Ficha de Trabalho 2: Métodos de Pesquisa

- Resolução-

Objetivo: *Pretende-se promover a aquisição de conhecimentos e desenvolvimento de competências relativas aos seguintes algoritmos: subida da colina (Hill Climbing) e Simulated Annealing.*

1) Explique o princípio de funcionamento do algoritmo da subida da colina (maximização). Apresente o pseudo-código para este algoritmo. ***(T)***

R: *Ver diapositivos ou notas da UC*

2) Considere a função unidimensional (1) ilustrada na figura seguinte para o intervalo especificado:

$$f_1(x) = 4(\sin(5\pi\, x + 0.5)^6 \exp(\log_2((x-0.8)^2))) \quad 0 \le x \le 1.6 \qquad (1)$$

Como se pode visualizar na figura seguinte esta função tem vários máximos locais e um máximo global.

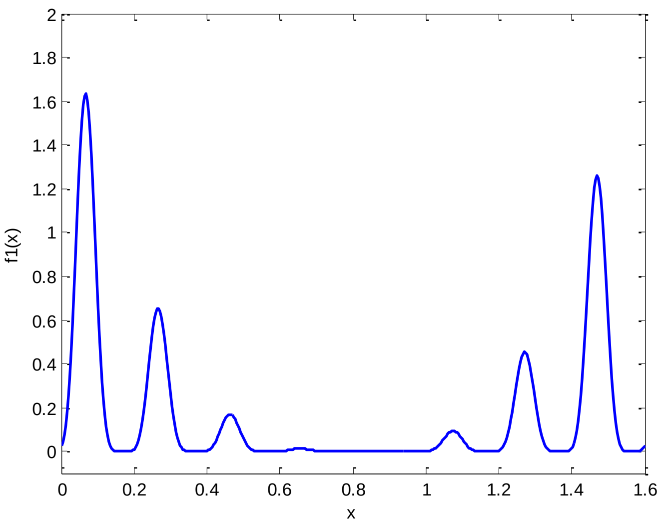


Figura 1: Espaço de pesquisa unidimensional definido pela função (1).

i) Identifique os máximos (global e locais) desta função. ***(T)***

ii) Determine os máximos (global e locais) desta função. ***(P)***

R: *Para além dos que é possível identificar visualmente na Figura há mais alguns máximos de pequena amplitude:*

x =[0.067, 0.266, 0.462, 0.654, 0.791, 0.891, 1.075, 1.271, 1.471]

f(x)=[1.632, 0.652, 0.170, 0.013, 0.0000000095, 0.0027, 0.09318, 0.453, 1.257]

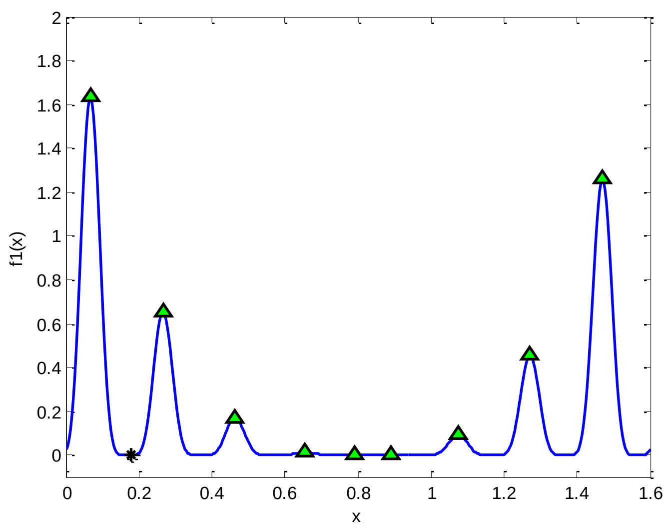
iii) Replique a Figura 1 representando os máximos da função. ***(P)***

