



UNIVERSIDADE DE  
COIMBRA

FACULDADE  
DE CIÊNCIAS  
E TECNOLOGIA

Licenciatura em Engenharia Informática  
Introdução à Inteligência Artificial  
2019/2020 – 2º Semestre

D31: The Last Search

<i>Diogo Jorge da Rocha Cota</i>	2017267601	<a href="mailto:uc2017267601@student.uc.pt">uc2017267601@student.uc.pt</a>	PL4
<i>João Filipe Carnide de Jesus Nunes</i>	2017247442	<a href="mailto:uc2017247442@student.uc.pt">uc2017247442@student.uc.pt</a>	PL4
<i>Raul Filipe Enes Nogueira</i>	2017267634	<a href="mailto:uc2017267634@student.uc.pt">uc2017267634@student.uc.pt</a>	PL4

# Índice

Introdução	3
Objetivos Alcançados	3
Mapa NoObstacles	4
Pesquisa Aleatória	4
Trepas Colinas	5
Recristalização Simulada	5
Conclusões	5
Mapa ObstaclesSmall	6
Pesquisa Aleatória	6
Trepas Colinas	6
Recristalização Simulada	6
Conclusões	7
Mapa ObstaclesSmallManyBoxes	7
Pesquisa Aleatória	7
Trepas Colinas	8
Recristalização Simulada	8
Conclusões	8
Mapa ReturnTo2b	9
Pesquisa Aleatória	9
Trepas Colinas	9
Recristalização Simulada	10
Conclusões	10
Mapa ReturnTo2bHarder	10
Pesquisa Aleatória	10
Trepas Colinas	11
Recristalização Simulada	11
Conclusões	11
Mapa Bomberman	12
Pesquisa Aleatória	12
Trepas Colinas	12
Recristalização Simulada	13
Conclusões	13
Mapa PacmanOriginal	14
Pesquisa Aleatória	14
Trepas Colinas	14
Recristalização Simulada	14
Conclusões	15
Conclusão	15
Referências	16

## Introdução

Neste segundo trabalho prático pretende-se fazer a implementação de um agente de procura que, através de algoritmos de procura e pesquisa, consiga, num certo ambiente, encontrar a melhor forma de colecionar todos os recursos que estão presentes no mapa. Ou seja, pretende-se que o agente D31 tenha incorporado em si memória, raciocínio com as respetivas regras para proceder à procura dos recursos presentes no ambiente.

Na meta 1 foi pedida a implementação e experimentação do algoritmo Trepacolinhas de modo a otimizar a pesquisa feita pelo agente usando o algoritmo previamente implementado de Pesquisa Aleatória.

Na meta 2 foi feita a implementação do algoritmo de Recristalização Simulada bem como a sua experimentação. Também a experimentação do algoritmo de Pesquisa Aleatória foi realizada de modo a ser possível chegar a uma conclusão de qual dos três algoritmos implementados no trabalho prático é o melhor num certo ambiente e no geral.

## Objetivos Alcançados

Visto que o algoritmo de procura já se encontrava previamente implementado ( $A^*$ ), o objetivo deste trabalho prático era a implementação do algoritmo de pesquisa Trepacolinhas e de Recristalização Simulada de modo a otimizar a distância do algoritmo de Pesquisa Aleatória, já implementado previamente.

O algoritmo de Pesquisa Aleatória consiste em encontrar uma solução arbitrária de um espaço de pesquisa onde é usada uma distribuição de probabilidades uniforme. Cada uma das iterações da pesquisa é completamente independente das anteriores. Contudo, a previa implementação do algoritmo permite uma comparação do valor adquirido com o anterior e, caso seja uma solução pior, manter-se-á a mesma solução.

O Trepacolinhas é um algoritmo iterativo que começa com uma solução arbitrária para um problema e, a partir daí, tenta encontrar uma melhor solução fazendo uma alteração incremental na solução. Caso esta alteração produza uma melhor solução, será feita outra incrementação nesta nova solução até não serem encontradas soluções melhores.

