

# O Problema do Caixeiro Viajante usando Têmpera Simulada e Algoritmo Genético

Alexandre Alberto Menon<sup>1</sup>, Gabriel Rodrigues Estefanes<sup>1</sup>

<sup>1</sup>Universidade Tecnológica Federal do Paraná (UTFPR)

Curitiba – PR – Brasil

{alexandremenon,gabrielestefanes}@alunos.utfpr.edu.br

**Abstract.** *This paper presents the application of the Simulated Annealing (SA) and Genetic Algorithm (GA) heuristics to solve the Traveling Salesman Problem (TSP), a classic combinatorial optimization challenge. Simulated Annealing explores the solution space through a controlled cooling process that allows it to escape local optima. In turn, Genetic Algorithms use a population of routes that evolves through selection, crossover, and mutation operators. The study concludes that both approaches proved to be powerful and effective tools for finding high-quality solutions to the TSP.*

**Resumo.** *Este trabalho apresenta a aplicação das heurísticas de Têmpera Simulada (TS) e Algoritmos Genéticos (AGs) para resolver o Problema do Caixeiro Viajante (PCV), um desafio clássico de otimização combinatória. A Têmpera Simulada explora o espaço de soluções por meio de um processo de resfriamento controlado que permite escapar de ótimos locais. Por sua vez, os Algoritmos Genéticos utilizam uma população de rotas que evolui através de operadores de seleção, cruzamento e mutação. O estudo conclui que ambas as abordagens se mostraram ferramentas poderosas e eficazes para encontrar soluções de alta qualidade para o PCV.*

## 1. Introdução

A resolução de problemas de otimização complexos, como a do Caixeiro Viajante, que será o foco deste trabalho, frequentemente requer o uso de heurísticas capazes de explorar um vasto espaço de busca para encontrar soluções que sejam vistas como boas o bastante. Com base no problema escolhido e citado acima, as abordagens escolhidas para o desenvolvimento de soluções nesse artigo serão a da Têmpera Simulada (TS) e a de Algoritmos Genéticos (AGs). A Têmpera Simulada é um algoritmo que busca evitar mínimos locais, ao permitir com certa probabilidade, movimentos para soluções de pior qualidade. Por sua vez, os Algoritmos Genéticos são inspirados na teoria da evolução natural, trabalhando com uma população de soluções candidatas.

O presente artigo está organizado da seguinte forma: a Seção 2 apresenta uma definição formal do Problema do Caixeiro Viajante. As seções 3 e 4 detalham a fundamentação teórica e a modelagem do problema do Caixeiro Viajante com base, respectivamente, nos algoritmos da Têmpera Simulada e Algoritmos Genéticos. A Seção 5 descreve a metodologia utilizada para a implementação e teste dos algoritmos, incluindo os parâmetros e as instâncias do problema. Na Seção 6, são apresentados e discutidos os resultados obtidos por cada abordagem. Por último, na Seção 7, são relatadas as conclusões deste trabalho.

## 2. Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) consiste em determinar a rota de menor custo (ou distância) para um caixeiro que necessita visitar um conjunto de  $N$  cidades distintas. A

viagem deve obrigatoriamente começar e terminar na mesma cidade de partida, e cada uma das outras cidades do percurso deve ser visitada exatamente uma vez.

A principal dificuldade do PCV reside na explosão combinatória do número de rotas possíveis. Para exemplificar, em um cenário com apenas quatro cidades (A, B, C e D), um caixeiro precisa considerar múltiplas rotas possíveis, como ABCDA, ABDCA, ACBDA, entre outras. O número de percursos únicos cresce de forma fatorial com o aumento do número de cidades, tornando a busca pela solução ótima através da verificação de todas as possibilidades uma tarefa computacionalmente inviável para instâncias de tamanho moderado ou grande. Por essa razão, o PCV é classificado como um problema NP-difícil.

### 3. Têmpera Simulada

A Têmpera Simulada (TS) leva esse nome e tem seu funcionamento do algoritmo em alusão ao processo metalúrgico de têmpera, no qual um metal ou vidro é aquecido a uma alta temperatura e depois resfriado de forma lenta e controlada. Esse processo permite que a estrutura do material se organize em um estado de mínima energia, aumentando sua dureza e reduzindo seus defeitos. De forma análoga, o algoritmo TS inicia sua busca com um parâmetro de "temperatura" ( $T$ ) elevado, o que permite "agitar" a solução atual com força suficiente para que ela salte para fora de mínimos locais. Conforme a temperatura é gradualmente reduzida, a intensidade dessa agitação diminui, permitindo que a solução se estabilize em um mínimo global ou próximo dele.

