Simulated Annealing

O tópico destas aulas é a implementação e utilização do Simulated Annealing para a resolução 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 directo entre elas com um determinado comprimento (caso não exista essa via directa, considera-se um comprimento arbitrariamente grande).

1 – Descrição do problema e algoritmo

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