

# TSP: Simulated Annealing

Matheus Farnese Lacerda Senna

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais  
Belo Horizonte, MG – Brasil

## 1. Descrição do problema

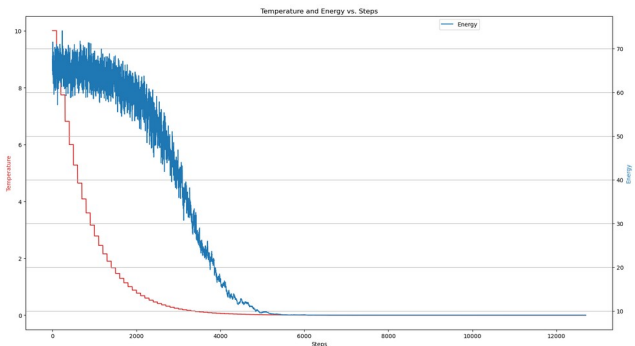
O problema do caixeiro-viajante é um problema NP-hard bem conhecido e estudado. Devido a sua natureza combinatorial, diversos métodos aproximados já foram propostos para resolvê-lo em tempo polinomial. No presente relatório, foi utilizado um modelo de Simulated Annealing para resolver o TSP de forma aproximada e rápida.

Assim sendo, o restante do relatório está organizado da seguinte forma: primeiro, foi utilizado o exemplo disponibilizado no Moodle contendo 130 cidades para plotar os gráficos da energia total e do caminho percorrido, ilustrando o funcionamento do método. Após isso, foram feitos vários experimentos variando-se parâmetros do modelo e colocando os resultados em tabelas. Finalmente, foram plotados outros gráficos de energia e caminho.

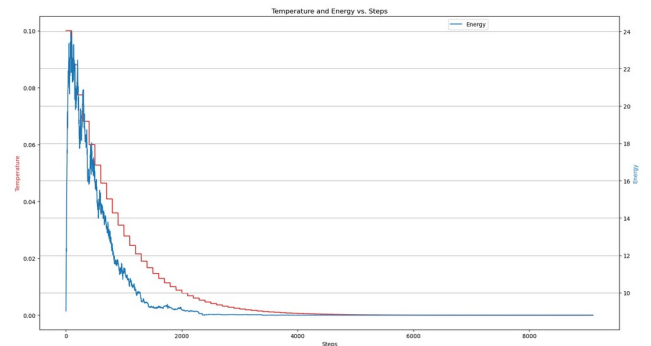
## 2. 130 cidades

Foram utilizados 100 passos de Monte Carlo em cada iteração, sendo que cada passo consiste em 100 alterações no caminho da forma proposta (i.e. o processo de escolher duas cidades e inverter o caminho entre elas é realizado 100 vezes em cada passo de Monte Carlo). Ademais, a temperatura inicial foi de 10, a final foi de 0.000001 e o decaimento foi de 0.88 a cada iteração. Finalmente, o algoritmo foi executado 4 vezes, sendo que a cada vez que foi executado, a configuração inicial era a melhor configuração obtida na execução anterior e a temperatura inicial era igual a anterior multiplicada por 0.01. Abaixo estão os gráficos representando a evolução da energia e do melhor caminho encontrado.