O algoritmo de Recristalização Simulada, baseado no arrefecimento de metais, gera uma nova possível solução para um dado problema pela alteração do estado atual a cada recristalização virtual da temperatura (diminuição escalar da temperatura), de acordo com critérios predefinidos. Este algoritmo no trabalho prático conteve a implementação da função *TemperatureSchedule* que permite que a temperatura diminua ao longo de cada iteração, sendo que esta função foi desenvolvida de duas formas. Uma implementação linear onde a temperatura diminui segundo a expressão  $T_i = T_{inicial} - n*i$ , onde  $T_{inicial}$  é a temperatura inicial (igual a 5000),  $i$  corresponde à iteração em que nos encontramos e  $n$  uma constante de decaimento (implementado com o valor de 2). A outra implementação foi uma diminuição da temperatura geometricamente onde a temperatura diminui com base numa certa percentagem da temperatura anterior, que pode ser traduzida pela

seguinte equação  $T_{i+1} = T_i - (T_i * (dim / 100))$ , onde  $T_i$  corresponde à temperatura na iteração atual e  $dim$  à percentagem que a temperatura diminuirá em cada iteração.

No que diz respeito à experimentação dos três algoritmos foram feitas variações no número máximo de iterações (sublinhado a verde) e, após ser encontrado um valor ideal deste número, foi variada também a *random seed* para facilitar a análise do algoritmo em média. O valor ideal do número máximo de iterações é encontrado através de se obter o melhor custo de deslocação com o menos número de iterações possível. No caso do algoritmo de Recristalização Simulada existe uma altura em que a temperatura chega a 0 (2224 iterações) logo é considerado que não há melhorias no que diz respeito à distância e ao caminho percorrido pelo agente nas iterações seguintes.

No que diz respeito à Recristalização Simulada e às duas funções de *TemperatureSchedule* que foram implementadas foram feitos testes para verificar qual das duas implementações era a melhor, ou seja, aquela que dava um caminho melhor no geral. Para este teste foram feitas várias medições no que diz respeito ao número máximo de iterações num ambiente com um dado grau de complexidade (mapa ReturnTo2bHarder). A conclusão chegada foi que a implementação geométrica é melhor que a linear, pois nas medições realizadas apresenta um melhor valor do custo. O melhor valor a que a implementação linear consegui chegar foi de 137, enquanto a geométrica apresentou um valor muito mais otimizado de 83. Assim, para os seguintes testes de cada ambiente será utilizada a Recristalização Simulada com uma diminuição de temperatura geométrica.

Ver pasta Medições em anexo para mais dados que os presentes neste relatório.

## Mapa NoObstacles

Mapa simples com poucos recursos para recolher e sem obstáculos.

## Pesquisa Aleatória

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	17	17	17	17	17	17	17	17	17	17
<b>Teste 2</b>	17	17	17	17	17	17	17	17	17	17
<b>Teste 3</b>	17	17	17	17	17	17	17	17	17	17

Tabela das distâncias percorridas com *random seed* variável (5 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	17	21	17	17	17	26	17
<b>Teste 2</b>	17	21	17	17	17	26	17
<b>Teste 3</b>	17	21	17	17	17	26	17

Melhor caminho encontrado: [2, 3, 1]

## Trepa Colinas

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	17	17	17	17	17	17	17	17	17	17
Teste 2	17	17	17	17	17	17	17	17	17	17
Teste 3	17	17	17	17	17	17	17	17	17	17

Tabela das distâncias percorridas com *random seed* variável (5 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	17	17	17	21	17	17	17
Teste 2	17	17	17	21	17	17	17
Teste 3	17	17	17	21	17	17	17

Melhor caminho encontrado: [2, 3, 1]

