

1. Introdução e Metodologia

O presente trabalho corresponde a adoção do método de otimização *Simulated Annealing* [1] aplicado para o clássico problema do Caixeiro Viajante [2], 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 foi implementado o algoritmo de *Simulated Annealing* (SA) reproduzido no Anexo 1, na linguagem de programação Python e a convergência para três diferentes conjuntos de cidades, aumentando a complexidade do problema, foi analisada. Primeiramente as cidades são visitadas em uma ordem ou rota aleatória na próxima rodada do algoritmo ocorre uma perturbação dessa rota de forma que duas posições são invertidas, criando a nova rota. Dessa forma, o custo (distância total da rota) atual e o custo da rota anterior são comparados e a probabilidade dessa solução ser ou não aceita segue o critério de Metrópolis [3].

Os parâmetros do algoritmo são N (número de execuções do algoritmo em uma temperatura), K (quantidade de vezes que ocorre o resfriamento) e T_0 (temperatura inicial) e estes serão variados, assim como a forma de resfriamento que também será variada, primeiro analisou-se a convergência com a função de resfriamento clássica do SA (Eq. (1)) e depois as variações dos parâmetros realizadas com a função de resfriamento adotada no *Fast Simulated Annealing* (FSA) [4], indicada na Eq. (2). Este problema será chamado de FSA modificado, uma vez que o FSA clássico além de seguir o padrão de resfriamento da Eq. (2), a perturbação das variáveis segue uma distribuição de Cauchy, o que é inviável de ser aplicado neste tipo de problema de perturbação discreta. Nas equações, k é a variação inteira de 0 até K e em cada um desses k ocorre o resfriamento da temperatura. A variação dos parâmetros ocorreu de forma que um parâmetro é modificado enquanto os outros dois ficam constantes, para os dois tipos de resfriamentos, dessa forma, pode-se avaliar a influência de cada parâmetro na convergência final do algoritmo.

$$T = T_0 / \log_2(2 + k) \quad (1)$$

$$T = T_0 / (k + 1) \quad (2)$$

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. Os três conjuntos 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 [5]. Para este trabalho foram escolhidos conjuntos de dados com (i) 29 cidades na Bavaria, chamado de bayg29; (ii) 52 pontos em Berlim, chamado de berlin52; (iii) 130 cidades, chamado de ch130 (Churritz) [6].

O objetivo do presente trabalho é encontrar parâmetros que determinam a convergência do algoritmo e como essa variação tem influência na convergência e no tempo de processamento. 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.

2. Resultados

A planilha com todos os tempos de execuções e as variações dos parâmetros realizados com o algoritmo está reproduzida no Anexo 2. Para os três problemas analisados as coordenadas foram divididas por 1000 para limitar o custo a no máximo 40, em vez de 40000, observou-se que esse artifício diminuiu o tempo de

processamento. A Fig. 1 indica o valor do custo J para o aumento da variável N , onde observa-se que com o aumento do N o algoritmo converge (a linha horizontal roxa é o valor do ótimo global), porém quanto maior o N maior o gasto computacional e o tempo de processamento. A Fig. 2 indica o valor do custo J para o aumento da variável K , onde observa-se que com o aumento de K o algoritmo também converge no valor esperado, porém novamente quanto maior o K maior o gasto computacional e o tempo de processamento. A Fig. 3 indica o aumento de $T0$, onde observa-se que para este problema existe uma região de $T0$ ótima para a convergência e valores maiores ou menores que ele, não resultam no custo esperado. Essas conclusões foram também observadas no problema com 52 cidades e 130 cidades.