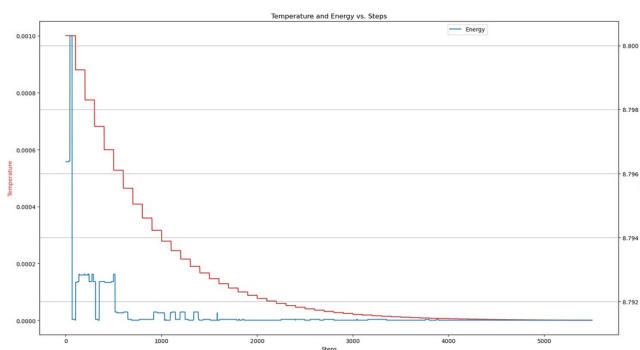
1



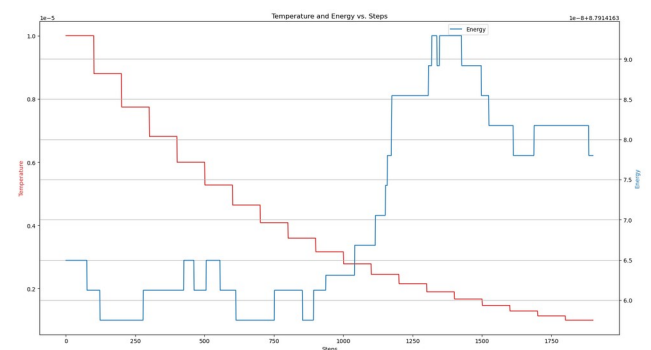
2



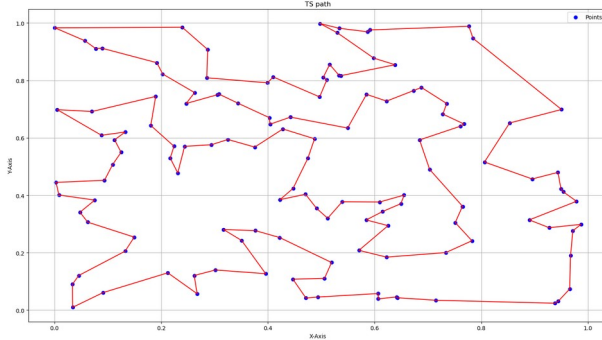
3



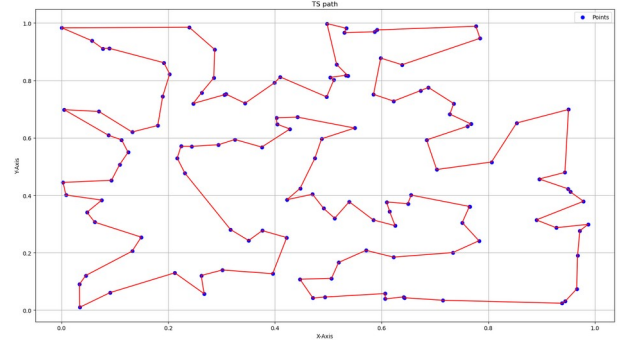
4



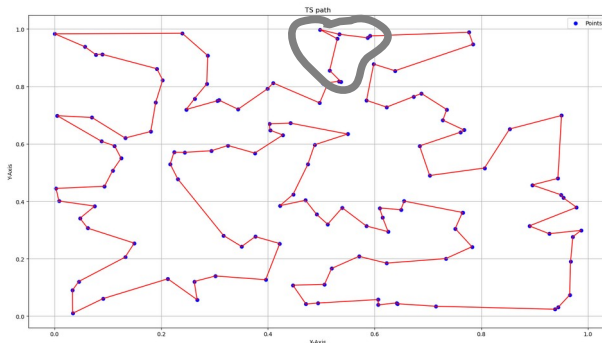
1



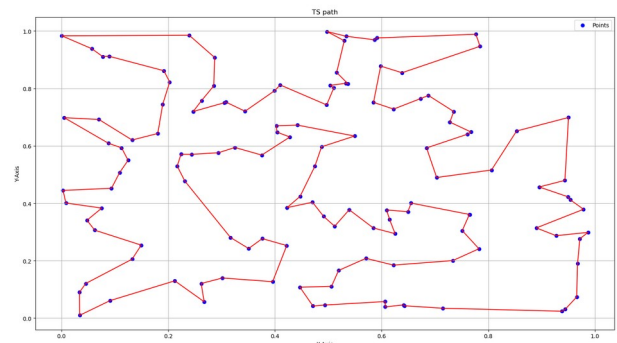
2



3



4



Observa-se que a energia apresenta o comportamento esperado no primeiro gráfico, i.e., oscila por um tempo e decai após isso. Observa-se que com 6000 passos de Monte Carlo, a energia já possui um valor próximo ao valor mínimo do gráfico. Os próximos gráficos mostram uma grande oscilação na energia devido ao fato de que a escala é 100 vezes menor a cada gráfico, mas é possível perceber que, em todos os casos, a energia final do sistema foi melhorada, isso é, foi encontrado um valor novo de energia que era menor do que o menor valor que já fora encontrado.

