

UNIVERSIDADE FEDERAL DO PAMPA

LEONARDO PIRES RUBIM

**ALGORITMO GENÉTICO PARA O PROBLEMA DO CAIXEIRO VIAJANTE: UMA
ABORDAGEM BASEADA EM ENTROPIA PARA AVALIAÇÃO DE ‘GENES’ NO
OPERADOR DE CRUZAMENTO**

Alegrete

2025

LEONARDO PIRES RUBIM

**ALGORITMO GENÉTICO PARA O PROBLEMA DO CAIXEIRO VIAJANTE: UMA
ABORDAGEM BASEADA EM ENTROPIA PARA AVALIAÇÃO DE ‘GENES’ NO
OPERADOR DE CRUZAMENTO**

Trabalho de Conclusão de Curso (Graduação) apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Marcelo Resende Thielo

**Alegrete
2025**

G216d Rubim, Leonardo Pires

Algoritmo Genético para o Problema do Caixeiro Viajante: Uma Abordagem Baseada em Entropia para Avaliação de ‘Genes’ no Operador de Cruzamento / Leonardo Pires Rubim. – 2025.

75p. : il.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Pampa, CIÊNCIA DA COMPUTAÇÃO, 2025.

“Orientação: Marcelo Resende Thielo”.

1. Algoritmos Genéticos. 2. Problema do Caixeiro Viajante. 3. Otimização Combinatória. I. Título.

Leonardo Pires Rubim

**ALGORITMO GENÉTICO PARA O PROBLEMA DO CAIXEIRO VIAJANTE: UMA
ABORDAGEM BASEADA EM ENTROPIA PARA AVALIAÇÃO DE 'GENES' NO
OPERADOR DE CRUZAMENTO**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação
em Ciência da Computação da
Universidade Federal do Pampa como
requisito parcial para a obtenção do
título de Bacharel em Ciência da
Computação

Monografia defendida e aprovada em: 10 de dezembro de 2025.

Banca examinadora:

Marcelo Resende Thielo

Orientador

Unipampa

Claudio Schepke

Unipampa

Amanda Meincke Melo

Unipampa



Assinado eletronicamente por **AMANDA MEINCKE MELO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 10/12/2025, às 22:00, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **CLAUDIO SCHEPKE, PROFESSOR DO MAGISTERIO SUPERIOR**, em 10/12/2025, às 22:13, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **MARCELO RESENDE THIELO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 17/12/2025, às 03:02, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1919499** e o código CRC **587A1001**.

Dedico este trabalho à minha amada esposa Yasmin, por sua paciência inesgotável, pelo amor que me nutre e pelo apoio incondicional.

AGRADECIMENTO

A conclusão deste trabalho marca a realização de uma meta significativa, fruto de um esforço que contou com apoio fundamental em diversas esferas.

Primeiramente, agradeço a Deus por ter me concedido força, saúde, perseverança e a clareza necessária para trilhar este caminho e alcançar este objetivo.

Em seguida, dedico minha profunda gratidão ao meu orientador, Prof. Dr. Marcelo Resende oo, por ter confiado no potencial deste tema e pela inestimável paciência demonstrada em todas as etapas. Sua orientação precisa, conhecimento técnico e disponibilidade foram cruciais para a superação dos desafios e para a qualidade final desta pesquisa.

À Universidade Federal do Pampa (UNIPAMPA), pela estrutura, pelo ambiente de estudo e por ter sido o palco de toda minha formação em Ciência da Computação.

Ao Corpo Docente da UNIPAMPA, pelos conhecimentos transmitidos ao longo dos anos, que serviram de base sólida para o desenvolvimento deste TCC e para a minha carreira profissional.

À minha amada esposa, Yasmin Caldas Araujo Rubim, cujo suporte emocional incondicional e amor foram o meu principal alicerce e motivação nos momentos de maior dedicação.

Aos meus familiares, por toda a base, incentivo e compreensão das ausências, garantindo que eu sempre tivesse o apoio necessário para trilhar este caminho.

Aos meus amigos, em especial aos colegas de turma e de jornada, por terem tornado o ambiente acadêmico mais leve, pelas discussões técnicas e pelo companheirismo valioso.

A todos vocês, minha eterna gratidão.

”E não vos conformeis com este século, mas transformai-vos pela renovação da vossa mente, para que experimenteis qual seja a boa, agradável, e perfeita vontade de Deus.”
– (Bíblia Sagrada, Romanos 12:2)

RESUMO

O Problema do Caixeiro Viajante (PCV) é um dos grandes desafios da Otimização Combinatória, sendo classificado como NP-difícil. A complexidade de determinar a rota mais curta que visita todas as cidades exatamente uma vez torna o uso de meta-heurísticas essencial para a obtenção de soluções de alta qualidade em tempo hábil. Entre essas abordagens, os Algoritmos Genéticos (AGs) se destacam devido à sua robustez na exploração do espaço de soluções.

