

Simulated Annealing

O tópico destas aulas é a Implementação e utilização do Simulated Annealing para a reslução de problemas de caixeiro viajante. Pretende-se que implemente, usando uma linguagem julgada adequada (C, Java, ...), o Simulated Annealing para pesquisar o melhor caminho num problema do caixeiro viajante. Na sua formulação original, o problema consiste em descobrir o circuito mais curto que percorra N cidades, sem repetir nenhuma das cidades e voltando à cidade de partida. Entre quaisquer duas cidades existe um caminho (estrada) directo entre elas com um determinado comprimento (caso não existisse essa via directa, considera-se um comprimento arbitrariamente grande).

1 - Descrição do problema e algoritmo

Começe por ler com atenção o [guião](#) que contém uma descrição do problema bem como algumas sugestões sobre a melhor forma de o modelar para ser resolvido pelo algoritmo de simulated annealing.

2 - Implementação do algoritmo

Implemente um programa que utilize o Simulated Annealing para a reslução de problemas de caixeiro viajante. O programa a implementar terá as seguintes partes:

1. Inicialização das estruturas de dados, incluindo a matriz de distâncias (ver [exemplo](#))
2. Leitura dos parametros de execução, incluindo qual o conjunto das cidades que vão ser consideradas. Toda esta informação pode estar num ficheiro ou ser fornecido na linha de comando
3. Execução (ver [algoritmo](#))
4. Apresentação dos resultados

Algumas indicações para a implementação

- O cálculo da função de avaliação deve ser diferencial, i.e. deve adicionar ao valor da função de avaliação da solução anterior a diferença de custo para a solução seguinte.
- Não perca tempo em interfaces gráficas.

Sugestões de funcionalidades a implementar:

- Possibilidade de definir o modo de funcionamento do algoritmo, nomeadamente no que se refere ao método de decaimento da temperatura (geométrico, gradual, ...), ao método de variação do número de iterações por temperatura (constante, linear, ...), ao critério de paragem (temperatura mínima, total de iterações, percentagem de soluções aceites,...), à forma de determinação da solução inicial (aleatória, inteligente, ...), e à forma de determinação da temperatura inicial (dada pelo utilizador, calculada, ...).
- Ajuste automático dos parâmetros de acordo com a instância a tratar. O programa deveria propor valores por defeito para os parâmetros, preferencialmente dependentes da instância em resolução.
- No final de uma execução deve apresentar os seguintes resultados: a melhor solução, a pior solução, a primeira e a última soluções, o número total de iterações, tempo de execução. Note que para cada solução apresentada pode indicar, a iteração em que ocorreu, o valor da temperatura, a solução (o percurso) e o valor da função.

3 - Teste do algoritmo

Teste a sua implementação com os seguintes exemplos:

- **E1:** Atroeira, Douro, Pinhal, Teixoso, Ulgueira, Vilar.
- **E2:** Cerdeira, Douro, Gonta, Infantado, Lourel, Nelas, Oura, Quebrada, Roseiral, Serra, Teixoso, Ulgueira
- **E3:** Belmar, Cerdeira, Douro, Encosta, Freita, Gonta, Horta, Infantado, Lourel, Monte, Nelas, Oura, Pinhal, Quebrada, Roseiral, Serra, Teixoso, Ulgueira.
- **E4:** todas as cidades

TABELA 1 - Cidades e suas distâncias (em Km) [\[versão texto simples\]](#)

Distância	Atroeira																				
Belmar	383	Belmar																			
Cerdeira	129	504	Cerdeira																		
Douro	287	566	185	Douro																	
Encosta	239	271	366	299	Encosta																
Freita	60	329	178	314	91	Freita															
Gonta	305	78	426	488	191	251	Gonta														
Horta	522	165	643	732	437	468	244	Horta													
Infantado	163	369	260	197	102	161	291	535	Infantado												
Jardim	506	210	622	765	456	452	270	85	652	Jardim											
Lourel	126	273	244	402	179	72	195	412	233	385	Lourel										
Monte	256	183	372	530	264	202	138	296	402	250	146	Monte									
Nelas	438	98	558	700	365	380	172	93	512	68	324	182	Nelas								
Oura	276	178	403	390	93	222	100	344	193	360	172	219	235	Oura							
Pinhal	71	446	58	216	308	123	368	585	202	565	189	317	505	339	Pinhal						
Quebrada	188	195	309	464	181	134	117	346	295	330	78	80	229	147	251	Quebrada					
Roseiral	299	143	420	575	316	245	105	256	406	205	189	47	150	186	362	123	Roseiral				
Serra	383	93	501	641	340	325	115	143	478	126	270	128	61	203	451	191	90	Serra			
Teixoso	144	519	56	241	382	191	441	658	275	641	262	367	578	412	69	324	435	520	Teixoso		
Ulgueira	169	528	94	120	261	199	450	683	159	656	282	412	585	352	98	349	460	535	150	Ulgueira	
Vilar	86	415	185	228	177	86	366	554	75	327	158	288	452	268	127	220	331	395	241	113	Vilar

TABELA 2 - Cidades e suas coordenadas (em quilómetros) [\[versão texto simples\]](#)

Cidade	X	Y
Atroeira	116	464
Belmar	184	137
Cerdeira	134	564
Douro	272	593
Encosta	216	374
Freita	137	417
Gonta	180	198
Horta	179	27
Infantado	233	453
Jardim	113	36
Lourel	104	365
Monte	73	214
Nelas	115	91
Oura	222	316
Pinhal	119	521
Quebrada	113	274
Roseiral	95	193
Serra	96	131
Teixoso	100	580
Ulgueira	192	537
Vilar	180	467