A modelagem do algoritmo para o Problema do Caixeiro Viajante inicia-se com uma solução inicial (uma rota aleatória) e uma temperatura  $T$  elevada. A cada iteração, o algoritmo gera uma nova solução "vizinha", por exemplo, trocando a posição de duas cidades na rota atual. Em seguida, calcula-se a variação do custo (distância) entre a nova rota e a rota atual, definida por  $\Delta E = \text{Valor do novo estado (nova rota)} - \text{Valor do estado atual (rota atual)}$ . A decisão de aceitar a nova rota segue duas regras:

1. Se  $\Delta E < 0$ : A nova rota é melhor (mais curta) que a atual. Neste caso, ela é sempre aceita, e passa a ser a nova rota atual.
2. Se  $\Delta E \geq 0$ : A nova rota é pior (mais longa) ou igual à atual. Mesmo assim, ela pode ser aceita com uma probabilidade  $p$ , calculada pela função de Boltzmann:  $p = e^{\frac{\Delta E}{T}}$

Para determinar a aceitação, um número pseudo aleatório entre 0 e 1 é gerado. Se este número for menor que a probabilidade  $p$ , a pior solução é aceita; caso contrário, é rejeitada. A probabilidade de aceitar uma solução pior depende tanto da qualidade da nova solução quanto da temperatura. No início, com  $T$  alto, a probabilidade de aceitar movimentos piores é maior, incentivando a exploração. À medida que  $T$  diminui, o valor de  $p$  tende a zero, reduzindo drasticamente a chance de aceitar movimentos que piorem a solução e forçando o algoritmo a convergir para uma solução de alta qualidade.

### 4. Algoritmos Genéticos

O Algoritmo Genético (AG) baseia-se na teoria da evolução natural para resolver o problema. A busca inicia-se com uma "população" de "cromossomos", onde cada um representa uma rota candidata. Essa população evolui ao longo de gerações e, assim como na natureza, os indivíduos mais "aptos" - neste caso, as rotas de menor custo - são favorecidos no processo de reprodução. A cada geração, uma nova e potencialmente melhor população é criada a partir da anterior, por meio dos seguintes operadores:

- **Avaliação:** Para cada rota na população, calcula-se o seu custo total (distância) e, a partir dele, o seu valor *fitness*. Rotas mais curtas possuem maior *fitness*.
- **Seleção:** Os pais para a próxima geração são escolhidos através de comparações. Indivíduos são selecionados aleatoriamente da população, e aquele com o maior *fitness* é escolhido como um dos pais. Este processo favorece a seleção das melhores rotas.
- **Crossover:** Os pais selecionados combinam parte das suas soluções para criar “filhos”. No código, partes da rota de um pai são combinadas com as cidades restantes da rota do outro pai, gerando novas rotas que herdam características de ambas as soluções.
- **Mutação:** Com uma pequena probabilidade, um filho pode sofrer uma mutação, que consiste na troca de posição de duas cidades em sua rota. Este passo é crucial para introduzir diversidade na população e evitar a convergência para um mínimo local.

Além disso, a melhor solução da geração anterior é sempre transferida para a próxima (elitismo), garantindo que o progresso nunca seja perdido. Este ciclo de avaliação, seleção, crossover e mutação se repete por um número definido de gerações, refinando continuamente a qualidade das soluções.

## 5. Metodologia

Para a aplicação e teste do algoritmo da Têmpera Simulada, foi criada uma instância do Problema do Caixeiro Viajante (PCV). As coordenadas (x, y) de cada cidade foram geradas de forma pseudo aleatória, com valores uniformemente distribuídos no intervalo entre 0 e 1. A solução inicial, que serve como ponto de partida para o processo de otimização, foi definida pela sequência em que as cidades foram criadas. A qualidade de uma rota é medida pela sua distância total, calculada como a soma das distâncias Euclidianas entre as cidades consecutivas, incluindo o segmento de retorno da última cidade para a primeira, fechando o ciclo.

### 5.1 Têmpera Simulada

A modelagem do algoritmo se inicia pela configuração de seus parâmetros. A temperatura inicial, que define o ponto de partida para a exploração do espaço de soluções, foi estabelecida em 30. A partir da solução inicial, novas rotas candidatas são geradas através de um mecanismo de vizinhança baseado no movimento de troca. A cada passo, duas cidades da rota atual são escolhidas aleatoriamente e suas posições são permutadas, criando assim uma nova rota a ser avaliada.