O presente trabalho propõe um mecanismo adicional ao processo de cruzamento do Algoritmo Genético, sem substituir ou modificar o operador de crossover clássico aplicado ao PCV. Essa estratégia incorpora um procedimento de Entropia Controlada — uma perturbação local aplicada a segmentos de rota (genes) dos indivíduos selecionados. A deterioração resultante na função de fitness após a perturbação é utilizada como indicador de robustez e qualidade daquele gene.

Genes que demonstram maior “estabilidade” sob perturbação são valorizados, aumentando a probabilidade de serem preservados ou propagados durante o processo reprodutivo do AG. O objetivo central é fornecer um mecanismo adaptativo complementar, capaz de favorecer a transmissão de “genes fortes” e reduzir a perda de características estruturais importantes ao longo das gerações.

Os resultados obtidos demonstram que o AG complementado com o Mecanismo de Entropia Controlada apresentou convergência mais rápida e evitou mínimos locais, evidenciando a eficácia da avaliação de robustez de genes como critério auxiliar ao crossover. Este estudo contribui para o campo da Computação Evolucionária ao oferecer um refinamento adaptativo que atua em sinergia com operadores tradicionais.

Palavras-chave: Algoritmos Genéticos. Problema do Caixeiro Viajante. Otimização Combinatória.

ABSTRACT

The Traveling Salesman Problem (TSP) is one of the major challenges in Combinatorial Optimization and is classified as NP-hard. The complexity of determining the shortest route that visits all cities exactly once makes metaheuristics essential for obtaining high-quality solutions within reasonable time. Among these approaches, Genetic Algorithms (GAs) stand out due to their robustness in exploring large search spaces.

This work introduces an additional mechanism applied during the crossover stage of the Genetic Algorithm, without replacing or modifying the classical crossover operator used for the TSP. The proposed strategy incorporates a Controlled Entropy procedure—a localized perturbation applied to route segments (genes) of selected individuals. The deterioration observed in the fitness function after the perturbation is used as an indicator of the robustness and quality of the affected gene.

Genes that exhibit greater stability under perturbation are favored, increasing the likelihood of being preserved or propagated during reproduction. The main goal is to provide a complementary adaptive mechanism capable of promoting the transmission of “strong genes” and reducing the loss of essential structural characteristics across generations.

Experimental results show that the GA enhanced with the Controlled Entropy Mechanism achieved faster convergence and avoided local minima, demonstrating the effectiveness of gene robustness evaluation as a supporting criterion in the crossover process. This study contributes to the field of Evolutionary Computation by offering an adaptive refinement that operates synergistically with traditional GA operators.

Keywords: Genetic Algorithms. Traveling Salesman Problem. Combinatorial Optimization.

LISTA DE FIGURAS

Figura 1 – Distribuição espacial das cidades em uma instância euclidiana do Problema do Caixeiro Viajante (<i>kroA100</i>).	32
Figura 2 – Exemplo de rota válida construída sobre o conjunto de cidades da instância <i>kroA100</i>	33
Figura 3 – Diagrama de Fluxo do Algoritmo Genético Clássico	37
Figura 4 – Representação de um indivíduo no AG aplicado ao PCV	38
Figura 5 – Demonstração do operador PMX.	41
Figura 6 – Demonstração do operador OX.	42
Figura 7 – Diagrama de Fluxo do AG-EC	51
Figura 8 – Demonstração inserção de entropia e cálculo da robustez.	52
Figura 9 – Curva de evolução do <i>fitness</i> e critério de parada por estagnação.	60
Figura 10 – Curva de convergência comparativa – instância <i>kroA100</i>	62
Figura 11 – Solução final obtida pelo AG Padrão para a instância <i>kroA100</i>	63
Figura 12 – Solução final obtida pelo AG-EC para a instância <i>kroA100</i>	64

LISTA DE TABELAS

Tabela 1	– Comparativo entre operadores de crossover aplicados ao PCV.	42
Tabela 2	– Classificação metodológica da pesquisa.	49
Tabela 3	– Parâmetros utilizados nas simulações.	56
Tabela 4	– Ambiente computacional de execução.	57
Tabela 5	– Síntese das métricas utilizadas na avaliação do AG-EC.	59
Tabela 6	– Instância do TSPLIB utilizada no experimento.	61
Tabela 7	– Resultados médios comparativos entre AG clássico e AG-EC.	61
Tabela 8	– Resultados do teste t pareado entre AG e AG-EC.	62