## Recristalização Simulada

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	22	26	26	17	27	32	17	17	17	17
Teste 2	22	26	26	17	27	27	17	17	17	17
Teste 3	22	26	26	17	27	21	17	17	17	17

Tabela das distâncias percorridas com *random seed* variável (100 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	27	17	21	17	17	27	21
Teste 2	22	17	26	17	17	27	27
Teste 3	22	17	21	17	17	27	27

Melhor caminho encontrado: [2, 3, 1]

## Conclusões

As conclusões que tiramos da análise da experimentação deste mapa são que todos os três algoritmos conseguem chegar ao melhor custo possível, distância de 17. Nos testes da *random seed* foi calculada a média e o desvio padrão de cada um dos algoritmos, com isto chegamos à conclusão de que, para este mapa, o pior algoritmo é o algoritmo de Recristalização Simulada devido à sua grande complexidade para um problema relativamente simples e, com uma média de aproximadamente 17.57, o algoritmo Trepa Colinas apresenta a melhor média e desvio padrão para este mapa.

Assim, para este mapa o melhor algoritmo é o Trepa Colinas devido à sua simplicidade para um mapa desta complexidade.

## Mapa ObstaclesSmall

Mapa pequeno com alguns recursos para recolha e vários obstáculos.

### Pesquisa Aleatória

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	41	41	30	30	25	25	25	25	25	25
Teste 2	41	41	30	30	25	25	25	25	25	25
Teste 3	41	41	30	30	25	25	25	25	25	25

Tabela das distâncias percorridas com *random seed* variável (250 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	27	25	30	31	25	27	29
Teste 2	27	25	30	31	25	27	29
Teste 3	27	25	30	31	25	27	29

Melhor caminho encontrado: [1, 2, 4, 3, 5, 6]

### Trepa Colinas

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	40	33	33	30	30	30	30	30	30	30
Teste 2	40	33	33	30	30	30	30	30	30	30
Teste 3	40	33	33	30	30	30	30	30	30	30

Tabela das distâncias percorridas com *random seed* variável (100 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	25	25	25	29	30	29	29
Teste 2	25	25	25	29	30	29	29
Teste 3	25	25	25	29	30	29	29

Melhor caminho encontrado: [1, 2, 4, 3, 5, 6]

### Recristalização Simulada

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	55	43	65	56	37	50	35	25	25	25
Teste 2	55	43	65	56	43	45	29	29	29	29
Teste 3	55	43	65	56	43	27	45	25	25	25

Tabela das distâncias percorridas com *random seed* variável (1000 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	25	25	25	29	25	29	30
<b>Teste 2</b>	29	25	29	29	29	25	25
<b>Teste 3</b>	29	25	30	25	25	29	29

Melhor caminho encontrado: [1, 2, 4, 3, 5, 6]

## Conclusões

Neste mapa, todos os algoritmos chegam ao menor custo possível com uma distância de 25. Nos testes do *random seed*, ao analisar o cálculo da média e do desvio padrão das distâncias em cada algoritmo, concluímos que o melhor algoritmo para este mapa é o Recristalização Simulada, com uma média e desvio padrão de aproximadamente 27.19 e 2.15, respetivamente. O algoritmo que teve o pior desempenho neste mapa, apesar dos valores da média e desvio padrão não diferenciarem muito do Recristalização simulada, foi o algoritmo de Pesquisa Aleatória com uma média de 27.71 e desvio padrão de 2.21.

Em suma, para um mapa relativamente simples como este, o melhor algoritmo é o Recristalização Simulada, apesar de todos os outros algoritmos conseguirem encontrar uma boa solução e obterem valores para a média e desvio padrão que diferenciam uns dos outros por uma pequena margem.

## Mapa ObstaclesSmallManyBoxes

Mapa com alguns obstáculos e vários recursos para o agente recolher.

