

## Algoritmos de Associação e Otimização

### Problema da Mochila – Algoritmo Genético

O problema da Mochila é uma questão que envolve otimização combinatória, a tarefa no caso que vamos caminhar é para isso precisamos levar alguns itens que tem atributos de peso e pontos de sobrevivência, para levar estes itens temos uma bolsa que suporta até 25 kg, logo será necessário priorizar alguns itens em detrimento de outros. A seguir a lista de itens:

	item	peso	sobrevivencia
1	casaco de chuva	2	5
2	canivete	1	3
3	água mineral	6	15
4	luvas	1	5
5	saco de dormir	4	6
6	barraca	9	18
7	fogão portátil	5	8
8	comida enlatada	8	20
9	lanches	3	8

Para escolher os itens vamos utilizar o algoritmo genético. No primeiro momento, vamos criar um modelo com 30 gerações (ou iterações), chamado GA, depois ajustamos o modelo para 50 gerações, chamando-o de GA2, e depois vamos ajustar novamente os parâmetros para 100 gerações, chamando-o de GA3. Criando os gráficos de Fitness value-Generation para cada modelo, eles ficam assim:

Gráfico Fitness value-Generation - GA

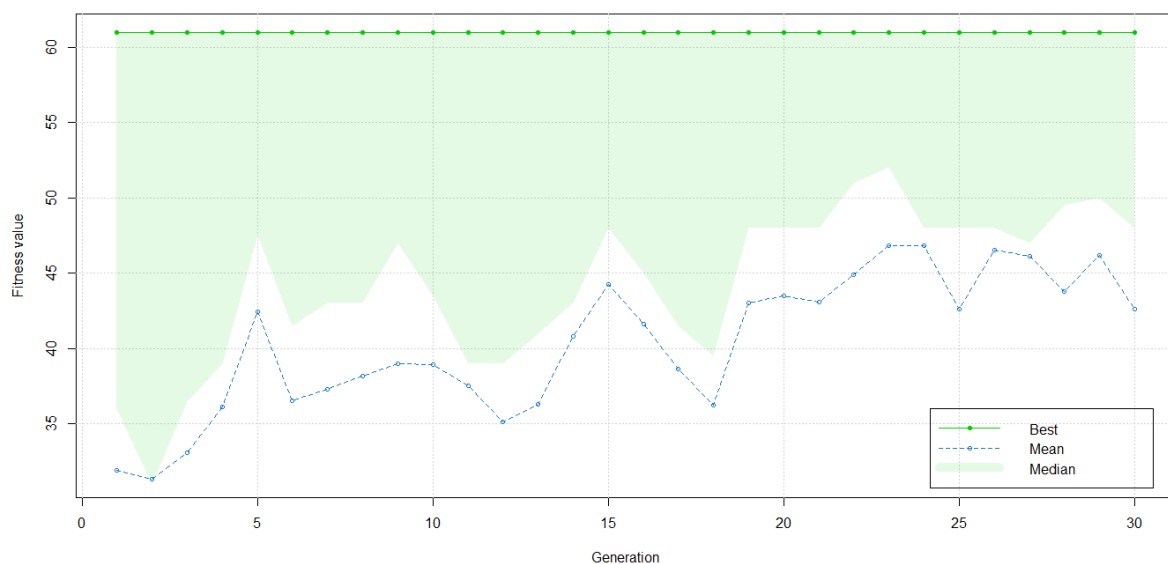


Gráfico Fitness value-Generation - GA2

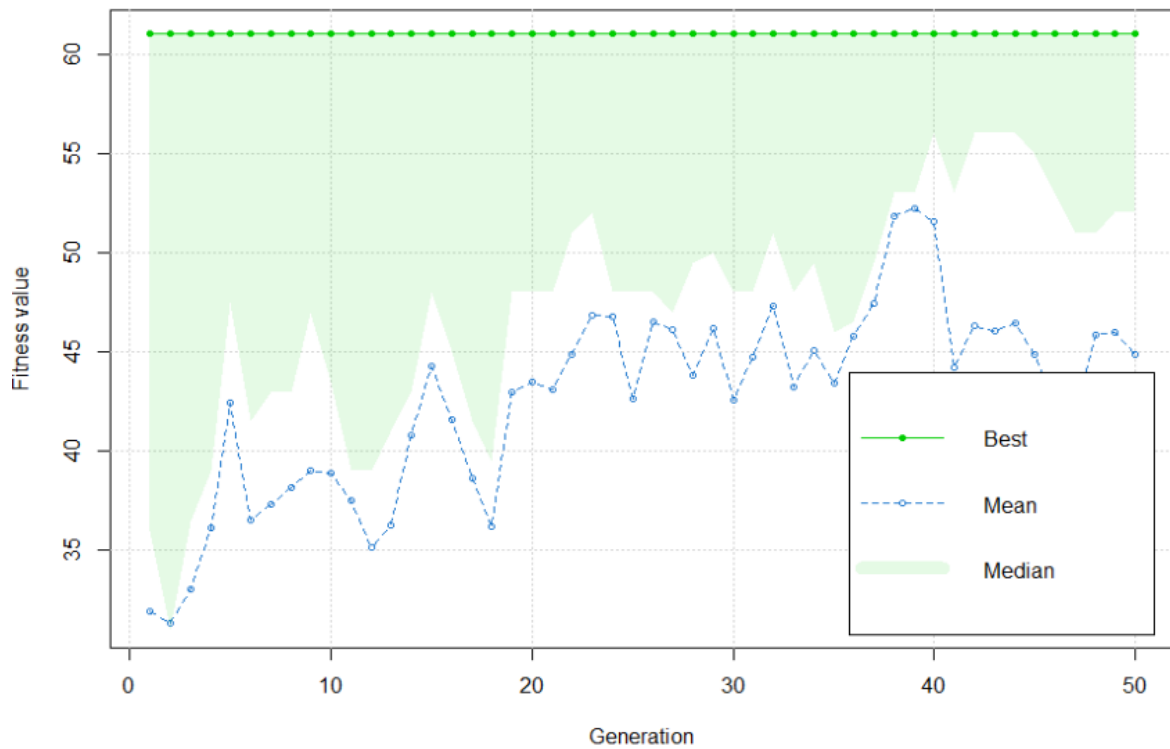
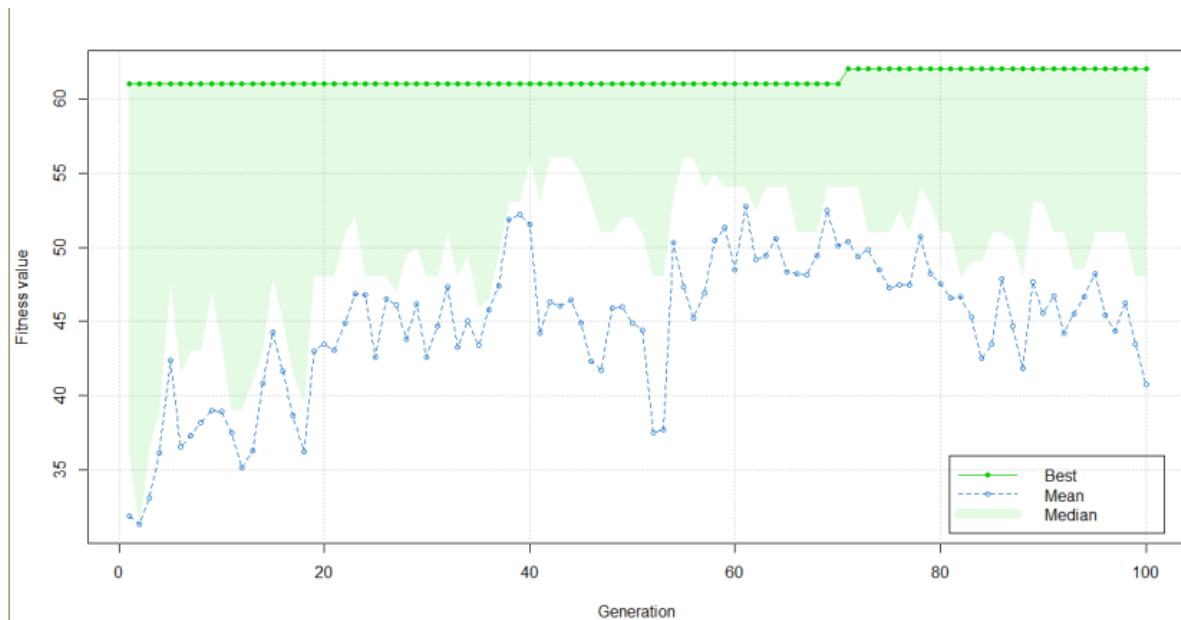


Gráfico Fitness value-Generation - GA3



Observando os gráficos acima, podemos concluir que, no caso do GA e do GA2, o valor de fitness tende a aumentar conforme aumenta a quantidade de gerações. Já em GA3, a média e mediana do valor fitness atinge resultados otimizados entre a geração 40 e 60, aproximadamente, e o melhor valor de fitness aumenta para 62 da 72ª geração em diante, diferente do GA e do GA2, em que o melhor valor de fitness é constante a cada geração.

