

Trabalho Prático 2: Métodos de Pesquisa (Search Methods)
- Parte A -

Motivação: Os métodos de pesquisa de base estocástica são fundamentais na resolução de múltiplos problemas de otimização.

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

Parte I- Introdução

- 1) Faça uma pesquisa de informação sobre:
 - Algoritmo da subida da colina (e variantes);
 - Algoritmo Simulated Annealing (e variantes).
 (Sugestão: assista também aos vídeos [V2-V5])
- 2) Escreva uma síntese [V1] introdutória da informação recolhida, que deve constar no relatório a entregar, **incluindo as referências bibliográficas utilizadas.**

Parte II- Otimização de Funções

- 3) Considere a função unidimensional (1) ilustrada na Figura 1 para o intervalo especificado:

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

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

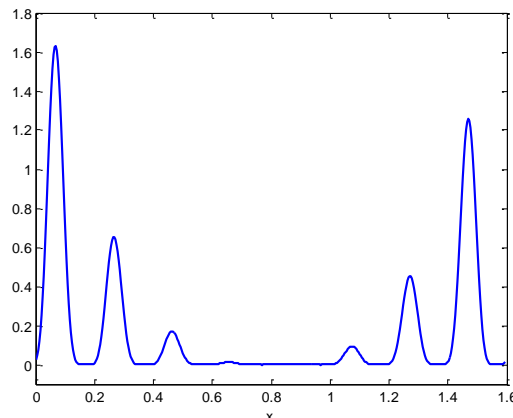


Figura 1: Espaço de pesquisa unidimensional definido pela função (1).

Implemente um algoritmo da subida da colina com reinicialização múltipla e teste-o na otimização desta função. Considere inicialmente: 300 iterações por teste, permitindo testar novos pontos do espaço que distam do melhor valor atingido, no máximo 0.02 da amplitude do espaço de pesquisa.

- 4) Implemente o algoritmo do *Simulated Annealing* (SA) para o problema apresentado nos pontos anterior compare os resultados obtidos.

Nota 1: Considere utilizar a seguinte designação para as variáveis:

T (temperatura); $nRep$ (número de repetições para cada valor da temperatura); α (fator de decaimento para a temperatura); dE (gradiente de energia); p (probabilidade).

Parte III- Problema do Caixeiro Viajante

- 5) Considere agora o famoso problema do caixeiro viajante (*Traveling Salesman Problem*, TSP) numa das suas variantes mais simples: pretende-se encontrar a rota mais curta (menor distância percorrida) que permita visitar um conjunto pré-definido de cidades. A soma das distâncias geográficas entre as várias cidades (nós) dispostas sequencialmente pode ser

calculada utilizando a seguinte função de custo (i representa uma cidade e n o número total de cidades):

$$J = \min \sum_{i=1}^{n-1} d(i, i+1) + d(n, 1) \quad (2)$$

Pretende-se utilizar o algoritmo *Simulated Annealing* para resolver este problema. Propõe-se uma abordagem que utilize as coordenadas das cidades em termos da sua latitude e longitude. O primeiro conjunto de cidades a considerar está ilustrado na figura seguinte e as coordenadas apresentam-se na Tabela 1 (ver Anexo):

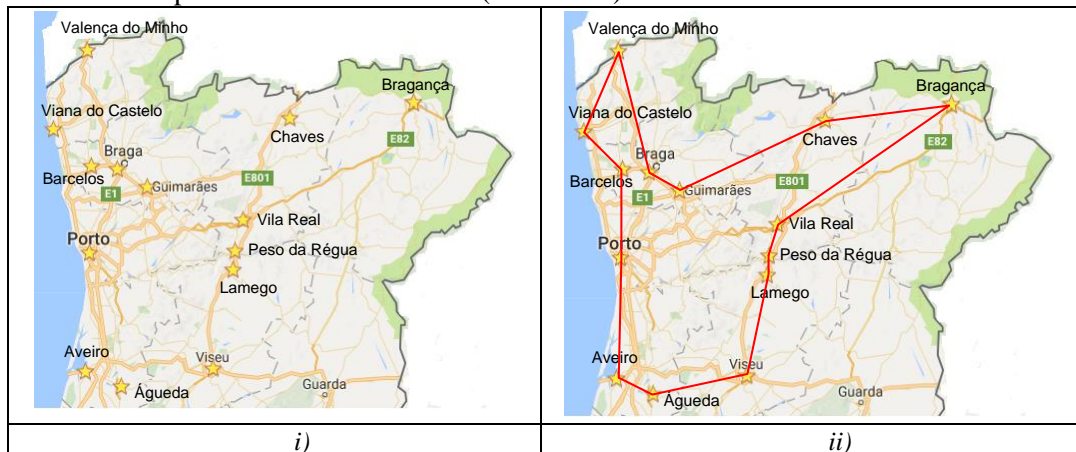


Figura 2: Conjunto de 14 cidades em Portugal (Figura adaptada do Google Maps, 2017).