LISTA DE ABREVIATURAS

n.	número
p.	página
f.	folha
cap.	capítulo
v.	volume
coord.	coordenador

LISTA DE SIGLAS

AG	Algoritmo Genético
AG-EC	Algoritmo Genético com Entropia Controlada
CE	Computação Evolucionária
CPU	Central Processing Unit
EC	Entropia Controlada
NP	Não-Determinístico Polinomial
OX	Order Crossover
PCI	Placa de Circuito Impresso
PCV	Problema do Caixeiro Viajante
PMX	Partial Match Crossover
TCC	Trabalho de Conclusão de Curso
TSP	Traveling Salesman Problem
TSPLIB	Traveling Salesman Problem Library

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Problema de Pesquisa	27
1.2	Justificativa e Relevância	28
1.2.1	Limitações dos Algoritmos Genéticos Clássicos para o PCV	28
1.3	Objetivos	28
1.3.1	Objetivo Geral	28
1.3.2	Objetivos Específicos	28
1.4	Estrutura do Trabalho	28
2	FUNDAMENTAÇÃO TEÓRICA E REVISÃO DE LITERATURA	31
2.1	Otimização Combinatória e Meta-Heurísticas	31
2.2	O Problema do Caixeiro Viajante (PCV)	31
2.2.1	Formulação em Grafo	33
2.2.2	Variações do PCV	34
2.2.3	Complexidade e Métodos de Solução	34
2.3	Algoritmos Genéticos (AGs)	34
2.3.1	Componentes Fundamentais de um Algoritmo Genético	35
2.3.1.1	Representação (Codificação)	35
2.3.1.2	População Inicial	35
2.3.1.3	Função de Avaliação (Fitness)	35
2.3.1.4	Seleção	35
2.3.1.5	Crossover (Cruzamento)	36
2.3.1.6	Mutação	36
2.3.1.7	Elitismo	36
2.3.1.8	Crítérios de Parada	36
2.4	Resolução do Problema do Caixeiro Viajante com Algoritmos Genéticos	38
2.4.1	Representação dos Indivíduos	38
2.4.2	Função de Avaliação (Fitness)	39
2.4.3	Adaptação dos Operadores Genéticos	39
2.4.3.1	Crossover	39
2.4.3.2	Mutação	39
2.4.4	População Inicial e Diversidade	40
2.4.5	Desafios Específicos do PCV	40
2.5	Operadores de Crossover Aplicados ao PCV	40
2.5.1	Partial Match Crossover (PMX)	40
2.5.2	Order Crossover (OX)	41
2.5.3	Limitações na preservação de bons schemas (genes)	42
2.6	Entropia e robustez em Algoritmos Evolutivos	43
3	TRABALHOS RELACIONADOS	45
3.1	Algoritmos Meméticos (AGs Híbridos)	45
3.2	AGs Adaptativos	45
3.3	Avaliação de Schemas:	46
3.4	EAX-EDO e Diversidade Estrutural:	46
3.5	Crossovers Modificados para o PCV:	47

3.6	Entropia em Algoritmos Genéticos:	47
4	METODOLOGIA E DESENHO EXPERIMENTAL	49
4.1	Tipo de Pesquisa	49
4.2	Arquitetura do Algoritmo Proposto (AG-EC)	50
4.2.1	Pré-processamento das distâncias	50
4.2.2	O Operador de Crossover com Entropia Controlada (EC)	51
4.2.3	Avaliação da Robustez nos Pais	52
4.2.3.1	Herança da Robustez	53
4.2.4	Fluxo Geral do AG-EC	53
4.2.5	População Inicial	53
4.2.6	Estrutura Geral do AG-EC	54
4.2.6.1	Fitness e Seleção Adaptativa	54
4.3	Parâmetros e Ambiente de Execução	55
4.4	Plano Experimental e Métricas de Avaliação	57
4.4.1	Métricas de Avaliação	57
4.4.1.1	Melhor Fitness	57
4.4.1.2	Gap percentual (<i>gap%</i>)	57
4.4.1.3	Tempo de Execução	58
4.4.1.4	Gerações até a Convergência	58
4.4.1.5	Robustez média populacional ($\overline{R_m}$)	58
4.4.1.6	Métricas Estatísticas Complementares	58
4.4.2	Tratamento Estatístico dos Resultados	59
4.5	Critério de Parada	59
5	RESULTADOS E DISCUSSÃO	61
5.1	Organização dos Experimentos	61
5.2	Resultados Quantitativos	61
5.3	Análise Comparativa	62
5.3.1	Melhores soluções obtidas pelo AG Padrão e o AG-EC	63
5.4	Discussão dos Resultados	65
6	CONCLUSÕES E TRABALHOS FUTUROS	67
6.1	Conclusões	67
6.2	Limitações da Pesquisa	68
6.3	Trabalhos Futuros	68
6.4	Considerações Finais	69
	REFERÊNCIAS	71
	APÊNDICE A – REPOSITÓRIO DO CÓDIGO FONTE DO ALGO- RITMO GENÉTICO PADRÃO (AG)	73
	APÊNDICE B – REPOSITÓRIO DO CÓDIGO FONTE DO ALGO- RITMO GENÉTICO COM ENTROPIA CONTRO- LADA (AG-EC)	75