### Pesquisa Aleatória

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	79	79	49	49	49	49	49	49	42	42
<b>Teste 2</b>	79	79	49	49	49	49	49	49	42	42
<b>Teste 3</b>	79	79	49	49	49	49	49	49	42	42

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	50	51	48	47	42	44	47
<b>Teste 2</b>	50	51	48	47	42	44	47
<b>Teste 3</b>	50	51	48	47	42	44	47

Melhor caminho encontrado: [9, 5, 7, 10, 6, 1, 2, 4, 11, 8, 12, 3]

## Trepa Colinas

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	77	76	50	50	44	36	30	30	30	30
Teste 2	77	76	50	50	44	36	30	30	30	30
Teste 3	77	76	50	50	44	36	30	30	30	30

Tabela das distâncias percorridas com *random seed* variável (750 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	40	34	34	30	30	44	41
Teste 2	40	34	34	30	30	44	41
Teste 3	40	34	34	30	30	44	41

Melhor caminho encontrado: [2, 11, 12, 8, 4, 1, 6, 9, 7, 10, 5, 3]

## Recristalização Simulada

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	81	74	90	70	86	84	61	37	35	35
Teste 2	81	74	90	70	81	73	52	43	37	37
Teste 3	81	74	90	70	72	100	45	41	37	37

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	39	41	30	40	35	30	39
Teste 2	35	30	34	35	37	40	40
Teste 3	31	34	34	37	37	38	35

Melhor caminho encontrado: [9, 6, 1, 2, 11, 4, 8, 12, 3, 5, 10, 7]

## Conclusões

No mapa *ObstaclesSmallManyBoxes*, os algoritmos Trepa Colinas e Recristalização Simulada chegaram ao melhor custo possível com uma distância de 30.

Fazendo uma análise dos resultados obtidos pelos três algoritmos, averiguámos que a Pesquisa Aleatória tem a pior performance comparativamente aos outros algoritmos, em que o melhor resultado obtido para a distância foi 42 e média de 47, embora o valor do desvio padrão seja o melhor valor obtido. O algoritmo que melhor satisfaz nos resultados que obteve foi o Recristalização Simulada com uma média de 35.76 e um desvio padrão de 3.50.



Concluimos que para este mapa, com alguma complexidade (ainda que baixa), os melhores algoritmos a aplicar são o Trepa Colinas e o Recristalização Simulada, sendo que este último obtém os melhores resultados.

## Mapa ReturnTo2b

Mapa complexo com vários obstáculos e recursos para o agente recolher.

### Pesquisa Aleatória

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	85	85	85	83	83	83	80	75	73	66
<b>Teste 2</b>	85	85	85	83	83	83	80	75	73	66
<b>Teste 3</b>	85	85	85	83	83	83	80	75	73	66

Tabela das distâncias percorridas com *random seed* variável (10000 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	69	71	67	67	66	63	63
<b>Teste 2</b>	69	71	67	67	66	63	63
<b>Teste 3</b>	69	71	67	67	66	63	63

Melhor caminho encontrado: [1, 2, 4, 5, 9, 8, 7, 6, 3]

### Trepa Colinas

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	135	117	71	71	67	67	67	67	67	67
<b>Teste 2</b>	135	117	71	71	67	67	67	67	67	67
<b>Teste 3</b>	135	117	71	71	67	67	67	67	67	67

Tabela das distâncias percorridas com *random seed* variável (250 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	69	70	67	71	67	63	67
<b>Teste 2</b>	69	70	67	71	67	63	67
<b>Teste 3</b>	69	70	67	71	67	63	67

Melhor caminho encontrado: [1, 2, 4, 5, 9, 8, 7, 6, 3]

## Recristalização Simulada

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	135	135	116	117	123	102	66	64	64	64
Teste 2	135	135	116	117	107	109	79	71	71	71
Teste 3	135	135	116	117	107	99	80	64	64	64

Tabela das distâncias percorridas com *random seed* variável (1000 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	69	65	64	67	64	63	67
Teste 2	67	67	64	64	71	70	67
Teste 3	63	69	67	63	64	65	69

Melhor caminho encontrado: [1, 2, 4, 9, 8, 7, 5, 6, 3]

## Conclusões

A partir do estudo dos três algoritmos para o mapa ReturnTo2b concluímos que todos os algoritmos conseguem obter a melhor solução possível, de custo 63.

