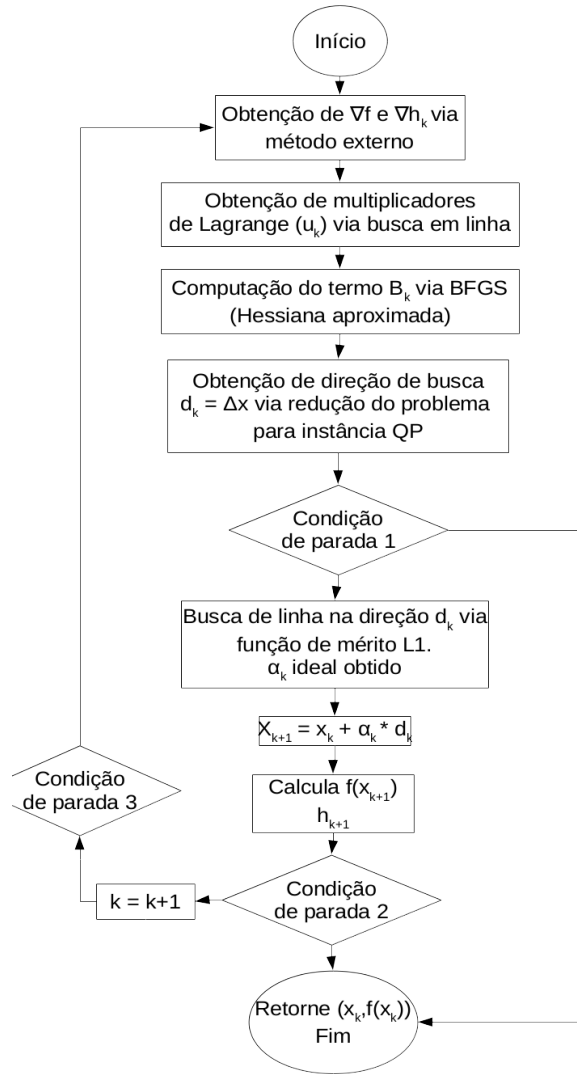


Figura 1: Fluxograma do algoritmo SLSQP

### 3.3 Método de Newton Truncado

O método de Newton truncado é um algoritmo de otimização que, dado um  $x_k$ , aproxima um  $x^*$  resolvendo as equações de Newton (...) usando algum algoritmo iterativo com condicionamento  $M^k$ .  $M^k$  é escolhido com informação de algum outro algoritmo de otimização (geralmente gradiente conjugado) ou com base em iterações anteriores. Usando a solução obtida para as equações de Newton, uma direção de descida é computada, e uma nova aproximação  $x_{k+1}$  de  $x^*$  é obtida. As equações de Newton são definidas da seguinte forma:

Considerando que a função objetivo é aproximada por uma série de Taylor, temos:

$$F(x^{(0)} + p) = F(x^{(0)}) + p^T \nabla F(x^{(0)}) + \frac{1}{2} p^T \nabla^2 F(x^{(0)}) p + R_3(x^{(0)}, p) \quad (11)$$

onde  $R_3(x^{(0)}, p)$  representam os termos de alta ordem da série. Logo, as equações de Newton são dadas pela fórmula abaixo:

$$Hp = -g \quad (12)$$

Onde  $H$  é a matriz Hessiana,  $g$  é o gradiente de  $f$  e  $p$  é o ponto de interesse da série.