O processo de otimização é controlado por um esquema de resfriamento. A temperatura é atualizada a cada ciclo do laço principal de acordo com a função  $T_{k+1} = T_k \cdot \alpha$ , onde o fator de resfriamento  $\alpha$  foi definido como 0.99. O algoritmo opera com dois laços aninhados: um laço interno, que explora a vizinhança da solução atual realizando 100 tentativas de troca de cidades para cada nível de temperatura; e um laço externo, que controla o decaimento da temperatura. O critério de parada principal é o número de iterações do laço externo, que foi fixado em 1000 ciclos.

O critério de aceitação de uma nova rota gerada segue a lógica padrão da Têmpera Simulada. Se a nova rota for mais curta que a rota atual, ela é automaticamente aceita. Caso contrário, se a rota for mais longa (uma piora na solução), ela ainda pode ser aceita com uma probabilidade  $p$  dada pela função  $p = e^{\frac{\Delta E}{T}}$ . A aceitação de soluções piores permite que o algoritmo evite mínimos locais e explore de forma mais abrangente o espaço de soluções.

## 5.2 Algoritmo Genético

A modelagem do Algoritmo Genético inicia-se pela configuração de seus parâmetros fundamentais. O tamanho da população foi estabelecido em 100 indivíduos, o número de gerações em 1000, e a taxa de mutação em 0,2. A população inicial, que serve como ponto de partida para o processo evolutivo, é criada gerando 100 rotas únicas através de permutações aleatórias do conjunto de cidades, garantindo assim a diversidade genética no começo da busca.

O processo de otimização é executado por um número fixo de gerações. A cada geração, a aptidão de todos os indivíduos da população é avaliada com base na distância total do percurso. A melhor solução encontrada em toda a execução é armazenada e atualizada sempre que um indivíduo de menor custo surge.

A transição para a geração seguinte é governada por um mecanismo de seleção por truncamento, no qual os 50% melhores indivíduos da população atual são selecionados para formar o grupo de reprodução. O restante da nova população é gerado a partir deste grupo de elite. Pares de pais são selecionados aleatoriamente dentre os indivíduos mais aptos e geram um descendente por meio de um operador de crossover. Este método preserva uma sequência de coordenadas de um dos pais, preenchendo as posições restantes com as coordenadas do outro genitor na ordem em que aparecem.

Cada nova solução gerada tem uma probabilidade de 20% de sofrer uma mutação de troca, na qual duas coordenadas aleatórias em seu percurso trocam de posição, introduzindo novas variações genéticas na população. O critério de parada é o número de iterações, fixado em 1000 gerações, ao final do qual a melhor solução global encontrada durante todo o processo é apresentada como o resultado da otimização.

## 6. Resultados

A avaliação de desempenho dos algoritmos foi realizada em instâncias de múltiplos tamanhos para testar sua robustez e escalabilidade. Para problemas contendo 10, 20 e 50 cidades, tanto a Têmpera Simulada (TS) quanto os Algoritmos Genéticos (AGs) apresentaram resultados excelentes e superiores aos da heurística de "Escolha Gulosa", conforme o esperado. O comportamento detalhado no artigo, para uma instância de 10 cidades (Figuras 1 e 2), é representativo desses casos de sucesso, onde ambas as heurísticas convergiram para uma solução de alta qualidade.

Contudo, ao aumentar a complexidade do problema para uma instância com 100 cidades (Figuras 9 e 10), observou-se que a solução gerada pela "Escolha Gulosa" foi superior àquelas encontradas pela TS e pelo AG com seus parâmetros iniciais. Este resultado não invalida as heurísticas, mas evidencia que a configuração padrão pode não ser adequada para espaços de busca mais amplos, levando a uma convergência prematura.

Essa limitação pode ser contornada com o ajuste dos seguintes parâmetros, na Têmpera Simulada, aumentar a temperatura inicial para permitir uma exploração mais abrangente, utilizar uma taxa de resfriamento mais lenta (fator  $\alpha$  mais próximo de 1, como 0.999) e elevar o número de iterações. Nesse sentido, no Algoritmo Genético, utilizar um tamanho de população maior para garantir a diversidade genética, executar o algoritmo por um número maior de gerações e ajustar a taxa de mutação para evitar a estagnação em ótimos locais.

