

1. Introdução

O presente trabalho corresponde a comparação entre os métodos de otimização Algoritmo Genético [1] e Algoritmo Memético [2,3] aplicados para o clássico problema do Caixeiro Viajante [3], que corresponde a busca da menor distância para percorrer todos os dados pontos fixos, partindo de determinado ponto e retornando para o mesmo ponto inicial. Para isso foram implementados o *Algoritmo Genético* (AG) e o Algoritmo Memético (AM) reproduzidos no Anexo 1, na linguagem de programação Python e a convergência para diferentes parâmetros iniciais em conjunto de 29 cidades foi analisada. A população inicial destes algoritmos são um conjunto de possíveis soluções, ou seja, as cidades são visitadas em rotas aleatórias, cada uma dessas rotas serão um indivíduo da população. Na próxima geração do algoritmo ocorre recombinação e mutação destes indivíduos, de forma que em uma determinada rota posições são invertidas, criando a nova rota ou um novo indivíduo. Portanto, a função custo é a distância total de determinada rota e o objetivo do algoritmo é minimizar essa função custo.

O desempenho dos dois algoritmos será avaliado com os índices (i) Taxa de Sucesso SR (em inglês, *Success Rate*), que corresponde a porcentagem de execuções que terminaram em sucesso; (ii) MBF (em inglês, *Mean Best Fitness*) que indica a média das melhores aptidões, ou seja, os valores de melhor aptidão (*best fitness*) sobre todas as execuções e (iii) AES (em inglês, *Average Number of Evaluations to a Solution*) que é o mesmo que contar a quantidade de vezes que a função de fitness foi executada, ou seja, este número é contato para um número de execuções independentes (com sucesso): a média deste número é o AES.

Os parâmetros do algoritmo são tamanho da População (número de possíveis rotas que o algoritmo irá avaliar em uma geração), número de Gerações (quantidade gerações para a população), taxa de crossover (taxa de ocorrência da recombinação), taxa de mutação (taxa de ocorrência do processo de inversão dos bits de um indivíduo sorteado aleatoriamente) e estes serão variados, assim como o tipo de algoritmo (Genético e Memético que também será variado). O processo de elitismo será adotado para melhorar o desempenho, de forma que os melhores indivíduos sempre serão selecionados para a população seguinte.

2. Metodologia

Para este projeto optou-se por utilizar conjuntos de cidades reais, pois dessa forma, é conhecido o valor da menor distância para percorrer todos os pontos, ou seja, o ótimo global que o algoritmo deve encontrar e também é conhecida a melhor rota. O conjunto de cidades adotados estão disponibilizados na biblioteca de amostras de pontos com diferentes complexidades para o Problema do Caixeiro Viajante, esta indica as coordenadas geográficas reais entre cidades. A biblioteca é mantida pela Universidade de Heidelberg [4,5]. Para este trabalho foi escolhido um conjunto de dados com 29 cidades na Bavaria, chamado de bayg29, o mesmo conjunto de cidades já adotado no Projeto 1 e analisado com o algoritmo de *Simulated Annealing*. Esse conjunto de cidades tem como distância mínima o valor de 9.0741 unidades, para evitar problemas com números de casas decimais ou arredondamentos qualquer solução com 0.05% de variação desta rota ótima será aceita.

O procedimento de recombinação ou crossover adotado neste trabalho foi o *order crossover* [1], ideal para problemas que necessitam de uma permutação discreta e o processo de mutação consiste em inverter dois bits do cromossomo com o procedimento conhecido como *swap mutation* [1]. Ambos estão exemplificados na Fig. 1.