Analisando os testes da variação da *random seed* obtivemos a melhor média de resultados com o algoritmo da Recristalização Simulada e a pior média foi-nos dada pelo método Trepac Colinas, mas este conseguiu ter o melhor resultado para desvio padrão. O valor do desvio padrão dado pela Recristalização Simulada permitiu-nos concluir que este algoritmo poderá ser o melhor para este mapa, pois tem a melhor média de custos e um desvio padrão muito próximo do melhor desvio padrão encontrado.

## Mapa ReturnTo2bHarder

Mapa complexo com vários obstáculos e muitos recursos para recolha por parte do agente.

## Pesquisa Aleatória

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	181	181	165	165	165	145	145	145	145	141
Teste 2	181	181	165	165	165	145	145	145	145	141
Teste 3	181	181	165	165	165	145	145	145	145	141

Tabela das distâncias percorridas com *random seed* variável (10000 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	130	139	131	119	141	129	131
Teste 2	130	139	131	119	141	129	131
Teste 3	130	139	131	119	141	129	131

Melhor caminho encontrado: [17, 15, 2, 4, 14, 13, 9, 11, 8, 7, 5, 12, 6, 3, 10, 1, 16]

## Trepa Colinas

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	207	194	146	131	96	87	83	83	83	83
<b>Teste 2</b>	207	194	146	131	96	87	83	83	83	83
<b>Teste 3</b>	207	194	146	131	96	87	83	83	83	83

Tabela das distâncias percorridas com *random seed* variável (750 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	101	99	81	87	83	95	87
<b>Teste 2</b>	101	99	81	87	83	95	87
<b>Teste 3</b>	101	99	81	87	83	95	87

Melhor caminho encontrado: [3, 6, 5, 10, 1, 15, 17, 2, 14, 4, 13, 16, 9, 12, 11, 8, 7]

## Recristalização Simulada

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	237	224	217	205	199	239	136	105	89	89
<b>Teste 2</b>	237	224	217	205	235	201	108	83	83	83
<b>Teste 3</b>	237	224	217	205	229	193	137	119	117	117

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	100	91	86	79	89	81	85
<b>Teste 2</b>	91	94	91	103	83	87	93
<b>Teste 3</b>	85	99	87	79	117	83	86

Melhor caminho encontrado: [3, 6, 5, 13, 14, 2, 17, 15, 1, 10, 4, 16, 9, 12, 11, 8, 7]

## Conclusões

Analisando todos os métodos para o mapa ReturnTo2bHarder verificamos que o algoritmo de Recristalização Simulada obtém o melhor custo, 79, embora tenha o maior desvio padrão. O método Trepa Colinas dá-nos um custo de 81 e o algoritmo de Pesquisa Aleatória mostra-se ser o pior por ter um custo de 119, um valor muito superior ao dos restantes métodos, podendo, então, ser desprezado. Posto isto, podemos concluir que para um mapa como este, já com alguma complexidade, os métodos de Pesquisa Aleatória e Trepa Colinas poderão não encontrar a melhor solução, embora consigam ter um desvio padrão melhor que na Recristalização Simulada.

Num mapa desta complexidade já se esperaria que o melhor custo nos seria dado pelo método de Recristalização Simulada pois este é modelado após um processo onde há uma maior probabilidade de encontrar a melhor solução, ao contrário do que acontece no Trepa Colinas que tenta encontrar a melhor solução através do gradiente de erro, o que em situações de maior complexidade, como neste mapa, poderá não funcionar bem.