Dessa forma, a performance em instâncias maiores pode ser significativamente melhorada, demonstrando a flexibilidade e o potencial dos métodos quando devidamente calibrados

## 7. Conclusões

Este trabalho investigou a aplicação das heurísticas Têmpera Simulada e Algoritmo Genético na resolução do Problema do Caixeiro Viajante. Ambas as abordagens demonstraram eficácia superior em relação ao algoritmo de Escolha Gulosa para a mesma instância proposta. O Algoritmo Genético destacou-se pela busca paralela e exploração abrangente do espaço de soluções, enquanto a Têmpera Simulada mostrou-se eficiente na exploração flexível, aceitando soluções piores para escapar de ótimos locais.

Em conclusão, tanto a Têmpera como o Algoritmo Genético são ferramentas poderosas e adaptáveis para problemas de otimização complexos como o Problema do Caixeiro Viajante, com aplicações em diversas áreas. A combinação da exploração paralela do Algoritmo Genético com a estratégia de escape de ótimos locais da Têmpera oferece uma abordagem robusta para encontrar soluções ótimas ou quase ótimas em ambientes de grande escala.

## 9. Descrição de papéis e estimativa de horas

Alexandre Alberto Menon (20 horas): Desenvolvimento e modelagem geral; Implementação do AG; Implementação da Têmpera Simulada; Implementação do Algoritmo Guloso; Documentação.

Gabriel Rodrigues Estefanes (20 horas): Desenvolvimento e modelagem geral; Implementação do AG; Implementação da Têmpera Simulada; Implementação do Algoritmo Guloso; Documentação.

## 10. Referências

MENON, A.A.; ESTEFANES, G.R. **caixeiro-viajante**. 2025. Disponível em: <https://github.com/EstefanesGabriel/caixeiro-viajante>. Acesso em: 09 set. 2025.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013.

FREDDO, A. R.; BRITO, R. C. **Implementações do Simulated Annealing para o problema do Caixeiro Viajante Simétrico**. Relatório Técnico, Universidade Federal do Paraná, Curitiba.

RODRIGUES, M. A. P. **Problema do Caixeiro Viajante: um algoritmo para resolução de problemas de grande porte baseado em busca local dirigida**. 2000. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 2000.

ENAMI, L. M. et al. **O Problema do caixeiro viajante através de algoritmo genético**. In: CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO, 9., 2019, Ponta Grossa.

# 11. Apêndice A - Ambiente com 10 coordenadas

Figura 1. Análise visual das rotas

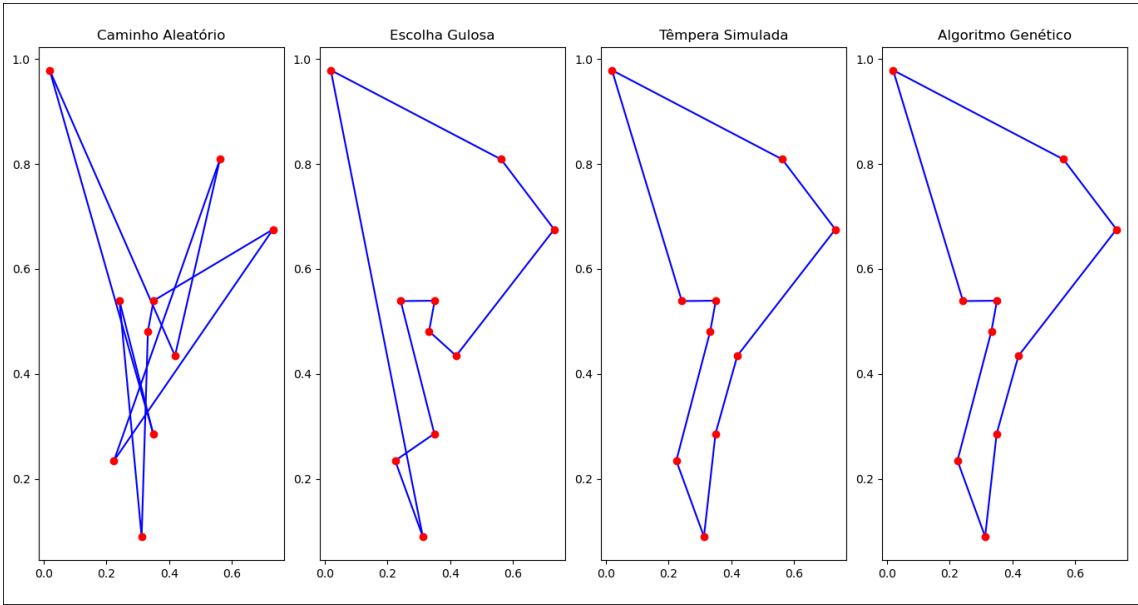


Figura 2. Gráfico de Convergência (Iterações X Distância)