R: *Apresenta-se de seguida o código em Matlab:*

```
format long
set(0,'defaultlinelinewidth',2);
set(0,'defaultlinemarkersize',8);
%-----------------------------------------------------------------------
% Exercício 2
%-----------------------------------------------------------------------
% iii) Replicar a imagem
figure
% Function f1: Maximization
```

```
h=ezplot('4*(sin(5*pi*x+0.5)^6)*exp(log2((x-0.8)^2))',[0,1.6]);
axis([0,1.6,-0.1,2]);  % Delimitar os eixos
ylabel('f1(x)')
title('');
% iii) Agora com os máximos sobrepostos
figure
h=ezplot('4*(sin(5*pi*x+0.5)^6)*exp(log2((x-0.8)^2))',[0,1.6]);
axis([0,1.6,-0.1,2]);  % Delimitar os eixos
ylabel('f1(x)')
title('');
x_p = get(h,'XData');
y_p = get(h,'YData');
hold on   % Congela o gráfico
% pks- valor dos picos ( máximo global e local )
[pks,locs] = findpeaks(y_p);
x_pks = x_p(locs)
f_pks = y_p(locs)
plot(x_pks,f_pks,'k^','markerfacecolor','g')
```

que permite detetar os máximos da função e a sua representação gráfica que se apresenta na Figura seguinte:



iv) Considere a solução inicial *x*=0.180. Represente o ponto correspondente no gráfico ***(T-P)***.

***R**: A representação já está feita na figura anterior (*). Continuação do programa anterior:*

```
% iv)
% Ponto inicial fixo
x_current=0.18;
F_current=4*(sin(5*pi*x_current+0.5)^6)*exp(log2((x_current-0.8)^2))
plot(x_current,F_current,'k*')
hold off
```

v) Considere que o algoritmo pode testar pontos aleatórios gerados numa vizinhança com um raio 1/100 da amplitude do intervalo de pesquisa. Quais os limites inferior e superior do intervalo de pesquisa inicial? ***(T)***

***R**: Os cálculos podem também ser feitos num programa:*

```
% Raio da vizinhança
limits=[0,1.6];  % Limites da pesquisa
range=limits(2)-limits(1);
x_radius=range/100;
x_inf_init=x_current-x_radius
x_sup_init=x_current+x_radius
```

x_inf_init=0.1640; x_sup_init =0.1960

3) Considere o exemplo apresentado na Tabela 1, para um problema de maximização, na qual *It* representa o número da iteração e x_{new} uma potencial solução para o problema. As potenciais soluções estão representadas na Figura 2.

It	0	1	2	3	4	5	6	7	8	9	10
x	0.180	0.210	0.220	0.230	0.240	0.250	0.260	0.260	0.260	0.260	0.260
f(x)	0.000	0.0453	0.1228	0.2502	0.4089	0.5563	0.6433	0.6433	0.6433	0.6433	0.6433
x_new		0.210	0.220	0.230	0.240	0.250	0.260	0.2850	0.2450	0.120	0.110
f(x_new)		0.0453	0.1228	0.2502	0.4089	0.5563	0.6433	0.4768	0.4872	0.1358	0.3378

Tabela 1: Exemplo de potenciais pontos para a subida da colina.