1 INTRODUÇÃO

O Problema do Caixeiro Viajante (PCV) (do inglês *Traveling Salesman Problem (TSP)*) é um dos desafios mais estudados e relevantes da Otimização Combinatória (LAPORTE et al., 2000; REINELT, 1994). Apesar de sua formulação simples — encontrar a rota de menor custo que visita todas as cidades exatamente uma vez e retorna à origem —, o PCV pertence à classe de problemas NP-difíceis, cuja complexidade cresce exponencialmente com o número de cidades. Isso torna inviável o uso de métodos exatos em instâncias de grande escala, motivando o emprego de técnicas heurísticas e meta-heurísticas.

A Otimização Combinatória tem papel central na Ciência da Computação e na Pesquisa Operacional, com aplicações em roteamento de veículos, logística, design de circuitos, mapeamento genético e planejamento de produção (JOHNSON, 1990; GUYON et al., 2003). Nessas situações, encontrar uma solução ótima ou próxima do ótimo é essencial para reduzir custos e aumentar a eficiência de sistemas complexos.

Entre as meta-heurísticas disponíveis, os Algoritmos Genéticos (AGs) destacam-se pela sua robustez e capacidade de explorar amplamente o espaço de soluções. Inspirados na teoria da evolução das espécies de Charles Darwin, os AGs simulam o processo de seleção natural, recombinando e mutando soluções para gerar indivíduos cada vez mais aptos. No contexto do PCV, os AGs têm sido amplamente aplicados e aprimorados, com diferentes variações de operadores de crossover (cruzamento), mutação e seleção.

Contudo, os operadores de crossover tradicionais — como o PMX (GOLDBERG; LINGLE, 1985) e o OX (DAVIS, 1985) — enfrentam limitações importantes. Embora garantam a validade das rotas geradas, esses métodos não levam em consideração as subestruturas de boa qualidade (chamadas schemas ou genes fortes) durante a recombinação. Isso pode levar à perda de informação genética relevante na população, prejudicando a exploração do espaço de busca.

1.1 Problema de Pesquisa

Diante desse contexto, surge a seguinte questão: A inserção de entropia local controlada durante o crossover no algoritmo genético pode ajudar a identificar indivíduos com genes relevantes e assim melhorar a propagação destes genes robustos e acelerar a convergência em Algoritmos Genéticos aplicados ao Problema do Caixeiro Viajante?

Essa pergunta orienta o desenvolvimento deste trabalho, que propõe um mecanismo de Entropia Controlada (EC) como teste de robustez para os genes de um indivíduo durante o cruzamento. A ideia é introduzir uma pequena perturbação (embaralhamento local) em segmentos de rota e observar o impacto sobre a função de fitness. Se a perturbação causar uma piora significativa, o segmento é identificado como gene forte, sendo valorizado e tendo maior probabilidade de ser propagado nas próximas gerações.

1.2 Justificativa e Relevância

1.2.1 Limitações dos Algoritmos Genéticos Clássicos para o PCV

A principal motivação desta pesquisa é aprimorar a forma como os AGs avaliam e preservam partes promissoras de uma solução. O método proposto busca transformar o crossover — tradicionalmente um operador passivo — em um mecanismo ativo de avaliação e seleção de subestruturas estáveis, aproximando o comportamento dos AGs de um processo de melhoria genético adaptativo.

Relevância científica: o trabalho contribui para o campo da Computação Evolucionária ao introduzir o conceito de Entropia Controlada como métrica de robustez genética, abrindo espaço para operadores de crossover adaptativos baseados em avaliação local. Relevância prática: os resultados podem ser aplicados em problemas reais de roteamento e planejamento logístico, nos quais soluções mais estáveis e rápidas reduzem custos computacionais e operacionais.