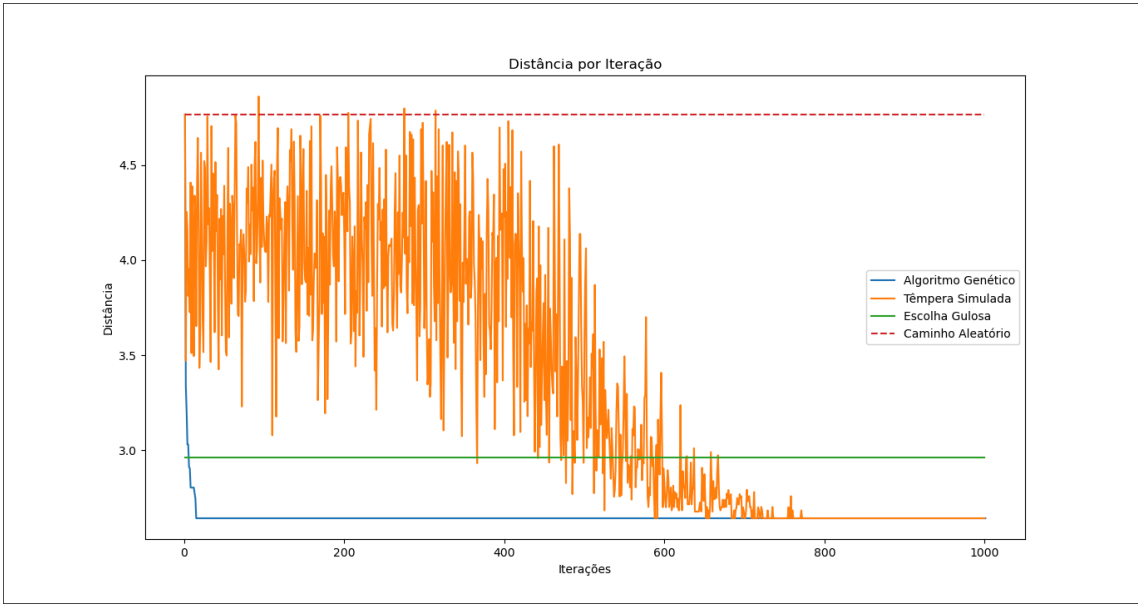


Figura 3. Análise visual das rotas

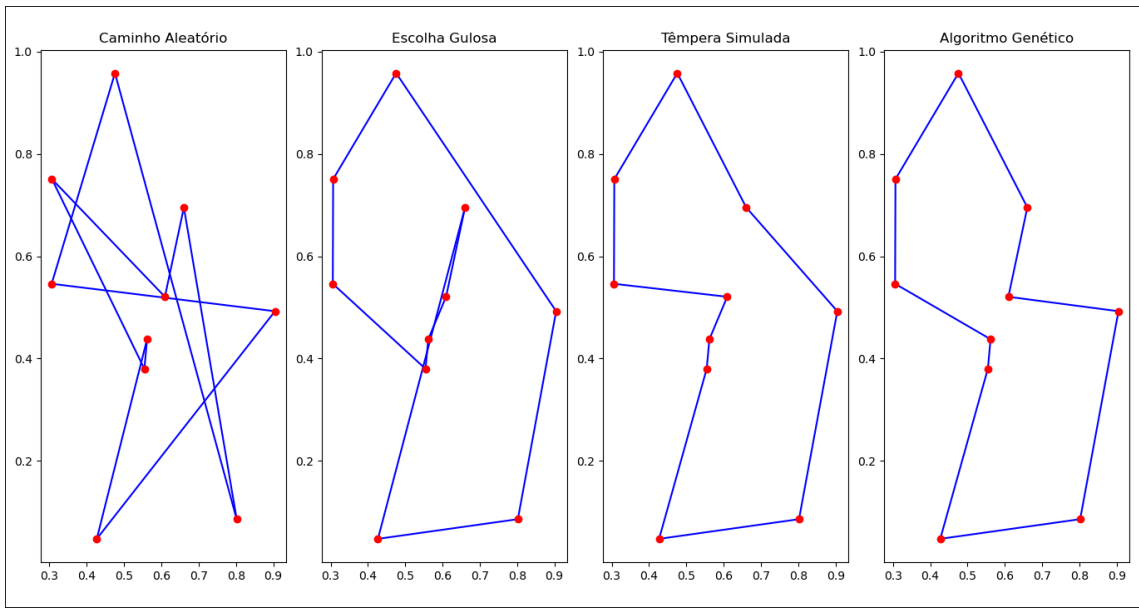