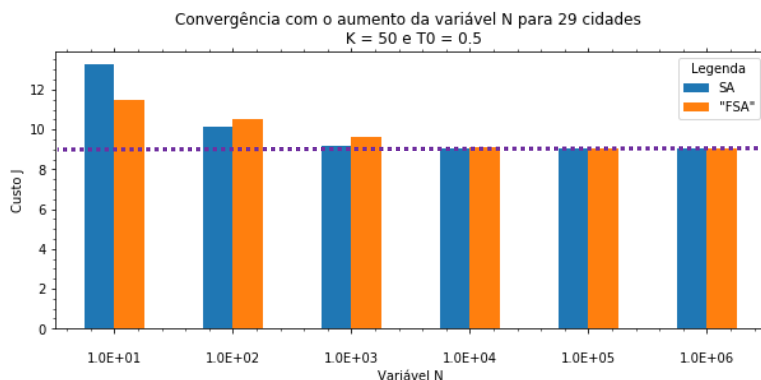


Figura 1. Análise da variação de N

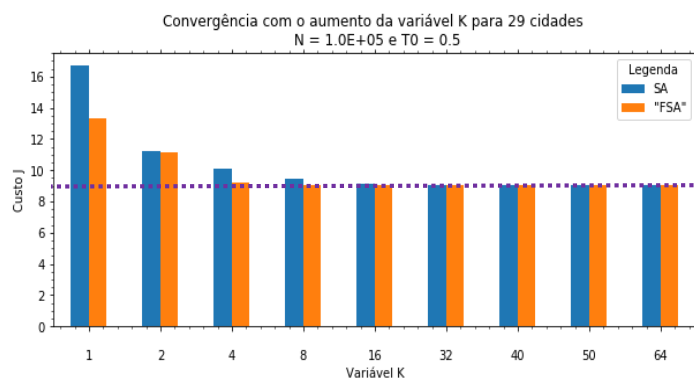


Figura 2. Análise da variação de K

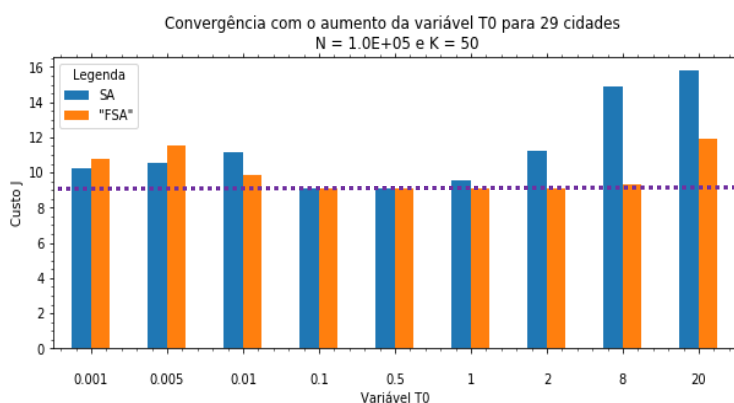


Figura 3. Análise da variação de $T0$

A Tab. 1 apresenta o número total de possíveis rotas para cada um dos problemas, cabe ressaltar que uma busca exaustiva (que calcula o custo de todas as possibilidades) demorou cerca de 5 horas para o problema com 29 cidades e não convergiu, o que comprova a aplicabilidade do SA como método de otimização que para o mesmo problema convergiu com menos de 1 minuto. A Tab. 1 também indica o menor tempo que cada um dos problemas levou para convergir, com o resfriamento clássico do SA e o resfriamento do FSA, é indicado também o tempo total gasto em todas as execuções.

Tabela 1. Análise do tempo de processamento

Nº de pontos do problema	Nº de possíveis rotas	Tempo SA	Tempo FSA modificado	Diferença do ótimo global	Total de execuções do algoritmo	Tempo gasto
29	8.84176E+30	00:00:41	00:01:03	0%	58	12:23:21
52	8.06582E+67	00:26:12	00:08:22	0%	47	06:07:55
130	6.4669E+219	05:03:37	03:15:19	8.62%	22	13:37:40

Nos itens a seguir estão indicados para cada conjunto de dados um caso de execução que encontrou o ótimo global e um caso em que não convergiu, ou seja, chegou em um ótimo local.

2.1. Problema com 29 cidades (menor distância: 9.07414804787284)

A Fig. 4 indica um exemplo de execução do algoritmo que convergiu além de indicar o histórico do custo J e da temperatura T . Custo: 9.074148047872841 e tempo de execução 00:00:41.

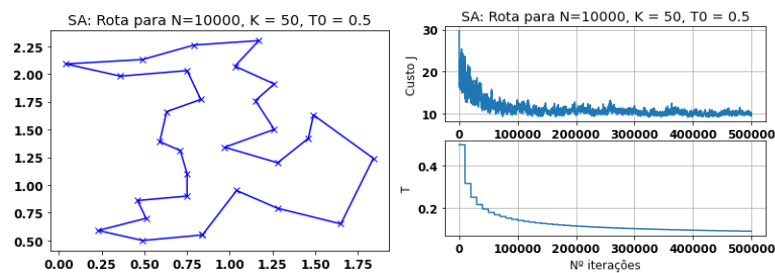


Figura 4. Convergência do problema com 29 cidades

A Fig. 5 apresenta um caso que não convergiu do problema, com custo 9.124398695923121 e tempo de execução 00:02:11.

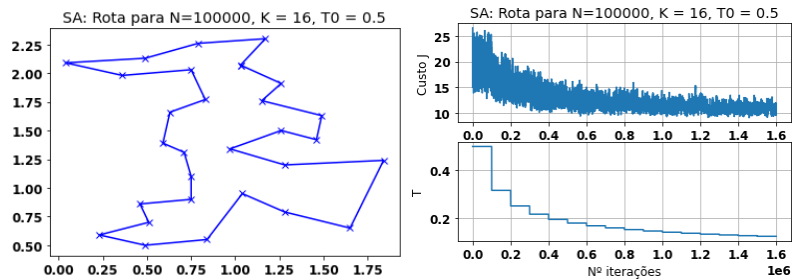


Figura 5. Ótimo local do problema com 29 cidades

2.2. Problema com 52 pontos (menor distância: 7.5443)

Na Fig. 6 está um exemplo de convergência com $J = 7.544365902$ e tempo de execução 00:26:12.