1.3 Objetivos

1.3.1 Objetivo Geral

Propor, implementar e avaliar um AG modificado que utiliza EC no operador de crossover, como mecanismo de avaliação e valorização do fitness de indivíduos com genes de alta robustez, visando a otimizar a solução do PCV.

1.3.2 Objetivos Específicos

- Implementar o novo operador de crossover e definir uma métrica de entropia local e robustez de genes;
- Desenvolver o Algoritmo Genético com Entropia Controlada (AG-EC) e realizar experimentos em instância kroA100 da TSPLIB;
- Comparar estatisticamente o desempenho do AG-EC com AGs que utilizam operadores tradicionais (PMX e OX);
- Avaliar o impacto da entropia na qualidade das soluções, na velocidade de convergência e na capacidade de evitar mínimos locais.

1.4 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos, além das seções introdutória e conclusiva, conforme descrito a seguir.

O **Capítulo 1** apresenta a introdução do tema, destacando o contexto dos Algoritmos Genéticos (AGs) e sua aplicação no Problema do Caixeiro Viajante (PCV). São discutidos a

motivação, os objetivos geral e específicos, a justificativa da pesquisa e a estrutura geral do documento.

O **Capítulo 2** aborda o referencial teórico que fundamenta o estudo, apresentando os principais conceitos relacionados aos Algoritmos Genéticos (AGs), aos operadores evolutivos, aos mecanismos de seleção e à formalização do Problema do Caixeiro Viajante (PCV).

O **Capítulo 3** apresenta a análise crítica do estado da arte dos Algoritmos Genéticos (AGs) em otimização combinatória. O capítulo é estruturado em torno de três abordagens principais: (i) métodos híbridos de Algoritmos Meméticos, (ii) mecanismos de AGs Adaptativos e (iii) estudos sobre a avaliação de qualidade e preservação de genes. Essa revisão estabelece o contexto para a necessidade e a concepção do AG-EC.

O **Capítulo 4** descreve a metodologia utilizada, detalhando o funcionamento do *Algoritmo Genético com Entropia Controlada* (AG-EC) proposto neste trabalho. São apresentados o operador de *crossover* modificado, o processo de cálculo e herança da robustez genética, os parâmetros experimentais, as métricas de avaliação e o plano de execução dos testes.

O **Capítulo 5** apresenta e discute os resultados obtidos a partir da execução experimental do AG-EC e do AG clássico, utilizando instâncias conhecidas da biblioteca TSPLIB. São analisados o desempenho, a convergência, a robustez média e a significância estatística dos resultados, permitindo avaliar a efetividade do método proposto.

Por fim, o **Capítulo 6** apresenta as conclusões do estudo, destacando as principais contribuições do trabalho, as limitações observadas e as propostas de continuidade para pesquisas futuras relacionadas ao tema.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DE LITERATURA

O presente capítulo estabelece os fundamentos teóricos necessários para a compreensão do problema abordado e da metodologia proposta neste trabalho. Inicialmente, são apresentados os princípios da Otimização Combinatória e o papel das meta-heurísticas na solução de problemas de alta complexidade. Em seguida, discute-se formalmente o Problema do Caixeiro Viajante (PCV), bem como os conceitos essenciais dos Algoritmos Genéticos (AGs). Também é detalhada a forma como os AGs são aplicados ao PCV, incluindo a representação das soluções, o cálculo do fitness e as adaptações necessárias aos operadores evolutivos. Por fim, realiza-se uma revisão dos principais operadores de crossover utilizados no contexto do PCV, destacando suas limitações, o que motiva o desenvolvimento da abordagem proposta neste estudo.

2.1 Otimização Combinatória e Meta-Heurísticas

A Otimização Combinatória é o ramo da matemática e da ciência da computação preocupado em encontrar um objeto ótimo em um conjunto finito de objetos discretos. A busca pelo elemento que maximiza ou minimiza uma função objetivo é um desafio onipresente em áreas como logística, inteligência artificial e engenharia de produção.

A complexidade desses problemas, como demonstrado pela teoria da NP-completude (GAREY; JOHNSON, 1979), exige que, para a maioria das instâncias de grande porte, métodos exatos sejam inviáveis devido ao tempo de execução exponencial. Em resposta a essa limitação, surgiram as meta-heurísticas.