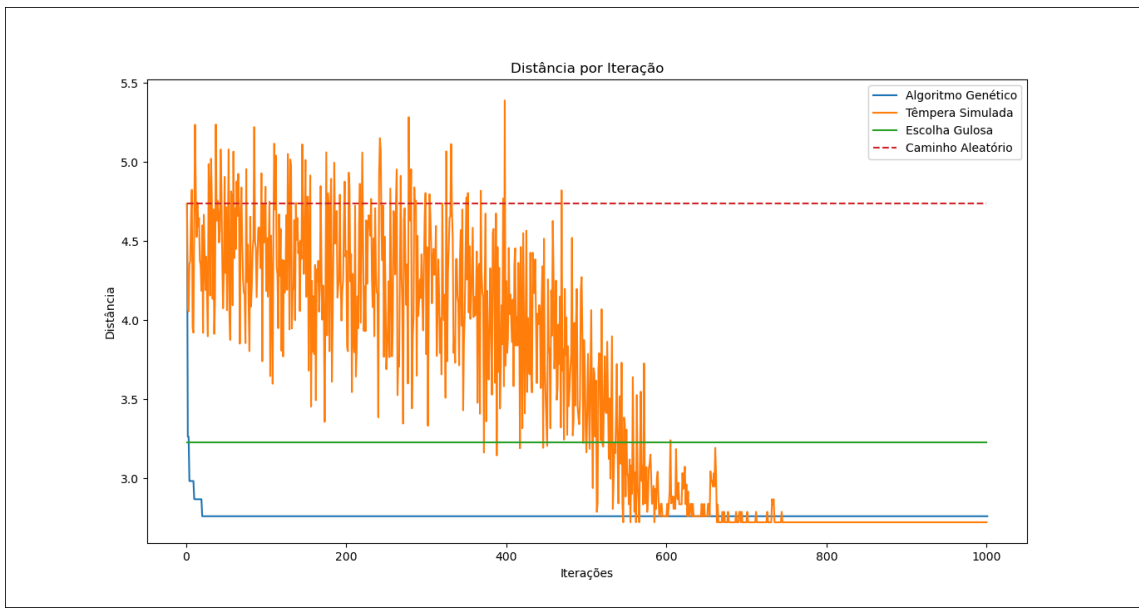


Figura 4. Gráfico de Convergência (Iterações X Distância)