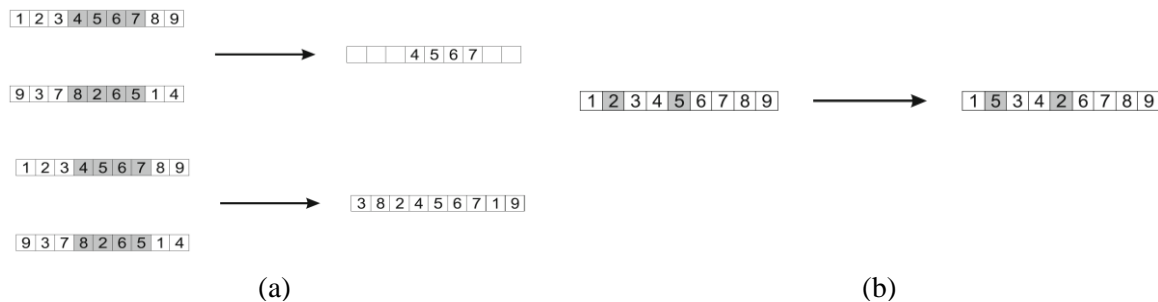


Figura 1. (a) Procedimento de recombinação e (b) mutação. Fonte: Eiben e Smith (2015)

O objetivo do presente trabalho é encontrar parâmetros que determinam a convergência do algoritmo e determinar como essa variação tem influência na convergência e nos índices de desempenho dos algoritmos. Cabe ressaltar que a capacidade de processamento do computador tem influência significativa no tempo de execução do algoritmo, neste trabalho, foi utilizado uma máquina com processador Intel® Core™ i5-8250U CPU @ 1.60GHz com 8GB de memória RAM.

O algoritmo memético implementado consiste em adicionar uma rotina no algoritmo genético tradicional, ou seja, a população sofre seleção, recombinação e mutação, porém, após a mutação é realizada também uma busca local, neste trabalho a busca local foi implementada da seguinte forma: é selecionado um indivíduo da população (que já sofreu os processos de seleção, recombinação e mutação) e este gera filhos a partir da permutação entre dois genes vizinhos, todos estes filhos tem a aptidão calculada e o filho com a menor aptidão substitui o pai. Essa estratégia tem como objetivo obter melhores soluções com o algoritmo memético do que com o algoritmo genético tradicional.

Todos os parâmetros que compõe o algoritmo genético serão variados, ou seja, o objetivo é executar 100 problemas aleatórios, com cada combinação dos parâmetros, com os dois algoritmos, gravando os dados de SR, AES e MBF. No final será indicado o conjunto de parâmetro que resulta na maior taxa de sucesso para cada um dos algoritmos para esse conjunto de dados.

2. Resultados

Primeiramente foi analisada a influência dos parâmetros na convergência com uma única execução dos algoritmos. Os resultados de uma destas execuções estão exemplificados na Fig. 2. Observa-se que o não cruzamento de linhas não é garantia de uma rota ótima, como na Fig. 2 (a) onde encontrou-se um ótimo local, porém não é a melhor rota possível. A melhor rota possível (ótimo global) para este problema de 29 cidades foi encontrada na Fig. 2 (b). A curva de progresso do algoritmo está apresentada na Fig. 2(c). Esta execução contou com os parâmetros de tamanho da população = 200, taxa de recombinação = 0.40, taxa de mutação=0.001, número de gerações = 1000.