Meta-heurísticas são métodos de alto nível que guiam um processo de busca de forma inteligente, equilibrando a exploração (busca por novas áreas do espaço de solução) e a exploração (refinamento de soluções promissoras já encontradas). Diferentemente das heurísticas simples (que são específicas e rápidas), as meta-heurísticas são genéricas e robustas, podendo ser adaptadas para diversos problemas. Segundo Talbi (2009), os métodos meta-heurísticos modernos — como Algoritmos Genéticos, Colônia de Formigas e Recozimento Simulado — evoluíram para abordagens híbridas e adaptativas, capazes de ajustar seus parâmetros dinamicamente conforme o comportamento da busca.

2.2 O Problema do Caixeiro Viajante (PCV)

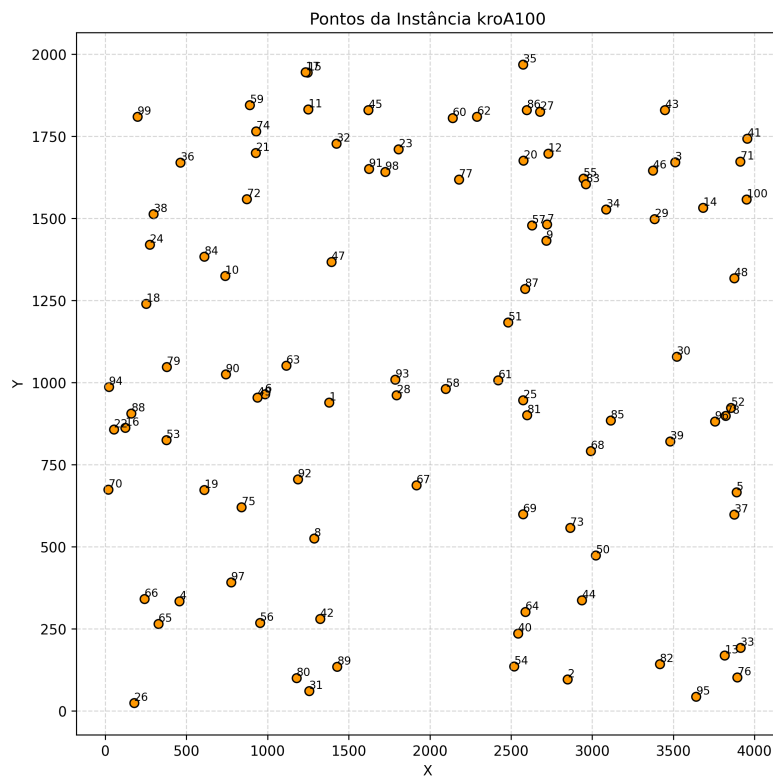
O Problema do Caixeiro Viajante (PCV), do inglês *Traveling Salesman Problem (TSP)*, é um dos problemas mais estudados da Otimização Combinatória. Desde sua formulação clássica por Dantzig, Fulkerson e Johnson (1954) e sua consolidação na literatura por Lawler et al. (1985) e Laporte (1992), o PCV tornou-se referência no estudo de algoritmos exatos, heurísticas, meta-heurísticas e teoria da complexidade computacional.

O enunciado do PCV é simples: dado um conjunto de cidades e as distâncias entre elas, determinar a rota de menor custo que visita cada cidade exatamente uma vez e retorna à cidade

inicial.

A Figura 1 ilustra visualmente um exemplo de conjunto de cidades em uma instância euclidiana do PCV. Cada ponto corresponde a uma cidade, definida por suas coordenadas bidimensionais. Nesta representação inicial não há indicação de rota, apenas o conjunto de locais a serem visitados, o que permite visualizar a distribuição espacial do problema.

Figura 1 – Distribuição espacial das cidades em uma instância euclidiana do Problema do Caixeiro Viajante (kroA100).