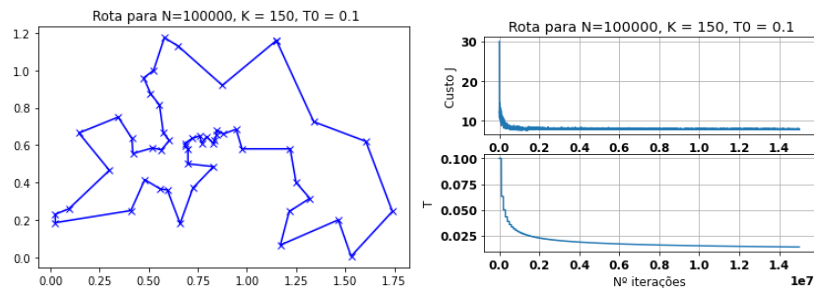


Figura 6. Convergência do problema com 52 pontos

A Fig. 7 indica um caso que não convergiu com $J = 8.826069568$ e tempo de execução 01:21:07.

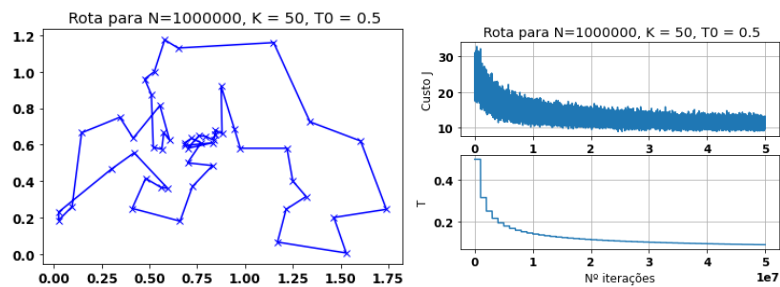


Figura 7. Ótimo local do problema com 52 pontos

2.3. Problema com 130 pontos (menor distância: 6.11086094968039)

A complexidade do problema foi aumentada para 130 pontos, porém até o momento em nenhum caso foi possível obter um ótimo global. A suspeita é que se aumentar o parâmetro N chegue-se no custo desejado. A Fig. 8 apresenta um caso que não convergiu do problema, com custo 6.538223768 e tempo de execução 05:03:37.

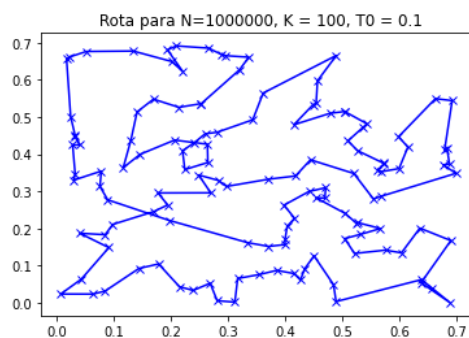


Figura 8. Ótimo local do problema com 130 pontos

3. Conclusões

Ao analisar todas as execuções do algoritmo apresentadas no Anexo 2 pode-se indicar como conclusões deste trabalho que: (i) Casos do parâmetro $T0$ muito pequeno ou muito alto não provocam convergência, ou seja, existe uma faixa de valores para $T0$ que geralmente resulta no ótimo global; (ii) O aumento de K e N geralmente resultam na convergência, porém geram um maior gasto computacional e tempo de processamento; (iii) a aplicação do FSA modificado não necessariamente resulta em uma convergência rápida. Este trabalho conseguiu bons resultados de convergência para um conjunto de dados com 130 cidades, porém com mais tempo e maiores variações de parâmetros acredita-se que a aplicação do *Simulated Annealing* encontraria melhores resultados para problemas até mais complexos confirmando a aplicabilidade desse método de otimização.

Referências:

- [1] Kirkpatrick, Scott & Gelatt, C. & Vecchi, M.. (1983). Optimization by Simulated Annealing. Science (New York, N.Y.). 220. 671-80. 10.1126/science.220.4598.671.
- [2] Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). The traveling salesman problem: a computational study. Princeton university press.
- [3] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, e E. Teller. Equation of state calculation by fast computing machines. The Journal of Chemical Physics, vol. 21, no. 6, pp. 1087-1092, junho de 1953.
- [4] H. Szu e R. Hartley. Fast simulated annealing. Physics Letters A, vol. 122, no. 3-4, pp. 157-162, junho de 1987.
- [5] Reinelt, G., 1991. TSPLIB-A traeling salesman problem library. ORSA. J. Comput., 3: 376-384. DOI: 10.1287/ijoc.3.4.376
- [6] MP-TESTDATA - The TSPLIB Symmetric Traveling Salesman Problem Instances. Disponível em: <<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/>>. Acesso em: 10 de ago. de 2020.