## Mapa Bomberman

Mapa simples, ilustrativo do jogo *Bomberman* dos anos 80, com muitos obstáculos e recursos para o agente recolher.

### Pesquisa Aleatória

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	179	176	166	166	137	137	137	137	125	125
<b>Teste 2</b>	179	176	166	166	137	137	137	137	125	125
<b>Teste 3</b>	179	176	166	166	137	137	137	137	125	125

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	135	135	139	133	125	133	135
<b>Teste 2</b>	135	135	139	133	125	133	135
<b>Teste 3</b>	135	135	139	133	125	133	135

Melhor caminho encontrado: [1, 2, 4, 14, 16, 15, 20, 17, 10, 18, 8, 13, 6, 7, 11, 12, 19, 9, 3, 5]

## Trepa Colinas

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	187	173	127	115	103	91	89	83	83	83
<b>Teste 2</b>	187	173	127	115	103	91	89	83	83	83
<b>Teste 3</b>	187	173	127	115	103	91	89	83	83	83

Tabela das distâncias percorridas com *random seed* variável (1000 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	91	91	92	89	83	85	99
<b>Teste 2</b>	91	91	92	89	83	85	99
<b>Teste 3</b>	91	91	92	89	83	85	99

Melhor caminho encontrado: [16, 15, 13, 11, 9, 3, 5, 8, 4, 2, 1, 6, 7, 10, 14, 17, 20, 18, 19, 12]

## Recristalização Simulada

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
<b>Teste 1</b>	195	221	204	169	191	193	126	101	89	89
<b>Teste 2</b>	195	221	204	169	197	183	135	106	98	98
<b>Teste 3</b>	195	221	204	169	191	213	135	97	77	77

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	50	100	500	1000	2020	5000	10000
<b>Teste 1</b>	93	87	95	98	89	87	84
<b>Teste 2</b>	93	91	98	83	98	96	93
<b>Teste 3</b>	81	88	89	87	77	82	86

Melhor caminho encontrado: [16, 10, 6, 7, 4, 1, 2, 8, 5, 3, 9, 12, 11, 13, 18, 19, 20, 17, 15, 14]

## Conclusões

Na análise deste mapa podemos concluir que existem dois algoritmos que apresentam um custo aceitável no que diz respeito ao melhor caminho encontrado, o Trepa Colinas e a Recristalização Simulada. O algoritmo de Pesquisa Aleatória é excluído aqui pois apresenta um custo muito maior que os outros dois, distância de 125.

No que diz respeito à média os dois algoritmos apresentam uma média semelhante, sendo a média da Recristalização Simulada sensivelmente melhor que a do Trepa Colinas. Contudo o algoritmo Trepa Colinas apresenta um desvio padrão melhor que a Recristalização Simulada.

Com isto podemos concluir que o algoritmo de Recristalização Simulada é melhor para encontrar uma solução ótima global neste mapa, onde o Trepa Colinas apenas encontra a melhor solução local.

## Mapa PacmanOriginal

Mapa complexo, ilustrativo do jogo dos anos 80 *Pac-Man*, com bastantes obstáculos e recursos para o agente colecionar.

## Pesquisa Aleatória

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	529	439	439	439	436	426	412	412	385	385
Teste 2	529	439	439	439	436	426	412	412	385	385
Teste 3	529	439	439	439	436	426	412	412	385	385

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	401	388	386	385	385	375	390
Teste 2	401	388	386	385	385	375	390
Teste 3	401	388	386	385	385	375	390

Melhor caminho encontrado: [8, 6, 3, 2, 10, 12, 17, 9, 7, 19, 18, 15, 11, 21, 22, 23, 14, 13, 16, 20, 24, 5, 4, 1]

## Trepa Colinas

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	521	484	424	388	338	304	285	257	235	235
Teste 2	521	484	424	388	338	304	285	257	235	235
Teste 3	521	484	424	388	338	304	285	257	235	235

