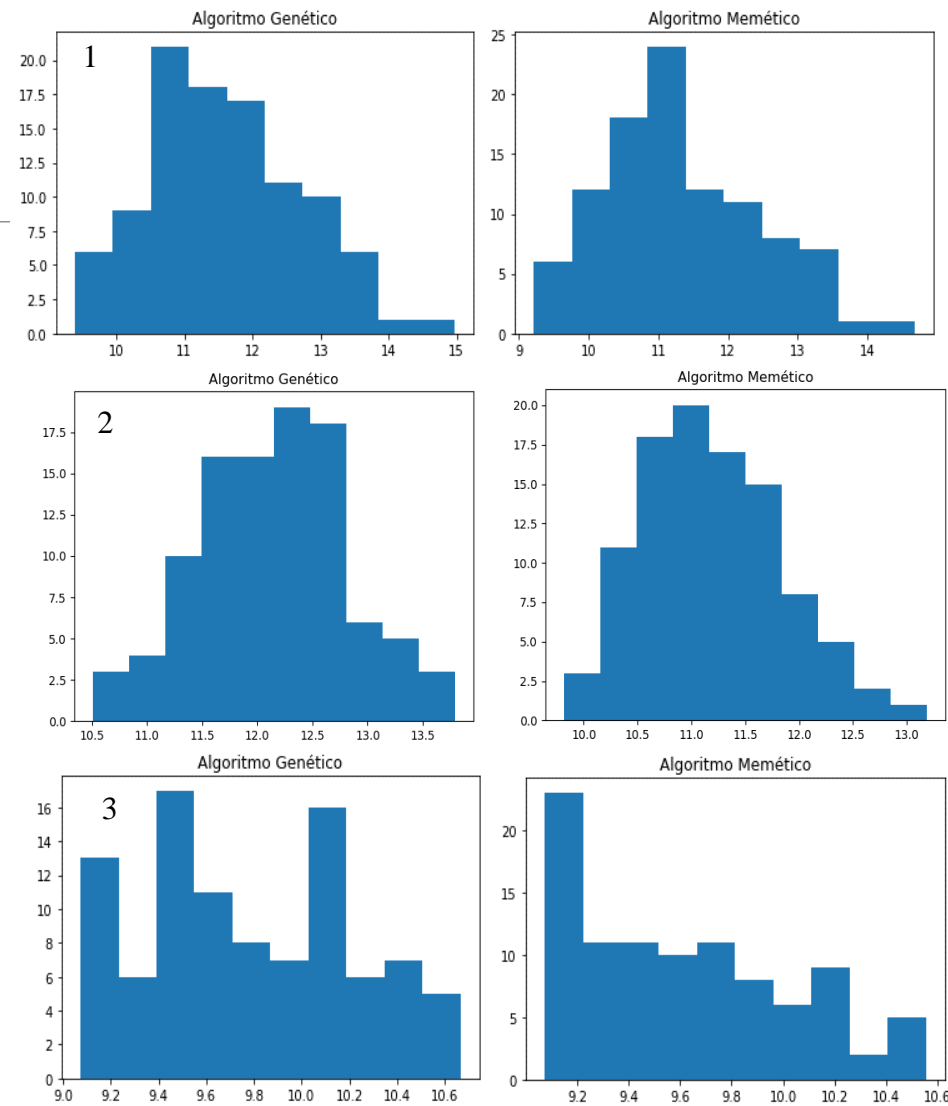
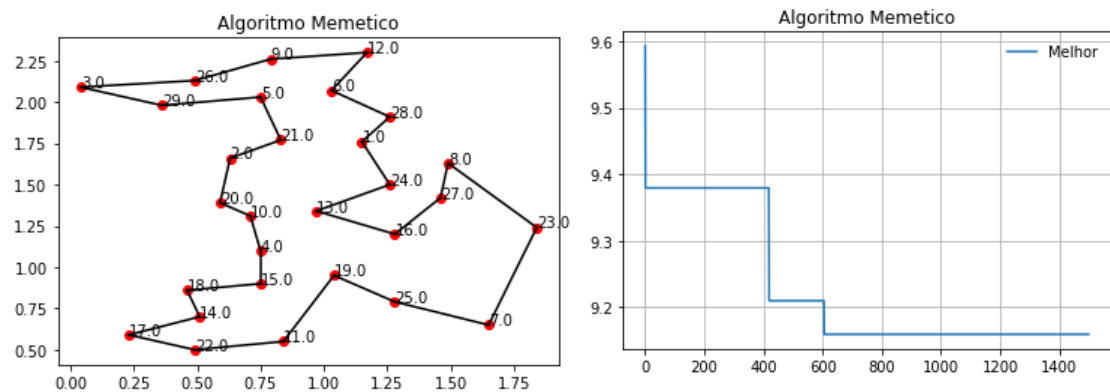
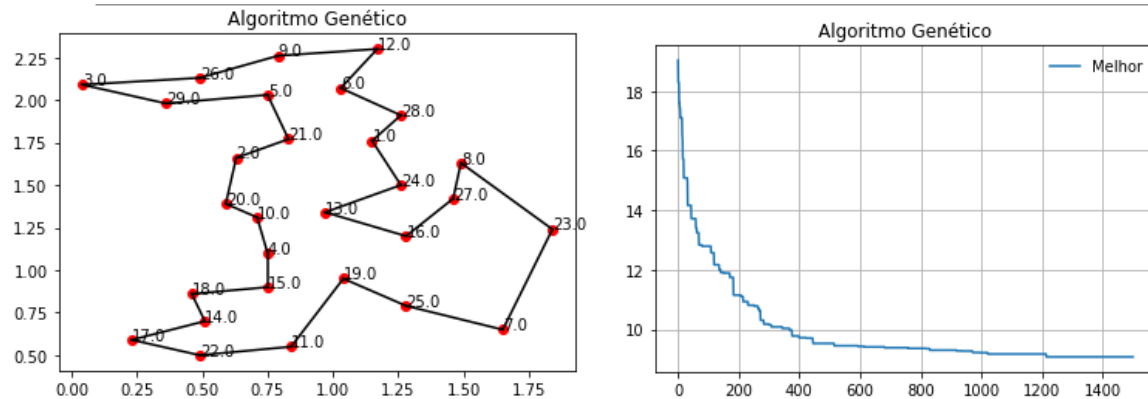