Dessa forma, podemos ver que o modelo GA3 apresentou resultados melhores. Visualizando o cromossomo final, podemos concluir que o resultado otimizado de itens que deve ser levados na caminhada é o seguinte:

	item	peso	sobrevivencia
1	casaco de chuva	2	5
2	canivete	1	3
3	água\mineral	6	15
4	luvas	1	5
5	saco de dormir	4	6
8	comida enlatada	8	20
9	lanches	3	8

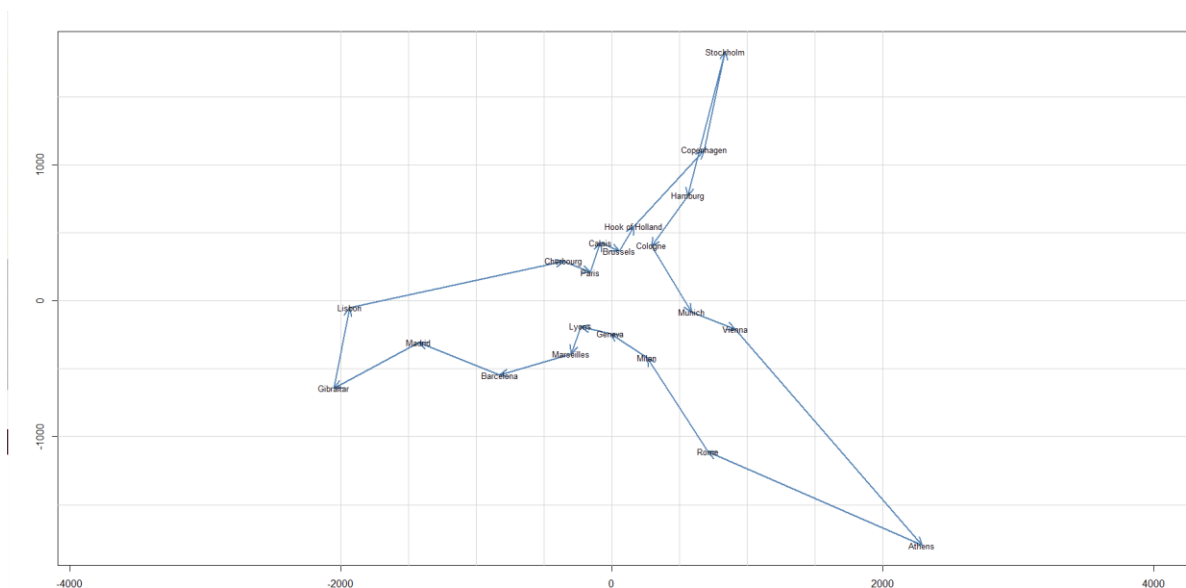
## O Problema do Caixeiro Viajante

O TSP (em inglês Travelling Salesman Problem) é um problema clássico que consiste em procurar o círculo com a menor distância começando em uma cidade, visitando várias outras e no final retornando para a cidade inicial. Nesta questão temos uma tabela com a distancia entre varias cidades da europa e buscase encontrar a melhor rota.

Para resolver o problema vamos utilizar o método de algoritmo genético. A seguir é possível visual a tabela com as cidades:

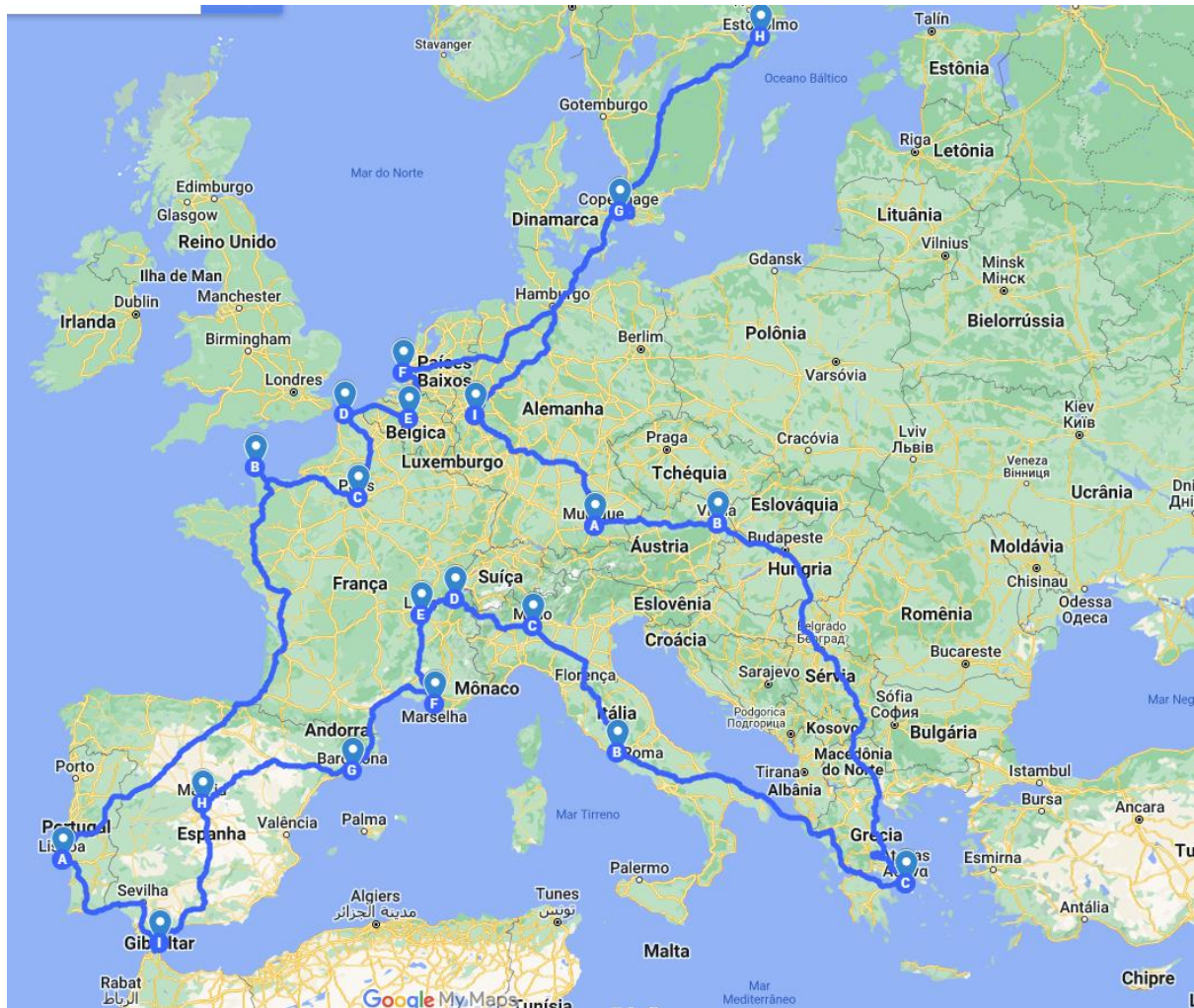
	Athens	Barcelona	Brussels	Calais	Cherbourg	Cologne	Copenhagen	Geneva	Gibraltar	Hamburg	Hook of Holland	Lisbon	Lyons	Madrid	Marseilles	Milan	Munich	Paris	Rome	Stockholm	Vienna
Athens	0	3313	2963	3175	3339	2762	3276	2610	4485	2977	3030	4532	2753	3949	2865	2282	2179	3000	817	3927	1991
Barcelona	3313	0	1318	1326	1294	1498	2218	803	1172	2018	1490	1305	645	636	521	1014	1365	1033	1460	2868	1802
Brussels	2963	1318	0	204	583	206	966	677	2256	597	172	2084	690	1558	1011	925	747	285	1511	1616	1175
Calais	3175	1326	204	0	460	409	1136	747	2324	714	330	2052	739	1550	1059	1077	977	280	1662	1786	1381
Cherbourg	3339	1294	583	460	0	785	1545	853	2047	1115	731	1827	789	1347	1101	1209	1160	340	1794	2196	1588
Cologne	2762	1498	206	409	785	0	760	1662	2436	460	269	2290	714	1764	1035	911	583	465	1497	1403	937
Copenhagen	3276	2218	966	1136	1545	760	0	1418	3196	460	269	2971	1458	2498	1778	1537	1104	1176	2050	650	1455
Geneva	2610	803	677	747	853	1662	1418	0	1975	1118	895	1936	158	1439	425	328	591	513	995	2068	1019
Gibraltar	4485	1172	2256	2324	2047	2436	3196	1975	0	2897	2428	676	1817	698	1693	2185	2565	1971	2631	3886	2974
Hamburg	2977	2018	597	714	1115	460	460	1118	2897	0	550	2671	1159	2198	1479	1238	805	877	1751	949	1155
Hook of Holland	3030	1490	172	330	731	269	269	895	2428	550	0	2280	863	1730	1183	1098	851	457	1683	1500	1205
Lisbon	4532	1305	2084	2052	1827	2290	2971	1936	676	2671	2280	0	1178	668	1762	2250	2507	1799	2700	3231	2937
Lyons	2753	645	690	739	789	714	1458	158	1817	1159	863	1178	0	1281	320	328	724	471	1048	2108	1157
Madrid	3949	636	1558	1550	1347	1764	2498	1439	698	2198	1730	668	1281	0	1157	1724	2010	1273	2097	3188	2409
Marseilles	2865	521	1011	1059	1101	1035	1778	425	1693	1479	1183	1762	320	1157	0	618	1109	792	1011	2428	1363
Milan	2282	1014	925	1077	1209	911	1537	328	2185	1238	1098	2250	328	1724	618	0	331	856	586	2187	886
Munich	2179	1365	747	977	1160	583	1104	591	2565	805	851	2507	724	2010	1109	331	0	821	946	1754	428
Paris	3000	1033	285	280	340	465	1176	513	1971	677	457	1799	471	1273	792	856	821	0	1476	1827	1249
Rome	817	1460	1511	1662	1794	1497	2050	995	2631	1751	1683	2700	1048	2097	1011	586	946	1476	0	2707	1209
Stockholm	3927	2868	1616	1786	2196	1403	650	2068	3886	949	1500	3231	2108	3188	2428	2187	1754	1827	2707	0	2105
Vienna	1991	1802	1175	1381	1588	937	1455	1019	2974	1155	1205	2937	1157	2409	1363	886	428	1249	1209	2105	0