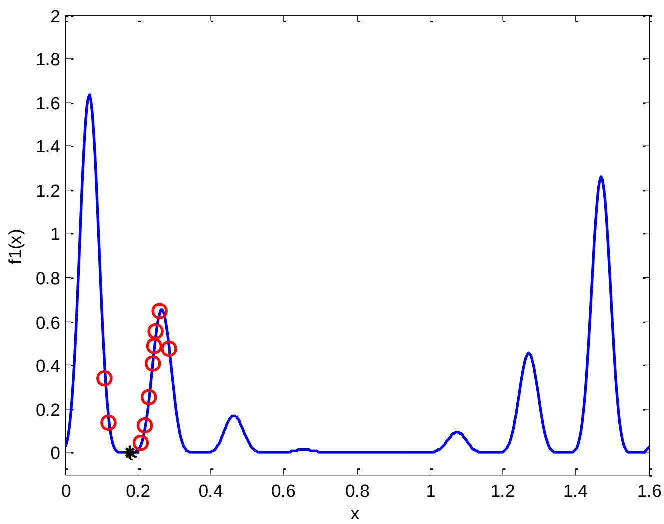


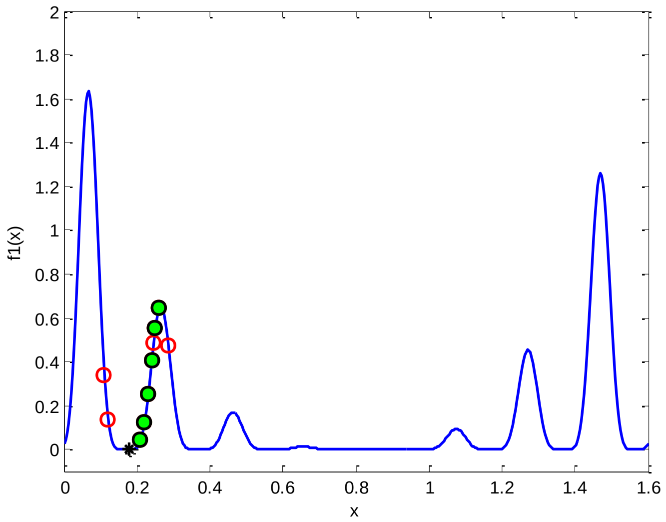
Figura 2: Representação de 10 potenciais soluções apresentadas nas Tabelas 1-3.

i) Aplicando o critério de decisão do algoritmo da subida da colina complete os espaços em branco das variáveis *x* e *f(x)*. ***(T)***

***R**: Basta comparar o valor da imagem, f(x_new), com o valor da solução atual, f(x). Nos casos em que seja maior atualiza-se a solução atual com a nova, bem como a imagem respetiva. Ver Tabela 1 completa.*

ii) Confirme os valores apresentados para *f(x)* e represente os pontos correspondentes a melhoria da função na figura. ***(T-P)***

***R**: A representação das soluções aceites como melhorias estão representadas na figura seguinte preenchidas a verde. Notar as tentativas finais que não são aceites. Estão representadas no gráfico a vermelho com fundo branco.*




```
%-------------------------------------------------------------------
% Exercício 3
% Hill-Climbing
%----------------------------------------------------------------------
x_trials=[0.21 0.22 0.23 0.24 0.25 0.26 0.285 0.245 0.12 0.11]
for j=1:size(x_trials,2)
   F_trials(j)=4*(sin(5*pi*x_trials(j)+0.5)^6)*exp(log2((x_trials(j)-
0.8)^2));
end
F_trials

%Plot de todas as soluções potenciais
figure
h=ezplot('4*(sin(5*pi*x+0.5)^6)*exp(log2((x-0.8)^2))',[0,1.6]);
hold on
axis([0,1.6,-0.1,2]);  % Delimitar os eixos
ylabel('f1(x)')
title('');
plot(x_current,F_current,'k*')
plot(x_trials,F_trials,'or');
hold off

%Plot de todas as soluções hill-climbing
figure
h=ezplot('4*(sin(5*pi*x+0.5)^6)*exp(log2((x-0.8)^2))',[0,1.6]);
hold on
axis([0,1.6,-0.1,2]);  % Delimitar os eixos
ylabel('f1(x)')
title('');
plot(x_current,F_current,'k*')
plot(x_trials,F_trials,'or');

it=10;
x_hc(1)=x_current;
F_hc(1)=F_current;
x_hc(2:it)=0;
F_hc(2:it)=0;
% Hill-climbing
for j=1:it,
    if F_trials(j) > F_current
       x_current = x_trials(j);
       F_current = F_trials(j);
       x_hc(j) = x_trials(j);
       F_hc(j) = F_trials(j);
       plot(x_hc(j),F_hc(j),'ko','markerfacecolor','g')
    else
```

```
        x_hc(j) = x_current;
        F_hc(j) = F_current;
    end
end
hold off
```

iii) Na eventualidade da pesquisa estar presa num máximo local, que estratégia utilizaria para explorar outras regiões do espaço ***(T)***

***R**: Porque não reinicializar a pesquisa noutro ponto ao fim de um determinado número de iterações?*

4) Explique o princípio de funcionamento do algoritmo *Simulated Annealing (SA)* (maximização). Apresente o pseudo-código para este algoritmo. ***(T)***