Com relação aos caminhos, percebe-se que houve grandes alterações do 1 para o 2, poucas alterações do 2 para o 3 (destacadas no gráfico 3) e nenhuma alteração do 3 para o 4 (sendo que a melhora na energia presente no gráfico 4 foi atribuída a erro de float na hora da computação). O comprimento dos caminhos foram:

1 →	9.023492429638281	2 →	8.796362526947632
3 →	8.791416317224503	4 →	8.791416357504204

Verifica-se que, de fato, o caminho melhorou muito do 1 para o 2, pouco do 2 para o 3 (terceira casa decimal) e permaneceu constante do 3 para o 4 (diferenças na sétima casa decimal atribuída a erros de float). A solução ótima para esse problema, como mostrado em sala de aula, é 8.210. De fato, o modelo de Simulated Annealing obteve uma boa resposta, sendo que o melhor caminho encontrado (em 3 e 4) é 7.08% pior do que a solução ótima.

### 3. Experimento: Tamanho do passo e número de repetições

Para essa seção, foram realizados experimentos com 3 quantidades de cidade (130, 500 e 1200) para verificar o efeito do tamanho do passo, i.e., quantas alterações de caminho (escolher duas cidades e inverter o caminho entre elas) são feitas por passo de Monte Carlo e da repetição da execução do algoritmo, assim como descrito na seção anterior (a cada vez que for executado, a configuração inicial é a melhor configuração obtida na execução anterior e a temperatura inicial é igual a anterior multiplicada por 0.01). O algoritmo foi executado com 100 passos de Monte Carlo e com os mesmos parâmetros de temperatura definidos na seção anterior. Os resultados encontrados estão nas tabelas a seguir, onde as linhas representam cada execução e as colunas representam o tamanho do passo. O tempo de execução está expresso em segundos e é relativo ao total das 4 execuções da respectiva coluna.

<b>130 Cidades</b>	<b>10</b>	<b>100</b>	<b>1000</b>	<b>10000</b>
<b>1ª</b>	Caminho : 9.3507548219058	Caminho : 9.0234924296382	Caminho : 8.7764748001936	Caminho : 8.7344063301570
<b>2ª</b>	Caminho : 9.2314302402082	Caminho : 8.7963625269476	Caminho : 8.7569603258743	Caminho : 8.7363315832335
<b>3ª</b>	Caminho : 9.2314302206505	Caminho : 8.7914163172245	Caminho : 8.7569729231763	Caminho : 8.7363193263299
<b>4ª</b>	Caminho : 9.2314302206505 Tempo total gasto: 0.1230599880218	Caminho : 8.7914163575042 Tempo total gasto: 0.08645355701446	Caminho : 8.7569742810446 Tempo total gasto: 8.2473311424255	Caminho : 8.7363300777506 Tempo total gasto: 82.496478080749

<b>500 Cidades</b>	<b>10</b>	<b>100</b>	<b>1000</b>	<b>10000</b>
<b>1ª</b>	Caminho : 27.927375903469	Caminho : 18.708494768943	Caminho : 17.402110207127	Caminho : 16.759436948108
<b>2ª</b>	Caminho : 24.383278620196	Caminho : 18.115895845927	Caminho : 17.241370805539	Caminho : 16.692072507459
<b>3ª</b>	Caminho : 21.952949629980	Caminho : 18.105137750506	Caminho : 17.241371791926	Caminho : 16.692079432308
<b>4ª</b>	Caminho : 21.535100365872 Tempo total gasto: 0.2276318073272	Caminho : 18.105137750506 Tempo total gasto: 1.7267880439758	Caminho : 17.241371822077 Tempo total gasto: 16.851138591766	Caminho : 16.692079646047 Tempo total gasto: 168.43468856811