Problema do Caixeiro Viajante - 29 cidades

1- Código

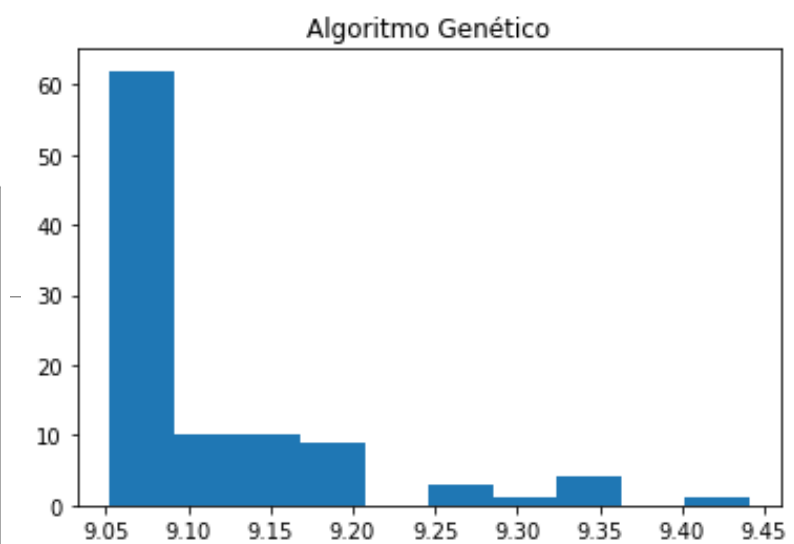
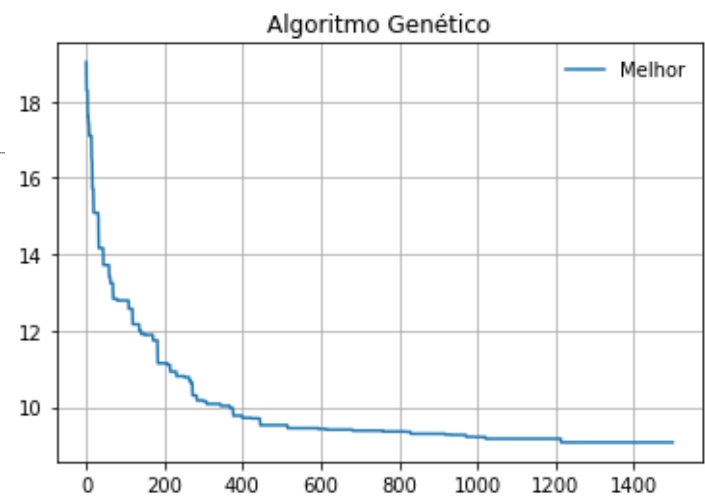
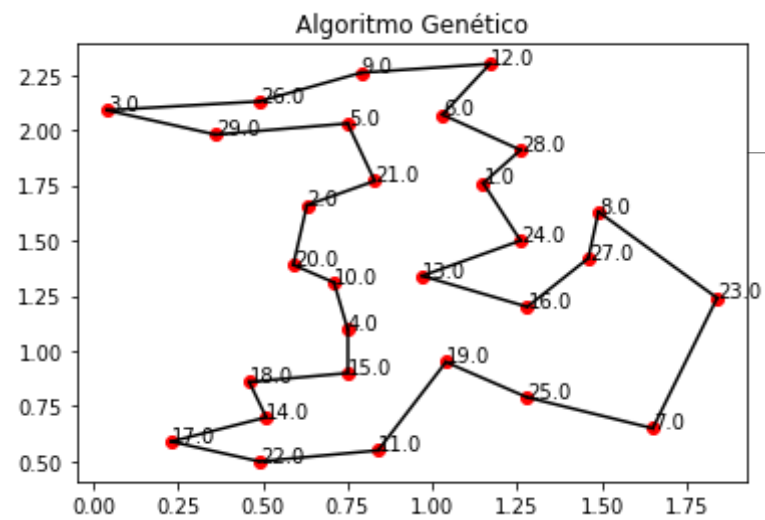