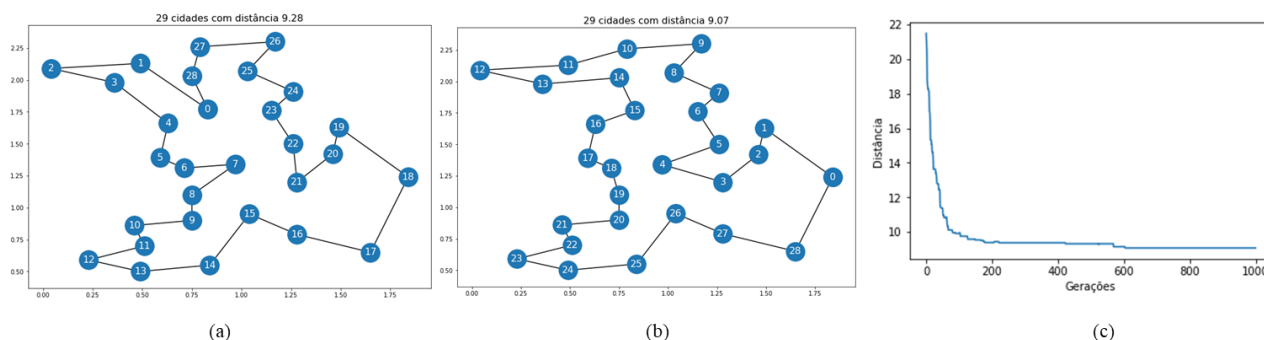


Figura 2. Resultados de (a) ótimo local de um algoritmo genético, (b) ótimo global e (c) curva de progresso

A tabela com as execuções únicas realizadas variando todos os parâmetros está reproduzida no anexo 2. Neste caso pode-se observar que quanto maior o número de gerações mais frequentemente o algoritmo converge, o tamanho da população tende a convergir com valores próximos de 100. Taxa de mutação com valores próximos a 0.001 e recombinação igual a 0.1 trouxeram bons resultados. Em todos os casos a adoção do elitismo garantiu melhores resultados.

Após compreender como cada parâmetro influencia na convergência dos algoritmos partiu-se para as 100 execuções de cada conjunto de parâmetros para analisar os índices de SR, AES e MBF. Os resultados de todas as execuções realizadas até o momento estão reproduzidos na Tab. 2, onde indica-se os todos os tempos de execuções, os valores dos índices e as variações dos parâmetros realizados com os dois algoritmos.

Tabela 1. Resultados de SR, MBF e AES

Nº de execuções	População	Nº de gerações	Prob. de crossover	Prob. 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%	11.22	10.32745	0	98920	08:22:16
100	50	1500	0.1	0.001	0.2	1%	1%	11.61	11.33576	60029	72034.8	08:25:11
100	100	1500	0.1	0.001	0.2	14%	19%	11.03	11.02371	120029	144034.8	18:25:29
100	100	1500	0.3	0.001	0.2	35%	45%	9.392	9.229517	120029	144034.8	14:07:19
100	100	1500	0.5	0.001	0.2	15%	36%	10.00	9.690392	120029	144034.8	10:55:05
100	1000	1000	0.1	0.01	0.2	35%	36%	9.145	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%	9.1820	9.09013	895565	1074679	20:55:30
100	50	1500	0.4	0.001	0.4	5%	19%	10.4888	10.16258	45029	54034.8	02:24:17

Ao analisar a Tab. 2 com os resultados vale observar que: (i) a busca local do algoritmo memético resultou em melhoras em todas as execuções; (ii) os índices SR, AES e MBF indicaram que o algoritmo memético é mais adequado para esse problema pois garante uma taxa de sucesso maior; (iii) foi possível determinar o conjunto de parâmetros que garantem a convergência do algoritmo; (iv) o aumento do tamanho da população e do número de gerações resultam em maior convergência porém aumentam o tempo de execução e o gasto computacional e (v) em todos os casos o algoritmo memético resultou em um maior índice AES, como esperado porque é um algoritmo que calcula a aptidão mais vezes.

A Fig. 3 apresenta o histograma da frequência de resultados das execuções para o conjunto de parâmetros da primeira linha da Tab. 2, lembrando que o ótimo global tem distância aproximadamente igual a 9.0741 unidades, então sabe-se que uma execução com taxa de sucesso de 100% (caso ideal) teria um histograma com apenas uma barra neste valor, porém o que se obteve com a pior execução (Fig. 3(a)) foi bem diferente do esperado, quando observa-se um histograma com uma distribuição próxima a gaussiana. Na Fig. 3(b) está indicado o histograma para a execução com o conjunto de parâmetros que obteve o melhor resultado de SR da Tab. 2 (10ª linha), neste caso os histogramas para os algoritmos genético e memético estão com a forma próxima ao esperado, encontrando o ótimo global dentre as execuções mais frequentes. Entretanto este é o caso onde observa-se também que o tempo de execução aumenta substancialmente, porém este aumento trouxe benefícios para a convergência do algoritmo.