### 3.4 Evolução Diferencial

O algoritmo de Evolução Diferencial é um algoritmo de otimização simples e eficiente que foi proposto por Rainer Storn e Kenneth Price em 1995 (Storn e Price, 1995). Mostra-se eficaz para funções objetivo que não são diferenciáveis ou convexas e tem facilidade na busca do ótimo com populações pequenas (Cheng e Hwang, 2001). O DE pode ser descrito como uma manipulação de indivíduos que representam as soluções candidatas. No decorrer das gerações, essas soluções candidatas sofrem modificações de mutação e cruzamento, onde são geradas novas soluções candidatas, e logo após é feita a seleção e o ciclo se repete. A figura 2 demonstra o funcionamento básico do algoritmo de Evolução Diferencial

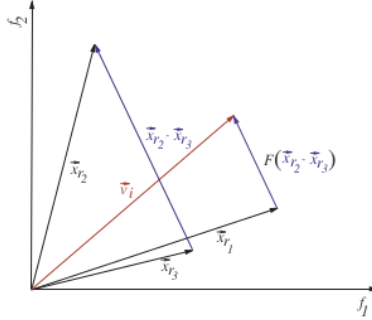


Figura 2: Funcionamento da Evolução Diferencial – Price et al. (2005)

Partindo de três vetores  $\vec{x}_{r1}$ ,  $\vec{x}_{r2}$  e  $\vec{x}_{r3}$ , são escolhidos aleatoriamente dois deles (nesse caso  $\vec{x}_{r2}$  e  $\vec{x}_{r3}$ ), sendo realizada a subtração dos mesmos. O resultado é multiplicado por um escalar  $F$ , gerando assim um vetor com módulo diferente da subtração original. Esse novo vetor é então somado ao vetor  $\vec{x}_{r1}$ , fornecendo assim um novo vetor  $\vec{v}_i$ . Esse vetor  $\vec{v}_i$  indicará uma nova posição no espaço, ou um novo indivíduo (muito semelhante a um algoritmo genético, mas com cromossomos compostos por valores reais).

## 4 Resultados e Conclusões

### 4.1 Métodos Clássicos

Parâmetros do experimento 1:

- massa –  $m = 0.1$ ;
- número de nós –  $n = 20$ ;
- distância inicial entre nós –  $e = 0.1$ ;
- comprimento das molas em situações sem deformação –  $l = 0.1$ ;
- constante de rigidez da mola  $k = 10000$

Configuração inicial da treliça:

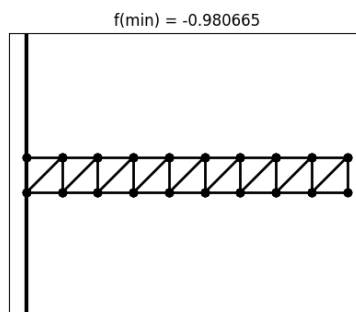


Figura 3: Representação gráfica da configuração inicial da treliça

Métodos x Métricas	Qtde de variáveis	Qtde de Iterações	Avaliações da função objetivo	Min F(x)
L-BFGS-B	40 (20 nós)	133	138	-2.52938265048
SLSQP	40 (20 nós)	58	2557	-2.529382692846
Newton Truncado	40 (20 nós)	830	62	-2.51249006635

Tabela 1: Resultados obtidos no experimento 1

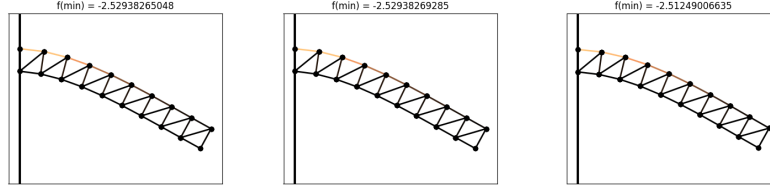


Figura 4: Configuração Final da treliça computada pelo Algoritmo L-BFGS-B, SLSQP e Newton Truncado respectivamente