1200 Cidades	10	100	1000	10000
1 <sup>a</sup>	Caminho : 90.144134638831	Caminho : 37.245630260091	Caminho : 28.289206322631	Caminho : 26.584841992822
2 <sup>a</sup>	Caminho : 72.352427348028	Caminho : 37.243656499427	Caminho : 28.148001041379	Caminho : 26.584636121406
3 <sup>a</sup>	Caminho : 63.013091108063	Caminho : 33.114993989584	Caminho : 28.147017510375	Caminho : 26.583389737061
4 <sup>a</sup>	Caminho : 60.661466252990 Tempo total gasto: 0.4764497280120	Caminho : 32.294180872733 Tempo total gasto: 3.5237612724304	Caminho : 28.147017530980 Tempo total gasto: 34.639154434204	Caminho : 26.583389598294 Tempo total gasto: 344.27824044227

O primeiro fato que se observa é o crescimento linear do tempo para o tamanho do passo, i.e., a cada vez que o tamanho foi multiplicado por 10, o tempo gasto também foi aproximadamente multiplicado por 10.

Com relação à qualidade da resposta, observa-se que o tamanho do passo de Monte Carlo tem efeito significativo no valor do caminho gerado, a saber, o pior valor de cada coluna é melhor do que o melhor valor da coluna anterior. Já o número de execuções do algoritmo, faz bastante diferença nos casos onde o tamanho do passo é pequeno (10 ou 100), principalmente para um número de cidades grande. Porém, para passos de tamanho 1000 ou 10000, não melhora a qualidade da solução de forma significativa em nenhuma das três tabelas geradas (considerando o nível de significância estatística como sendo 1%).

#### 4. Experimento: Temperaturas

O objetivo dessa seção é analisar o efeito que os parâmetros de temperatura, a saber, temperatura inicial, temperatura final e variação de temperatura a cada iteração, têm na qualidade da solução. Para tal, foram geradas três tabelas, uma para cada valor de  $\Delta T$ , onde  $\Delta T$  é o fator pelo qual se multiplica a temperatura a cada iteração. As colunas representam a temperatura inicial e as linhas representam a temperatura final. O modelo foi executado com 100 passos de Monte Carlo, tamanho do passo igual a 1000, 1200 cidades, e apenas uma execução do algoritmo.

$\Delta T = 0.88$	10	1	0.1	0.01
0.001	Caminho : 29.822289461386 Tempo: 9.8699300289154	Caminho : 29.744271049974 Tempo: 6.4170331954956	Caminho : 29.491118955193 Tempo: 4.3646006584167	Caminho : 30.826666333479 Tempo: 2.2747950553894

<b>0.0001</b>	Caminho : 28.300954514765 Tempo: 10.636663913726	Caminho : 28.506021371809 Tempo: 8.4944176673889	Caminho : 28.549278262187 Tempo: 6.4318804740905	Caminho : 28.595041638007 Tempo: 4.3486311435699
<b>0.00001</b>	Caminho : 28.023122116457 Tempo: 12.693967103958	Caminho : 28.589691163506 Tempo: 10.593178033828	Caminho : 28.234381915535 Tempo: 8.5212247371673	Caminho : 28.892225573770 Tempo: 6.4049055576324
<b>0.000001</b>	Caminho : 27.800275651621 Tempo: 14.707975864410	Caminho : 28.647646463010 Tempo: 12.699868202209	Caminho : 28.120340141700 Tempo: 10.584140539169	Caminho : 28.405742622213 Tempo: 8.4937927722930

<b><math>\Delta T = 0.704</math></b>	<b>10</b>	<b>1</b>	<b>0.1</b>	<b>0.01</b>
<b>0.001</b>	Caminho : 35.335821053013 Tempo: 3.2183279991149	Caminho : 37.104944760911 Tempo: 2.3936724662780	Caminho : 34.942919508554 Tempo: 1.7101297378540	Caminho : 39.445102193159 Tempo: 0.8965115547180
<b>0.0001</b>	Caminho : 31.719388854224 Tempo: 3.8978905677795	Caminho : 31.207864756230 Tempo: 3.2121469974517	Caminho : 31.908384674228 Tempo: 2.4052724838256	Caminho : 32.469022806035 Tempo: 1.7094757556915
<b>0.00001</b>	Caminho : 30.000820284709 Tempo: 4.7022464275360	Caminho : 30.029767874395 Tempo: 3.9007489681243	Caminho : 29.909863912966 Tempo: 3.2049238681793	Caminho : 30.431067660450 Tempo: 2.3990271091461
<b>0.000001</b>	Caminho : 29.298114453675 Tempo: 5.3974111080169	Caminho : 29.243104260647 Tempo: 4.7097604274749	Caminho : 29.138324197381 Tempo: 3.9053144454956	Caminho : 29.902489885454 Tempo: 3.2091193199157