***R**: Ver diapositivos ou notas da UC*

5) Considere que utiliza o algoritmo SA para o mesmo problema apresentado na alínea 3).

i) Sabendo que os perfis utilizados para a probabilidade (*p*) e temperatura (T) são os apresentados na Figura 2, explique qual a relação entre as duas variáveis. ***(T)***

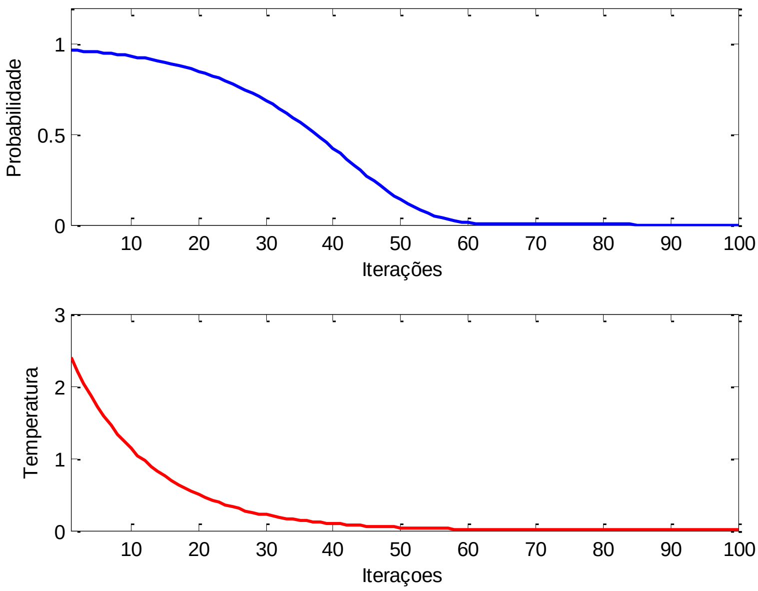


Figura 2: Perfis da probabilidade e temperatura.

***R**: Com a diminuição do valor da temperatura o valor da probabilidade neste caso diminui também.*

ii) Qual é relação entre o decaimento do valor da probabilidade e a diminuição do grau de aleatoriedade do SA? ***(T)***

***R**: No início da pesquisa a probabilidade de aceitar uma solução pior é muito elevada, obtendo-se um elevado grau de aleatoriedade. No fim da pesquisa a probabilidade de aceitar uma solução pior é muito baixa (ou mesmo nula), obtendo-se um baixo grau de aleatoriedade.*

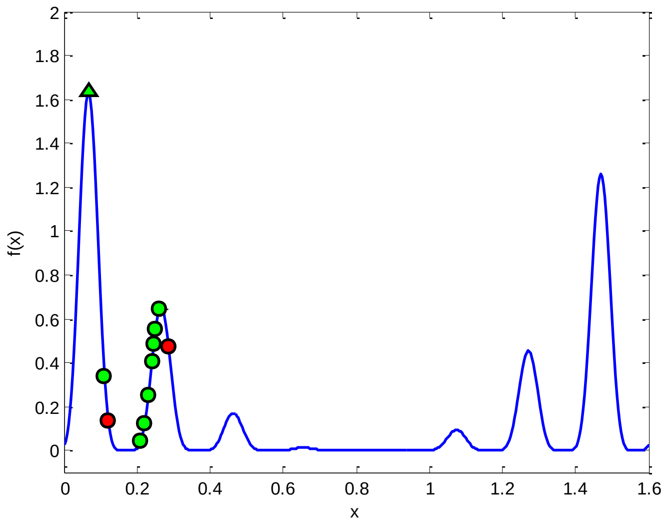
iii) Represente a condição associada ao critério de decisão a utilizar quando a solução de teste piorar o valor atual da solução? ***(T)***

R*: Neste caso é: rand < p*

iv) Considere o exemplo apresentado na Tabela 2, na qual *p* representa o valor da probabilidade e *rand* um valor aleatório gerado no intervalo [0,1]. Aplicando o critério de decisão do algoritmo do SA complete os espaços em branco na linha das variáveis *x* e *f(x)*. ***(T)***

R*: Ver na tabela.*

iv) Represente os pontos aceites na figura. ***(T-P)***



v) Sabendo que a expressão utilizada no arrefecimento da temperatura é representada por: $T = \alpha T$, com base nos valores apresentados na Tabela 2 determine o valor de α.