2- Parâmetros

$$J_{\text{ótimo}} = 9.074148047872841$$

Número de execuções	Tamanho da população	Número de gerações	Probabilidade de crossover	Probabilidade de mutação	Elitismo	SR GA	SR ME	MBF GA	MBF ME	AES GA	AES ME	Tempo
100	50	1500	0.9	0.001	0.2	0%	2%	11.22623	10.32745	0	0	08:22:16
100	50	1500	0.1	0.001	0.2	1%	1%	11.61428	11.33576	60029	72034.8	08:25:11
100	100	1500	0.1	0.001	0.2	14%	19%	11.03545	11.02371	120029	144034.8	18:25:29
100	100	1500	0.3	0.001	0.2	35%	45%	9.392361	9.229517	120029	144034.8	14:07:19
100	100	1500	0.5	0.001	0.2	15%	36%	10.00139	9.690392	120029	144034.8	10:55:05
100	1000	1000	0.1	0.01	0.2	35%	36%	9.14514	9.13209	800029	960034.8	09:26:17
100	80	1500	0.3	0.001	0.2	18%	32%	9.9914	9.777	96029	115234.8	06:29:29
100	60	1500	0.3	0.001	0.2	11%	19%	10.454	10.2318	72029	86434.8	04:44:25
100	80	1500	0.4	0.001	0.4	9%	29%	10.1697	9.8302	60029	72034.8	02:55:55
100	500	1500	0.3	0.001	0.2	62%	67%	9.182045	9.09013	895565.7	1074678.806	20:55:30

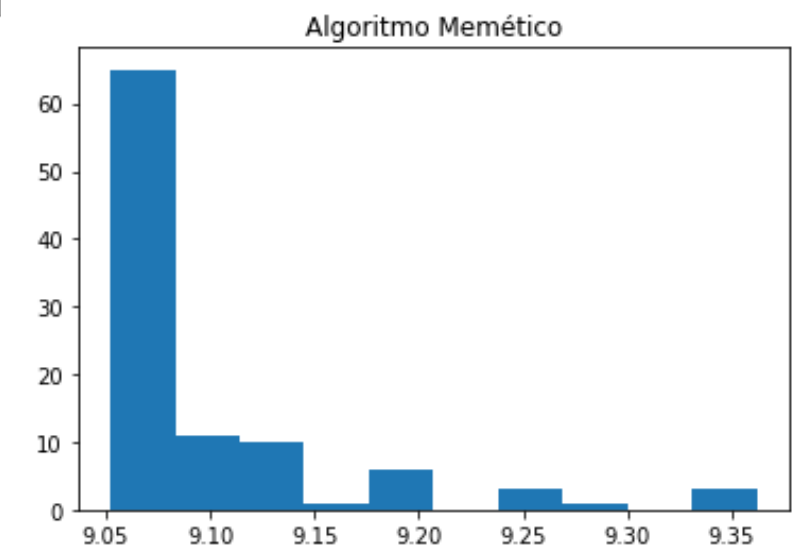
3- Comentários



	SR	MBF	AES	Tempo de execução
GA	62%	9.1820	895566	20:55:30
ME	67%	9.0901	1074679	

Solução aceita com 0.05% de variação da rota ótima

TamanhoPopulação = [500]
num_Gerações = [1500]
taxa_crossover = [0.3]
Mutação_pc = [0.001]
elitism_pc = [0.2]
num_execuções = 100



4- Conclusões

- Busca local do algoritmo memético resultou em melhoras em todas as execuções;
- Os índices SR, AES e MBF indicaram que o AM é mais adequado para esse problema;
- Foi possível determinar o conjunto de parâmetros que garantem a convergência do algoritmo;
- O aumento do tamanho da população e do número de gerações resultam em maior convergência porém aumentam o tempo de execução;
- Sugestão de trabalhos futuros: comparar com outros AEs e implementar um problema com dois caixeiros atacando o mesmo conjunto de cidades.