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	50	100	500	1000	2020	5000	10000
Teste 1	241	251	260	224	235	239	221
Teste 2	241	251	260	224	235	239	221
Teste 3	241	251	260	224	235	239	221

Melhor caminho encontrado: [5, 6, 8, 14, 9, 11, 13, 16, 15, 10, 4, 1, 2, 3, 7, 12, 17, 20, 24, 18, 23, 22, 21, 19]

## Recristalização Simulada

Tabela das distâncias percorridas com número máximo de iterações variável

	5	10	50	100	250	500	750	1000	5000	10000
Teste 1	511	532	480	570	553	482	329	291	249	249
Teste 2	511	532	480	558	521	500	298	250	249	249
Teste 3	511	532	480	558	517	477	293	245	231	231

Tabela das distâncias percorridas com *random seed* variável (5000 iterações)

	<b>50</b>	<b>100</b>	<b>500</b>	<b>1000</b>	<b>2020</b>	<b>5000</b>	<b>10000</b>
<b>Teste 1</b>	272	234	249	236	249	245	262
<b>Teste 2</b>	242	222	244	226	249	220	229
<b>Teste 3</b>	271	230	240	264	231	254	227

Melhor caminho encontrado: [6, 5, 9, 4, 1, 2, 3, 7, 8, 14, 12, 17, 20, 24, 18, 16, 23, 22, 21, 19, 15, 10, 11, 13]

## Conclusões

As conclusões que podemos tirar da experimentação deste mapa é que dois dos algoritmos implementados apresentam uma solução semelhante no que diz respeito à distância percorrida pelo agente, o Trepa Colinas e a Recristalização Simulada. A Pesquisa Aleatória é excluída pois apresenta uma distância muito superior àquela obtida nos outros dois algoritmos.

Na análise das médias obtidas nos algoritmos concluímos que o algoritmo Trepa Colinas apresenta uma média e desvio padrão melhor que a Recristalização Simulada.

Em conclusão, para este mapa, o melhor algoritmo é o Trepa Colinas, apesar da Recristalização Simulada ter o melhor valor da distância, 220.

## Conclusão

Em suma, este trabalho prático permitiu a implementação e experimentação de três algoritmos de pesquisa, o de Pesquisa Aleatória, o Trepa Colinas e o de Recristalização Simulada, de modo a que o agente consiga colecionar todos os recursos existentes num certo ambiente com a menor distância possível.

A conclusão que se tirou com os resultados obtidos das medições de cada um dos algoritmos nos diferentes mapas existentes foi a seguinte: o algoritmo de Pesquisa Aleatória é, sem dúvidas, o pior dos três visto que se baseia apenas em combinações aleatórias para cada caminho; o Trepa Colinas é o algoritmo ideal para encontrar a solução ótima a partir do gradiente de erro, contudo este algoritmo apenas encontra a solução ótima local e não a global; já o algoritmo de Recristalização Simulada apresenta a melhor oportunidade de encontrar a solução ótima global, através do arrefecimento lento dos materiais.

Com isto, podemos, finalmente, concluir que o algoritmo de Recristalização Simulada é, geralmente, a melhor opção dos três algoritmos implementados neste trabalho prático para encontrar uma menor distância.

## Referências

[http://www.cleveralgorithms.com/nature-inspired/stochastic/random\\_search.html](http://www.cleveralgorithms.com/nature-inspired/stochastic/random_search.html)

[https://en.wikipedia.org/wiki/Hill\\_climbing](https://en.wikipedia.org/wiki/Hill_climbing)

<https://www.sciencedirect.com/topics/engineering/simulated-annealing-algorithm>

<https://www.quora.com/What-is-main-difference-between-hill-climbing-and-simulated-annealing>

<http://www.scielo.org.mx/pdf/cys/v21n3/1405-5546-cys-21-03-00493.pdf>