## 12. Apêndice B - Ambiente com 20 coordenadas

Figura 5. Análise visual das rotas

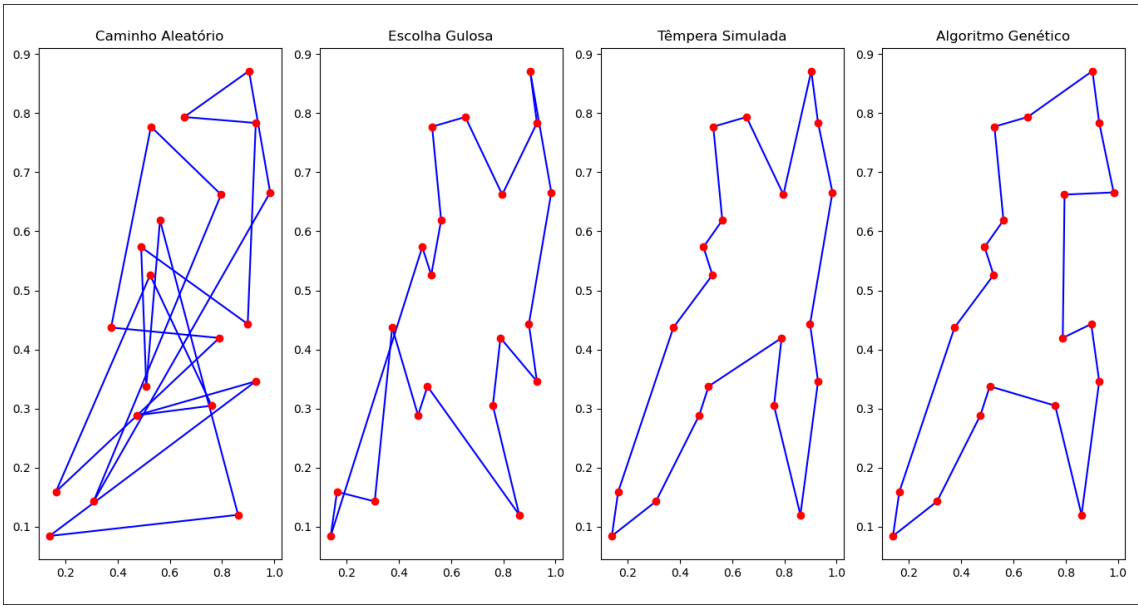
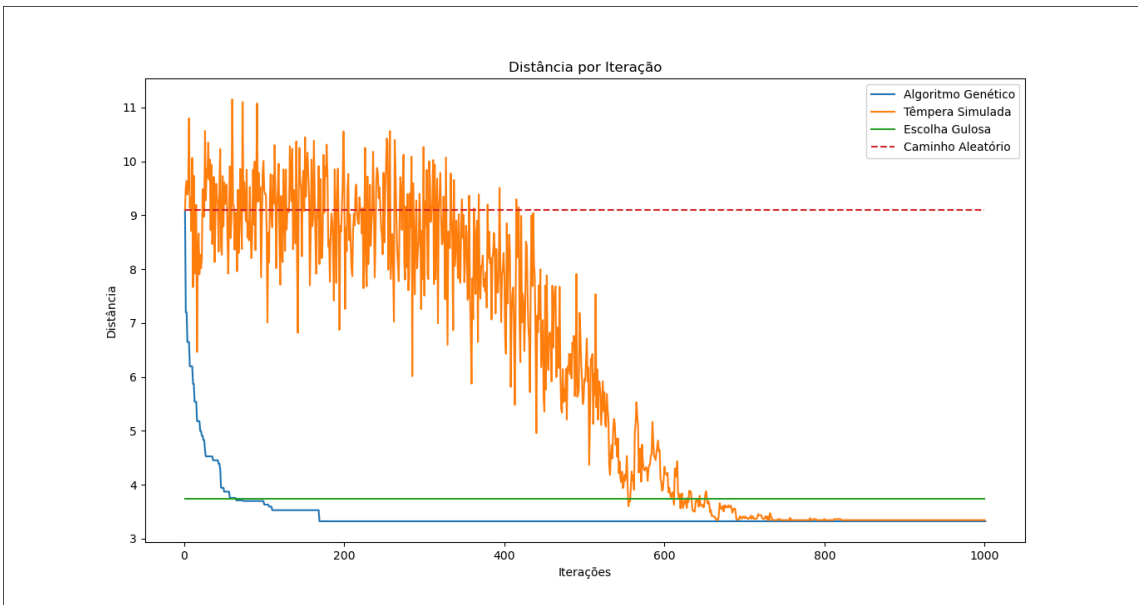


Figura 6. Gráfico de Convergência (Iterações X Distância)