Para resolver o problema utilizamos a função fitness que minimiza a duração do trajeto e depois utilizamos a função da biblioteca GA que calcula o caminho mais curto, colocando em formato de visualização o resultado do problema fica assim:



Podemos perceber que neste exemplo há uma grande simplificação do que seria no mundo real, e uma situação em uma empresa poderia se considerar estado das estradas, quantidade

de entregas, mercado consumidor e muitos outros fatores. Para exemplificar colocamos os destinos em um mapa



O link para melhor visualização: [https://www.google.com/maps/d/u/0/edit?mid=1QEXHrQx-B3VqJTGZQ\\_ttPP0a22Fp7zc&usp=sharing](https://www.google.com/maps/d/u/0/edit?mid=1QEXHrQx-B3VqJTGZQ_ttPP0a22Fp7zc&usp=sharing)

Assim, podemos concluir que matematicamente ele retorna o menor caminho, mas o algoritmo não tinha informações sobre a geografia do continente e além disso seria necessário analisar qual modal é melhor para cada trecho do caminho (no mapa acima, considera o percurso feito a partir do transporte rodoviário).

## **Conclusão**

A partir do presente laboratório, foi possível notar a vasta aplicabilidade do algoritmo genético em diferentes contextos, seja em um problema de logística, ou um problema de alocação de recursos, ou ainda problemas de seleção de portfólio de investimentos, por exemplo. Basta coletar os dados e definir uma função objetivo e os parâmetros utilizados na modelagem, buscando gerar um resultado otimizado a partir disso.

Entretanto, dificilmente o algoritmo irá gerar uma solução pronta para já ser aplicada no mundo real. Precisamos, nesse caso, levar em consideração uma série de outros fatores essenciais para a tomada de decisão e que nem sempre são possíveis de serem incorporados ao modelo. Como mencionado no laboratório, no caso do problema do caixante viajante, por exemplo, o algoritmo não considera as vias de transporte reais para calcular as distâncias entre os locais, como também não leva em conta outros fatores como o número de pedidos por cidade e a capacidade de cada veículo de transporte para chegar na solução final. Sendo assim, o algoritmo oferece um ponto de partida para que o tomador de decisão possa chegar em uma solução real ótima, levando em consideração os outros fatores determinantes do problema.