x	y
0.0000000000000000e+00	0.0000000000000000e+00
0.0000000000000000e+00	-1.0000000000000005e-01
1.076604963454961816e-01	-1.234808813968935916e-02
9.285609344426656864e-02	-1.121326312494193983e-01
2.110453836830193108e-01	-3.846447531552332072e-02
1.850558031116051882e-01	-1.360255187418834555e-01
3.100610838664475399e-01	-7.385988911778323951e-02
2.758761475637592775e-01	-1.688408715429658424e-01
4.052258921423890214e-01	-1.154953481559722744e-01
3.652746997896266223e-01	-2.081062035747603123e-01
4.973522619098407760e-01	-1.610865755944085043e-01
4.535493330435532711e-01	-2.518005457540001268e-01
5.873054061940690129e-01	-2.089383588685599091e-01
5.411242112703530394e-01	-2.982980063307638652e-01
6.758761883112499591e-01	-2.578169836431673523e-01
6.284127007501388862e-01	-3.463189503723093643e-01
7.637177846436586925e-01	-3.068631367068182514e-01
7.157347676073907428e-01	-3.948925412627222187e-01
8.513196221584131695e-01	-3.555422158899081975e-01
8.032695093465200831e-01	-4.433394937483930431e-01

Tabela 2: Posições finais dos nós computados pelo L-BFGS-B

x	y
0.0000000000000000e+00	0.0000000000000000e+00
0.0000000000000000e+00	-1.0000000000000005e-01
1.076581016588509659e-01	-1.234252877187444822e-02
9.285697905639224425e-02	-1.121273723211763246e-01
2.110435483900849618e-01	-3.845261169162977194e-02
1.850589607610042264e-01	-1.360139709305897282e-01
3.100614950867626729e-01	-7.383759639200435099e-02



Continuação da tabela 3

x	y
2.758808407504163562e-01	-1.688194012225558427e-01
4.052274160172885109e-01	-1.154721206354224761e-01
3.652794197173069302e-01	-2.080853121149189855e-01
4.973554329820704112e-01	-1.610579591497134644e-01
4.535571696553672427e-01	-2.517734303334946966e-01
5.873107838861030272e-01	-2.089042278612215919e-01
5.411329922867171849e-01	-2.982663049064730454e-01
6.758827785101579888e-01	-2.577831678481600219e-01
6.284233764303035485e-01	-3.462874535685726007e-01
7.637250577912088056e-01	-3.068229353638805001e-01
7.157473943252998794e-01	-3.948549020339718885e-01
8.513293735802619500e-01	-3.554980033699064457e-01
8.032843643335488837e-01	-4.432975511277799652e-01

Tabela 3: Posições finais dos nós computados pelo SLSQP

x	y
0.000000000000000000e+00	0.000000000000000000e+00
0.000000000000000000e+00	-1.000000000000000056e-01
1.073453320603787731e-01	-1.131906141904176315e-02
9.390695378617000544e-02	-1.111144547211459221e-01
2.109855703932395077e-01	-3.505453679425795527e-02
1.871162548763355415e-01	-1.329081016022125861e-01
3.109290112846055387e-01	-6.796005448866306031e-02
2.786578919106891261e-01	-1.634131799516077566e-01
4.072288059110498737e-01	-1.080791236868817312e-01
3.685794437575616311e-01	-2.012766910145951105e-01
5.002811533915770825e-01	-1.532024195396144706e-01
4.570264280056981487e-01	-2.440769482580765759e-01
5.908554038684504794e-01	-2.009954397605685228e-01
5.443707874541260372e-01	-2.902751946570642505e-01
6.792081549818771435e-01	-2.504898803189012924e-01
6.309492576946710285e-01	-3.387666102692873382e-01
7.663160328145873779e-01	-3.007088386085366438e-01
7.175186124147107103e-01	-3.884533589826867916e-01
8.533262796408570550e-01	-3.504586983865930483e-01
8.045345136075724435e-01	-4.377923103295616736e-01