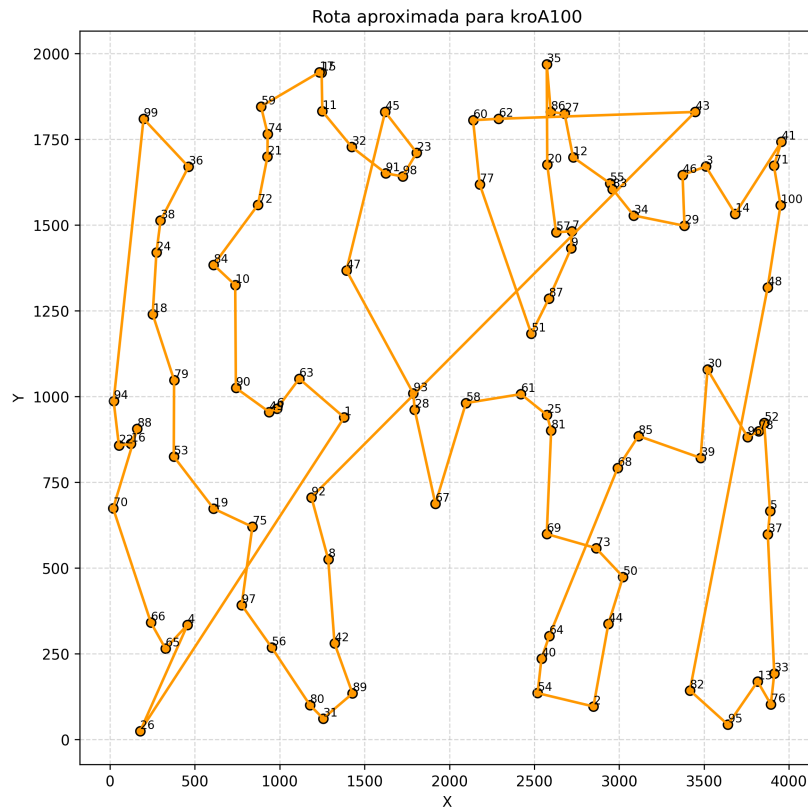


Fonte: Adaptado de Reinelt (1991).

A partir desse conjunto de cidades, uma solução para o PCV consiste em determinar uma ordem de visita que percorra todas as cidades exatamente uma vez e retorne ao ponto inicial, formando um ciclo hamiltoniano de custo mínimo. A Figura 2 apresenta um exemplo visual de uma rota válida construída sobre o mesmo conjunto de cidades. Embora não represente necessariamente a solução ótima, essa representação permite compreender intuitivamente como uma rota se organiza geometricamente no plano e como as distâncias entre as cidades influenciam o custo total do percurso.

Apesar da simplicidade da formulação, a dificuldade computacional do problema cresce rapidamente conforme o número de cidades aumenta, caracterizando-o como NP-difícil segundo a teoria de complexidade (GAREY; JOHNSON, 1979).

Figura 2 – Exemplo de rota válida construída sobre o conjunto de cidades da instância kroA100.



Fonte: Adaptado de Reinelt (1991).

2.2.1 Formulação em Grafo

O PCV pode ser modelado formalmente como um grafo completo ponderado $G = (V, E)$, onde:

- $V = \{1, 2, \dots, n\}$ é o conjunto de vértices (cidades);
- E é o conjunto de arestas conectando todos os pares de vértices;
- c_{ij} representa o custo (distância ou tempo) entre as cidades i e j .

Uma solução para o PCV é uma permutação π de V , representando a ordem das visitas às cidades. O custo total da rota é dado por:

$$C(\pi) = \sum_{k=1}^{n-1} c_{\pi_k \pi_{k+1}} + c_{\pi_n \pi_1}. \quad (2.1)$$

O objetivo é encontrar a permutação π^* que minimiza esse custo:

$$\pi^* = \arg \min_{\pi} C(\pi). \quad (2.2)$$

2.2.2 Variações do PCV

Diversas variantes do PCV surgem conforme se alteram as restrições ou o tipo de distância considerada. Entre as principais:

- **PCV Simétrico (PCVS):** $c_{ij} = c_{ji}$ para todos os pares de cidades.
- **PCV Assimétrico (PCVA):** $c_{ij} \neq c_{ji}$; comum em problemas de tráfego.
- **PCV Euclidiano (E-TSP):** as distâncias obedecem a métrica euclidiana no plano.
- **PCV com Restrições:** versões com janelas de tempo, capacidades, penalidades, etc.

No presente trabalho, são utilizadas instâncias do *TSPLIB* (REINELT, 1991), biblioteca clássica contendo variantes simétricas e euclidianas amplamente adotadas em estudos experimentais.

2.2.3 Complexidade e Métodos de Solução

A complexidade NP-difícil do PCV torna impraticável o uso de métodos exatos para instâncias de grande porte. Mesmo com avanços significativos, como o algoritmo *Concorde* (APPLEGATE et al., 2006), que resolveu o PCV de 85.900 cidades, tais métodos possuem custo computacional elevado.

Dessa forma, heurísticas e meta-heurísticas desempenham papel crucial, destacando-se:

- heurísticas construtivas, como Vizinheiro Mais Próximo;
- heurísticas de busca local, como 2-opt e 3-opt;
- meta-heurísticas, como Recozimento Simulado, Colônia de Formigas e Algoritmos Genéticos.

Estas abordagens permitem obter soluções de alta qualidade em tempo reduzido, sendo adequadas para aplicações práticas em logística, roteamento de veículos, manufatura e comunicações.