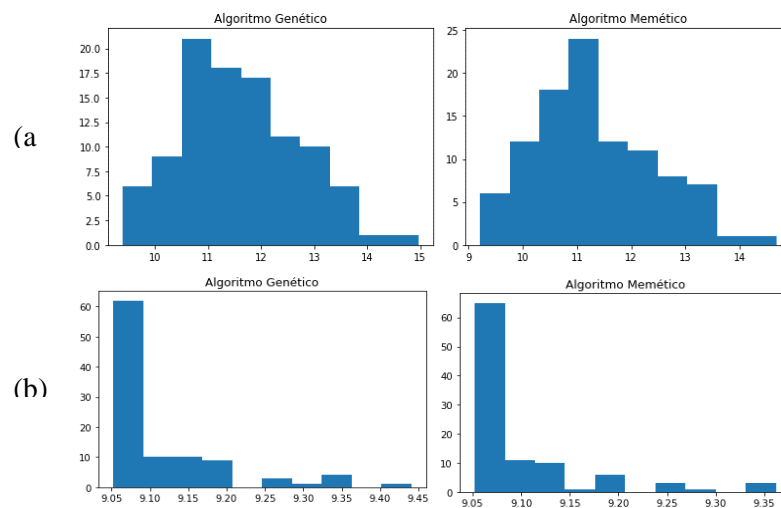


Figura 3. Histogramas dos resultados de dois diferentes conjuntos de parâmetros (a) pior execução e (b) melhor execução

3. Conclusões

Ao analisar todas as execuções do algoritmo apresentadas pode-se indicar como conclusões deste trabalho que: (i) Casos do parâmetro tamanho da população muito pequeno ou muito alto não provocam convergência, ou seja, existe uma faixa de valores que geralmente resultam no ótimo global; (ii) O aumento do número de gerações acarretam em valores melhores de SR, porém aumenta consideravelmente o tempo de processamento, pode-se dizer que para este problema o valor de 1500 gerações é adequado; (iii) os melhores valores para os parâmetros taxa de mutação e taxa de crossover encontrados foram de 0.1% e 40% respectivamente; (iv) a aplicação do algoritmo memético melhorou a taxa de sucesso em todos os casos porém causa um aumento o AES, portanto aumentando o gasto computacional.

Como conclusões finais pode-se afirmar que este trabalho encontrou bons resultados de convergência para um conjunto de dados com 29 cidades, porém com mais tempo e maiores variações de parâmetros acredita-se que a aplicação dos algoritmos genético e memético encontraria melhores resultados para problemas até mais complexos confirmando a aplicabilidade desse método de otimização. Como trabalhos futuros pode-se indicar (i) maior refinamento dos parâmetros; (ii) uma busca por outros parâmetros; (iii) ampliação da complexidade do problema para analisar melhor a eficiência dos algoritmos; (iv) comparar os resultados com outros algoritmos evolucionários e implementar um problema com dois caixeiros viajantes atacando o mesmo conjunto de cidades.

Referências:

- [1] A. E. Eiben e J. E. Smith. Introduction to Evolutionary Computing. Ed. Springer, Segunda Edição, 2015.
- [2] BURIOL, Luciana S. ; Paulo Morelato França ; MOSCATO, P. . A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem. Journal of Heuristics , USA, v. 10, n.05, p. 483-506, 2005.
- [3] BURIOL, L. ; FRANÇA, P. M. ; MOSCATO, P. . A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem. Journal of Heuristics, EUA, v. 10, n.5, p. 483-506, 2004.
- [4] Reinelt, G., 1991. TSPLIB-A traeling salesman problem library. ORSA. J. Comput., 3: 376-384. DOI: 10.1287/ijoc.3.4.376
- [5] MP-TESTDATA - The TSPLIB Symmetric Traveling Salesman Problem Instances. Disponível em: <<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/>>. Acesso em: 14 de set. de 2020.