Tabela 4: Posições finais dos nós computados pelo Método de Newton Truncado

## 4.2 Métodos Evolutivos

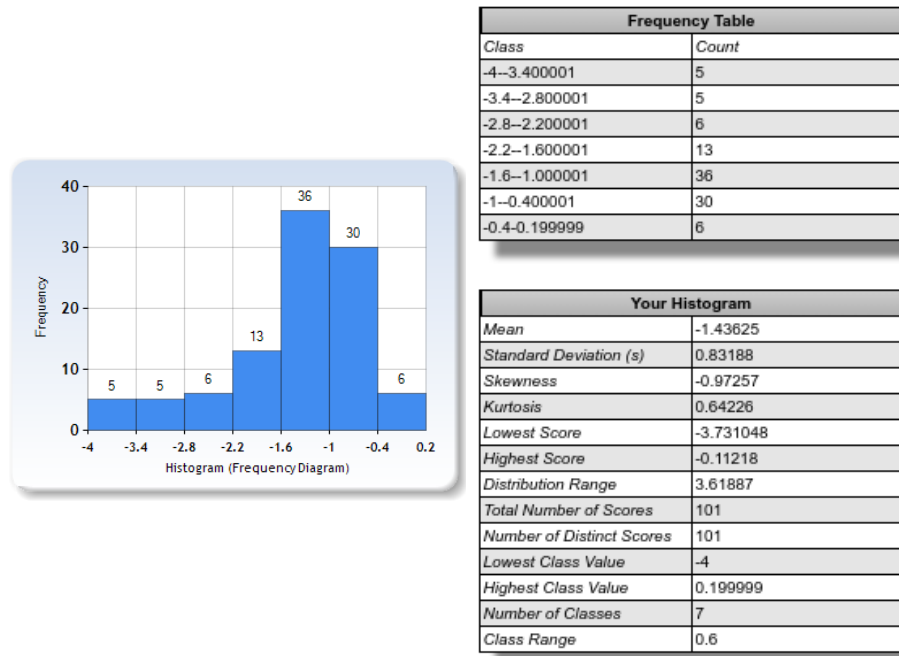


Figura 5: Histograma e análise estatística da resposta do algoritmo de Evolução Diferencial para uma população de 40 candidatos

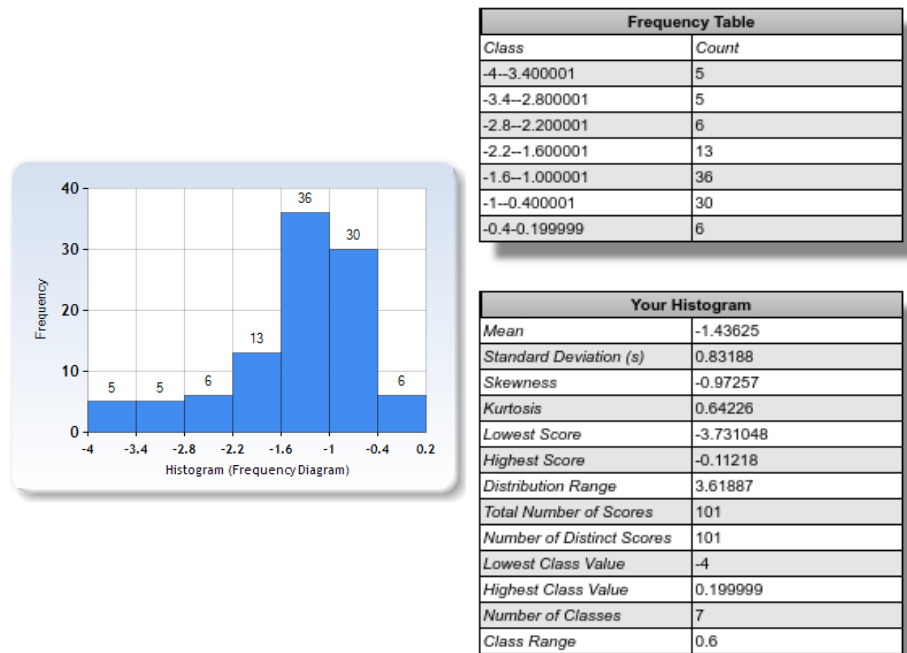


Figura 6: Histograma e análise estatística da resposta do algoritmo de Evolução Diferencial para uma população de 100 candidatos