It	0	1	2	3	4	5	6	7	8	9	10
x	0.180	0.210	0.220	0.230	0.240	0.250	0.260	0.2850	0.2450	0.120	0.110
f(x)	0.000	0.0453	0.1228	0.2502	0.4089	0.5563	0.6433	0.4768	0.4872	0.1358	0.3378
x_new		0.210	0.220	0.230	0.240	0.250	0.260	0.2850	0.2450	0.120	0.110
f(x_new)		0.0453	0.1228	0.2502	0.4089	0.5563	0.6433	0.4768	0.4872	0.1358	0.3378
T		2.4000	2.2080	2.0314	1.8689	1.7193	1.5818	1.4553	1.3388	1.2317	1.1332
p								0.8919		0.753	
rand								0.601		0.124	

Tabela 2: Exemplo de potenciais pontos para o *Simulated Annealing*: caso I.

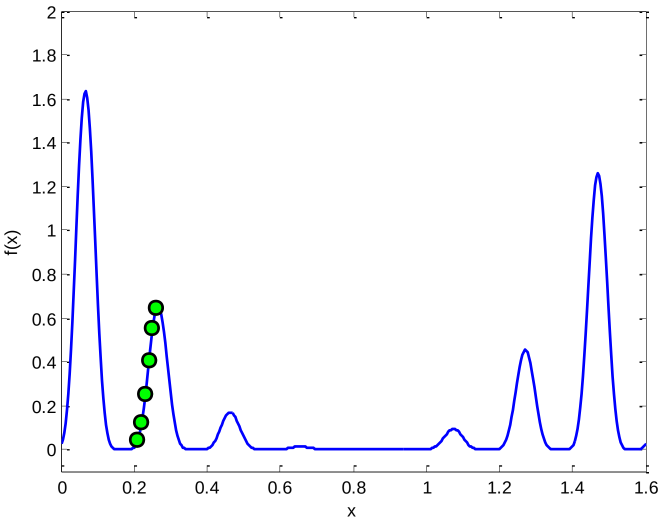
vi) Considere o exemplo apresentado na Tabela 3, que transpõe o hipotético caso apresentado na Tabela 3, para as últimas 10 iterações (num total de 100). Aplicando o critério de decisão do algoritmo do SA complete os espaços em branco na linha das variáveis *x* e *f(x)* ***(T)***

R*: Ver na tabela.*

vii) Compare os resultados obtidos na alínea *iv)* com os da alínea *vi)*. ***(T)***

R*: Verifica-se que se este caso estivesse acontecido no final da pesquisa esta não escaparia do máximo local..*

viii) Represente os pontos aceites na figura. ***(T-P)***



It	90	91	92	93	94	95	96	97	98	99	100
x	0.180	0.210	0.220	0.230	0.240	0.250	0.260	0.260	0.260	0.260	0.260
f(x)	0.000	0.0453	0.1228	0.2502	0.4089	0.5563	0.6433	0.6433	0.6433	0.6433	0.6433
x_new		0.210	0.220	0.230	0.240	0.250	0.260	0.2850	0.2450	0.120	0.110
f(x_new)		0.0453	0.1228	0.2502	0.4089	0.5563	0.6433	0.4768	0.4872	0.1358	0.3378
T		0.0014	0.0013	0.0012	0.0011	0.0010	0.0009	0.0008	0.0007	0.00067	0.00063
p								0.0000	0.0000	0.00000	0.00000
rand								0.844	0.7882	0.0370	0.8371

Tabela 3: Exemplo de potenciais pontos para o *Simulated Annealing*: caso II.

Referências:

P. B. de Moura Oliveira, E. J. Solteiro Pires e Paulo Novais, (2016), “Revisiting the Simulated Annealing Algorithm from a Teaching Perspective”, M. Graña et al. (eds.), International Joint Conference SOCO’16-CISIS’16-ICEUTE’16, Advances in Intelligent Systems and Computing 527, DOI 10.1007/978-3-319-47364-2_70