<b><math>\Delta T = 0.5632</math></b>	<b>10</b>	<b>1</b>	<b>0.1</b>	<b>0.01</b>
<b>0.001</b>	Caminho : 39.696088291006 Tempo: 2.0486609935760	Caminho : 39.994202989386 Tempo: 1.5877089500427	Caminho : 39.705724688945 Tempo: 1.1266000270843	Caminho : 44.817377463332 Tempo: 0.6656310558319

<b>0.0001</b>	Caminho : 35.034769261721 Tempo: 2.5191876888275	Caminho : 34.393672859994 Tempo: 2.0466597080230	Caminho : 34.461184682557 Tempo: 1.5943763256072	Caminho : 35.704796522622 Tempo: 1.1319291591644
<b>0.00001</b>	Caminho : 32.002419563475 Tempo: 2.9766798019409	Caminho : 32.458706476958 Tempo: 2.5037677288055	Caminho : 31.973343986319 Tempo: 2.0512840747833	Caminho : 33.034588574431 Tempo: 1.5882101058959
<b>0.000001</b>	Caminho : 30.447120799683 Tempo: 3.4484133720397	Caminho : 30.793301996542 Tempo: 2.9791097640991	Caminho : 30.951567894080 Tempo: 2.5038168430328	Caminho : 31.371406886959 Tempo: 2.0567812919616

Percebe-se que a temperatura inicial, desde que seja pelo menos duas ordens de grandeza maior do que a temperatura final, não influencia a qualidade da resposta, sendo que os caminhos obtidos são muito parecidos entre si e oscilam em qualidade, não havendo uma correlação clara entre temperatura inicial e caminho obtido. Já a redução da temperatura final, consistentemente melhora a qualidade da solução.

Com relação a  $\Delta T$ , quanto maior ele for (i.e. quanto mais devagar a temperatura reduzir a cada iteração), melhor a qualidade da solução. Esse comportamento já era de ser esperado, pois a ideia do Simulated Annealing é, como o nome já implica, simular o processo de Annealing usado na metalurgia, processo esse que envolve o resfriamento lento da liga metálica. Portanto, quanto mais lento for a redução de temperatura do método, a tendência é que os resultados sejam melhores.

Finalmente, ao analisar-se o tempo de execução, percebe-se uma correlação entre a temperatura inicial e o tempo gasto, assim como entre o  $\Delta T$  e dito tempo. Ou seja, quanto menor a temperatura inicial e o  $\Delta T$ , menor será o tempo gasto. Já para a temperatura final, a correlação com o tempo é negativa, ou seja, quanto menor a temperatura final, maior o tempo gasto. Todos esses comportamentos estão de acordo com o esperado, haja vista que reduzir o intervalo de execução (diminuir o início ou aumentar o fim) ou acelerar o percurso de dito intervalo (aumentar o  $\Delta T$ ) vão diminuir o número de iterações, diminuindo, pois, o tempo.

## 5. Experimento: Número de passos

Como um último experimento, quer-se analisar o efeito do número de passos de Monte Carlo no resultado do modelo. Para tal, o algoritmo foi executado com os seguintes parâmetros: temperatura inicial igual a 10, temperatura final igual a 0.000001,  $\Delta T = 0.88$ , tamanho do passo 1000, 1200 cidades e uma execução apenas. Os resultados estão expressos a seguir:

100 passos → Caminho = 27.976739042438567 ; Tempo = 16.753663539886475

1000 passos → Caminho = 26.664158537168987 ; Tempo = 154.9589502811432

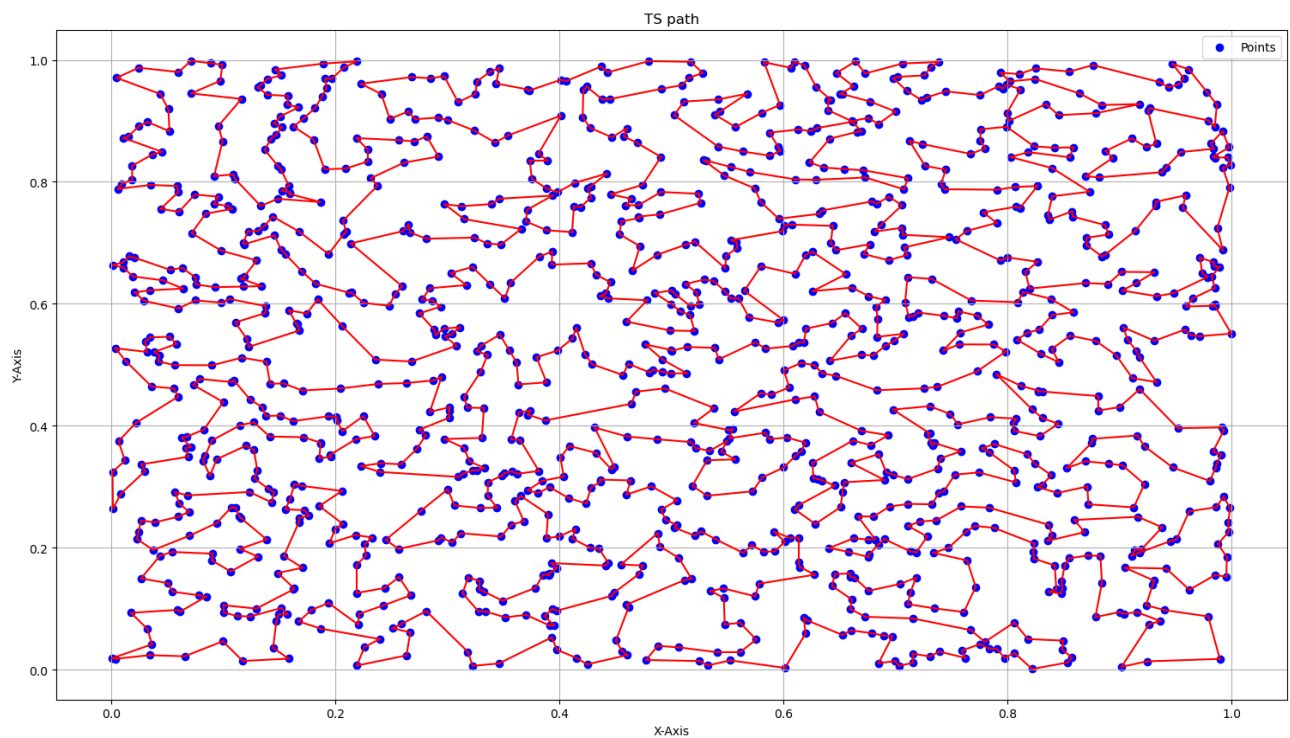
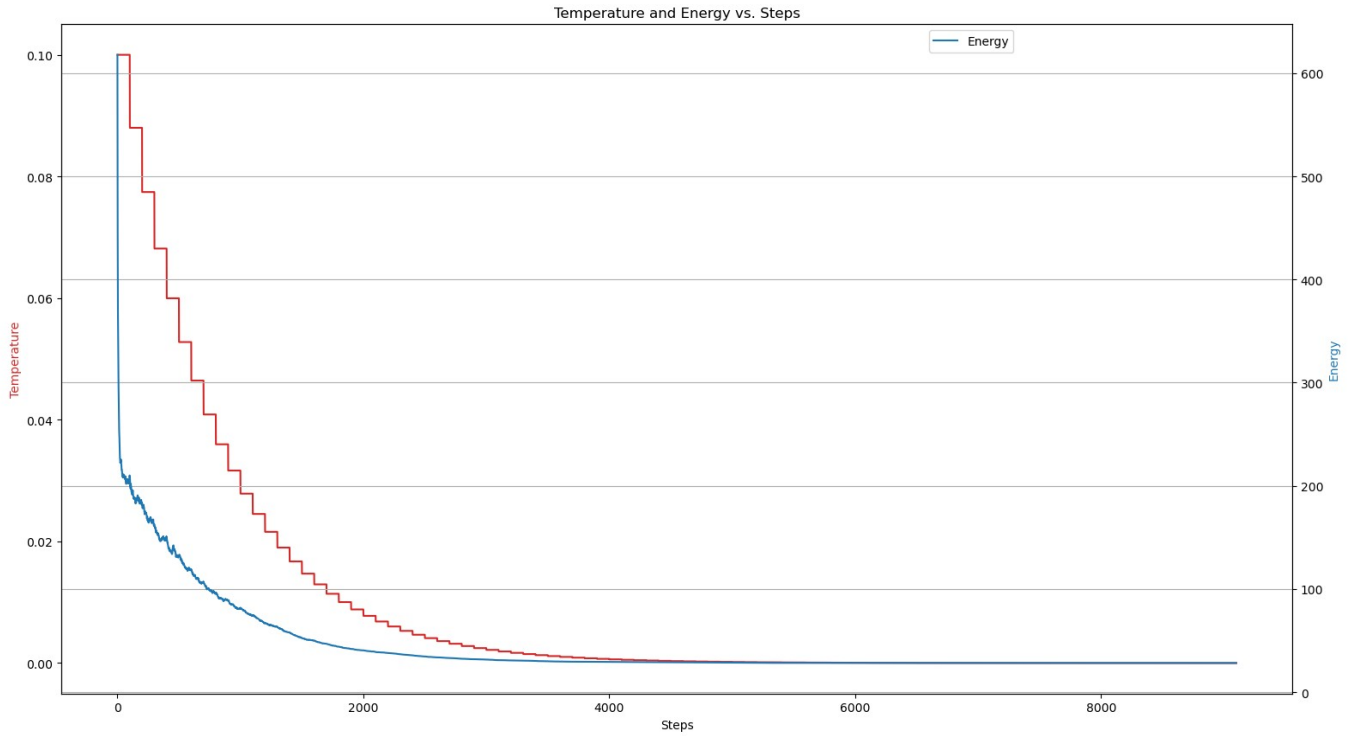
10000 passos → Caminho = 25.938709109672345 ; Tempo = 1501.3041412830353