## 13. Apêndice C - Ambiente com 50 coordenadas

Figura 7. Análise visual das rotas

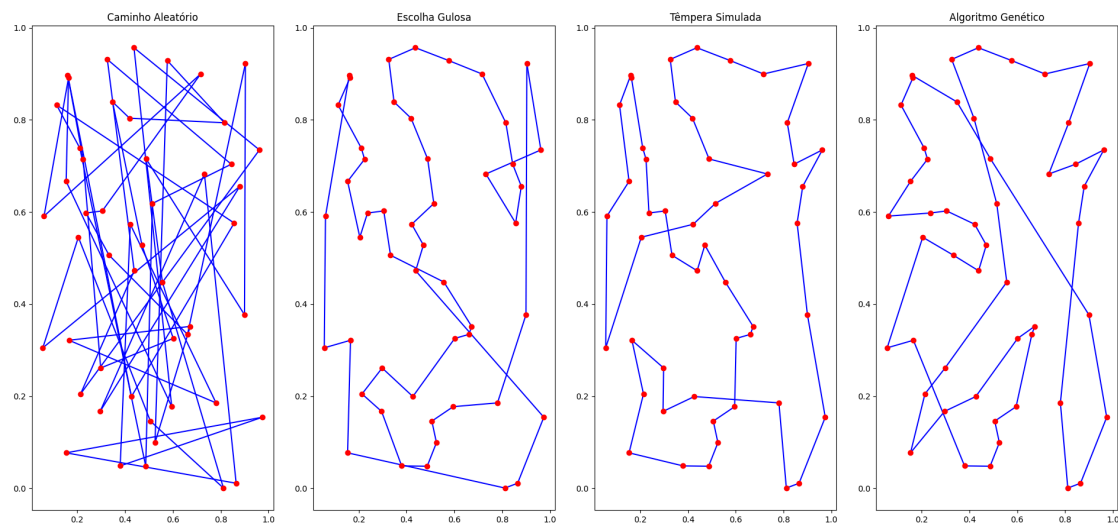
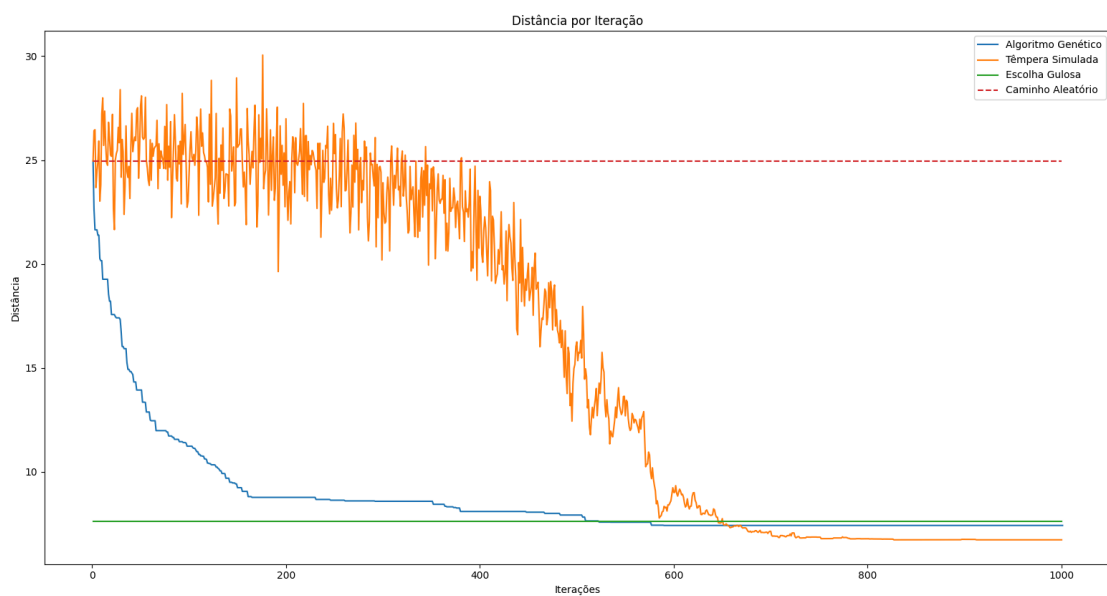


Figura 8. Gráfico de Convergência (Iterações X Distância)



# 14. Apêndice D - Ambiente com 100 coordenadas

Figura 9. Análise visual das rotas

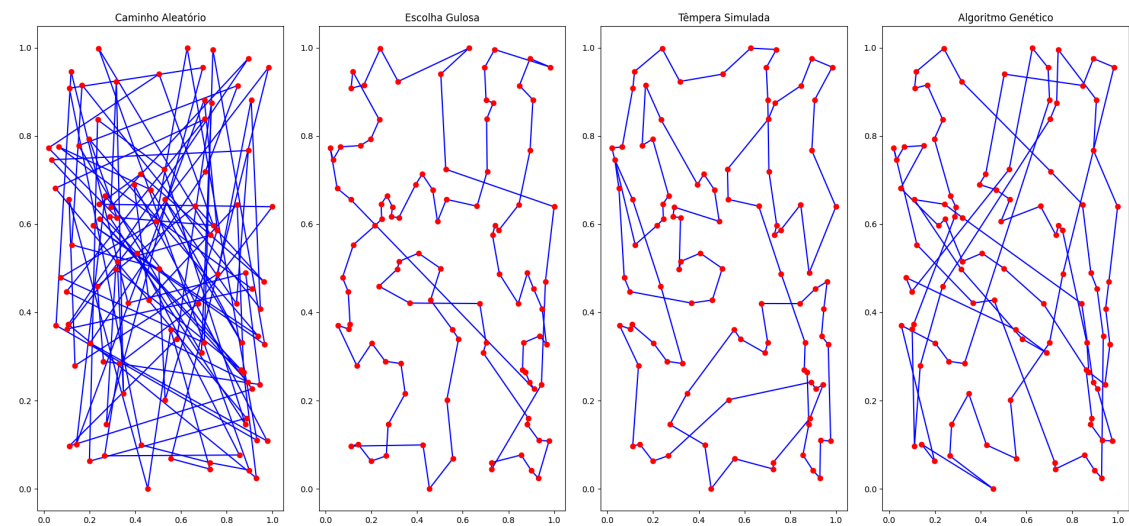


Figura 10. Gráfico de Convergência (Iterações X Distância)