Nota 2: Serão disponibilizados no NONIO alguns scripts em Matlab de apoio a esta componente.

Parte IV- Relatório

- 6) Escreva um relatório sucinto com todos os elementos que permitam a avaliação do trabalho. **Submeta o relatório no NONIO em PDF** juntamente com o(s) programas em Matlab numa **pasta zipada com o nome:** Trab_2_números_dos_alunos_do_grupo.

Nota 3:

Parte da avaliação do trabalho será feita de forma contínua nas aulas práticas.

Relatórios que não sejam submetidos em formato PDF não serão avaliados.

Vídeos:

- [V1] Trabalho de Síntese: https://youtu.be/t_OZbolXY9s (PT)
- [V2] Subida da Colina (Hill-Climbing): https://youtu.be/T_6pA3VNHFc (EN)
- [V3] Subida da Colina com Re-Inicialização Múltipla (Multiple Restart Hill-Climbing): <https://youtu.be/LtDvNXvfUzU> (EN)
- [V4] Simulated Annealing: <https://youtu.be/w2rBcPo88XM> (PT)
- [V5] Simulated Annealing: <https://youtu.be/RR5LbWWJ8Os> (Part 1) (EN)
- [V6] Simulated Annealing: <https://youtu.be/kye7sli7z94> (Part 2) (EN)

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

Paulo Moura Oliveira, *Nature Inspired Search and Optimization: a simplified approach, Part-I* available in InforEstudante (NONIO).

Anexo:

Tabela 1: Coordenadas para 14 cidades Portuguesas (conjunto *pt_nt*)

#	Cidade	Latitude	Longitude
1	Bragança	41N49	6W45
2	Vila Real	41N18	7W45
3	Chaves	41N44	7W28
4	Viana do Castelo	41N42	8W50
5	Braga	41N33	8W26
6	Aveiro	40N38	8W39
7	Porto	41N11	8W36
8	Viseu	40N39	7W55
9	Lamego	41N06	7W49
10	Águeda	40N34	8W27
11	Peso da Régua	41N10	7W47
12	Guimarães	41N27	8W18
13	Valença do Minho	42N02	8W38
14	Barcelos	41N32	8W37

Tabela 2: Coordenadas para 20 cidades Portuguesas (conjunto *pt_nt_sul_20*)

#	Cidade	Latitude	Longitude
1	Bragança	41N49	6W45
2	Vila Real	41N18	7W45
3	Chaves	41N44	7W28
4	Viana do Castelo	41N42	8W50
5	Braga	41N33	8W26
6	Aveiro	40N38	8W39
7	Porto	41N11	8W36
8	Viseu	40N39	7W55
9	Lamego	41N06	7W49
10	Guimarães	41N27	8W18
11	Coimbra	40N12	8W25
12	Faro	37N01	7W56
13	Évora	38N34	7W54
14	Lisboa	38N43	9W10
15	Portalegre	39N17	7W26
16	Tavira	37N07	7W39
17	Sagres	37N00	8W56
18	Setúbal	38N32	8W54
19	Guarda	40N32	7W16
20	Santarém	39N14	8W41

Tabela 3: Coordenadas para 30 cidades Portuguesas (conjunto *pt_nt_sul_30*)

#	City	Latitude	Longitude	#	City	Latitude	Longitude
1	Bragança	41N49	6W45	16	Tavira	37N07	7W39
2	Vila Real	41N18	7W45	17	Sagres	37N00	8W56
3	Chaves	41N44	7W28	18	Setúbal	38N32	8W54
4	Viana do Castelo	41N42	8W50	19	Guarda	40N32	7W16
5	Braga	41N33	8W26	20	Santarém	39N14	8W41
6	Aveiro	40N38	8W39	21	Beja	38N01	7W52
7	Porto	41N11	8W36	22	Sines	37N57	8W52
8	Viseu	40N39	7W55	23	Covilhã	40N17	7W30
9	Lamego	41N06	7W49	24	Tomar	39N36	8W25
10	Guimarães	41N27	8W18	25	Águeda	40N34	8W27
11	Coimbra	40N12	8W25	26	Leiria	39N45	8W48
12	Faro	37N01	7W56	27	Castelo Branco	39N49	7W30
13	Évora	38N34	7W54	28	Elvas	38N53	7W10

14	Lisboa	38N43	9W10	29	Miranda do Douro	41N30	6W16
15	Portalegre	39N17	7W26	30	Sintra	38N48	9W23