100000 passos → Caminho = X ; Tempo = X

Para o último caso, a execução foi cancelada, pois, caso ela siga o padrão linear, demoraria 4h para acabar a execução. Percebe-se que mais passos melhora a qualidade da solução, mas o tempo gasto passa a se tornar um fator proibitivo, tendo em vista que a solução exata para o problema para 1200 cidades também está na faixa de horas. Apesar da solução melhorar com o aumento do número de passos, foi julgado (subjetivamente) que tal melhoria não compensa, devido ao alto tempo de execução.

## 6. Conclusão

Finalmente, para encerrar a análise do método de Simulated Annealing, foram escolhidos (subjetivamente) os melhores parâmetros para o modelo. Tais parâmetros foram: 100 passos de Monte Carlo de tamanho 1000,  $\Delta T = 0.88$ ,  $T_i = 0.1$ ,  $T_f = 0.000001$  e 1 execução. Assim sendo, o modelo foi rodado para o mesmo conjunto de 1200 cidades usado nos experimentos anteriores, gerando-se os seguintes gráficos:



Os resultados foram: caminho = 28.35018536041025 ; tempo = 11.978688955307007. Observa-se, pelo gráfico de energia, que o comportamento está de acordo com o esperado. Já no gráfico do caminho, apesar de ser confuso de se analisar visualmente, verifica-se que não há linhas se cruzando, o que é um bom indício de que o caminho não é muito ruim.

## **7. Apêndice**

Os arquivos de entrada de tamanho 130, 500 e 1200 estão disponíveis no arquivo zip enviado pelo Moodle. O código usado no relatório (tsp.py) e o código a ser usado na competição (competition.py) também estão disponíveis.

Para a execução do código da competição, basta chamar “python3 competition.py” na linha de comando, sendo que o arquivo “posicoes.dat” deve estar disponível no mesmo diretório.

Para a execução do código usado no relatório, basta chamar “python3 tsp.py” na linha de comando, sendo que pode-se alterar os parâmetros nas últimas linhas do arquivo para controlar a execução como quiser.