2.3 Algoritmos Genéticos (AGs)

Os Algoritmos Genéticos (AGs) são uma classe de meta-heurísticas propostas inicialmente por John Holland em 1975, baseadas nos princípios de seleção natural e evolução observados na biologia. O AG simula o processo de evolução de uma população de indivíduos (soluções candidatas) através de operadores genéticos para encontrar soluções ótimas ou quase ótimas para problemas de otimização.

O AG opera com uma população de soluções codificadas (indivíduos), cada qual possuindo um fitness (aptidão) associado. Esse fitness é determinado pela Função Objetivo do problema, indicando a qualidade daquela solução. Em problemas de minimização, como o Problema do Caixeiro Viajante (PCV), um custo menor resulta em uma fitness maior (melhor aptidão).

2.3.1 Componentes Fundamentais de um Algoritmo Genético

Os Algoritmos Genéticos (AGs) são formados por um conjunto de componentes essenciais que definem seu comportamento evolutivo. A correta formulação desses elementos é crucial para garantir que o processo de busca seja eficiente, estável e capaz de explorar adequadamente o espaço de soluções. Nesta subseção, são apresentados os principais componentes de um AG tradicional, conforme as obras clássicas de Holland (1975), Goldberg (1989) e Mitchell (1998).

2.3.1.1 Representação (Codificação)

A representação define como cada indivíduo (solução candidata) é estruturado no algoritmo. Para problemas de permutação, como o Problema do Caixeiro Viajante (PCV), utiliza-se tipicamente a representação por permutação, na qual o cromossomo corresponde a uma sequência ordenada de cidades. Essa representação preserva a restrição fundamental do PCV: cada cidade deve ser visitada exatamente uma vez. A definição apropriada da representação é essencial, pois influencia diretamente a validade dos operadores de crossover e mutação aplicados.

2.3.1.2 População Inicial

A população inicial em AGs clássicos é geralmente composta por permutações aleatórias das cidades, assegurando diversidade e ampla cobertura do espaço de busca. Entretanto, abordagens modernas frequentemente combinam indivíduos aleatórios com soluções geradas por heurísticas construtivas, como o algoritmo do Vizinho Mais Próximo (Nearest Neighbor), com o objetivo de iniciar o processo evolutivo com soluções parcialmente estruturadas.

2.3.1.3 Função de Avaliação (Fitness)

A função de avaliação mede a qualidade de cada solução. No PCV, o fitness está associado ao custo total da rota percorrida. Como o PCV é um problema de minimização, é comum utilizar a inversão do custo ou uma normalização positiva para transformar o problema em maximização de aptidão. A função de avaliação é o elemento central do AG, pois direciona a seleção dos indivíduos mais aptos para produzir descendentes.

2.3.1.4 Seleção

O operador de seleção determina quais indivíduos serão escolhidos como pais para a reprodução. Métodos tradicionais incluem:

- **Roleta:** a probabilidade de seleção é proporcional ao fitness;
- **Torneio:** indivíduos competem entre si, sendo selecionado o de maior aptidão;

- **Rankeamento:** a probabilidade é baseada na ordenação dos indivíduos.

A seleção controla a pressão evolutiva do algoritmo, influenciando diretamente o balanço entre exploração (diversidade) e exploração (refinamento de boas soluções).

2.3.1.5 Crossover (Cruzamento)

O crossover é o operador responsável por recombinar características de dois pais para gerar novos indivíduos. No PCV, deve-se garantir que a recombinação produza permutações válidas. Entre os principais operadores estão o PMX (Partial Match Crossover), OX (Order Crossover) e CX (Cycle Crossover). Esses operadores diferem na preservação de ordem, posição e adjacências, características críticas para a manutenção de schemas de alta qualidade (HOLLAND, 1975). A escolha do crossover é determinante na capacidade do AG de explorar o espaço de soluções.

2.3.1.6 Mutação

A mutação introduz variabilidade genética, evitando a convergência prematura e permitindo que o algoritmo escape de mínimos locais. Em problemas de permutação, as mutações mais comuns incluem:

- **Swap:** troca de posição entre duas cidades;
- **Inversão:** reversão de um segmento da rota;
- **Scramble:** embaralhamento de um trecho do cromossomo.

A taxa de mutação é geralmente pequena, agindo como um mecanismo refinador.

2.3.1.7 Elitismo

O elitismo assegura que o melhor indivíduo da população seja preservado na próxima geração, impedindo a perda de boas soluções devido ao caráter estocástico dos operadores evolutivos. Goldberg (1989) destaca que o elitismo acelera significativamente a convergência do AG, sendo uma das estratégias mais eficazes para manter a estabilidade do processo evolutivo.

2.3.1.8 Critérios de Parada

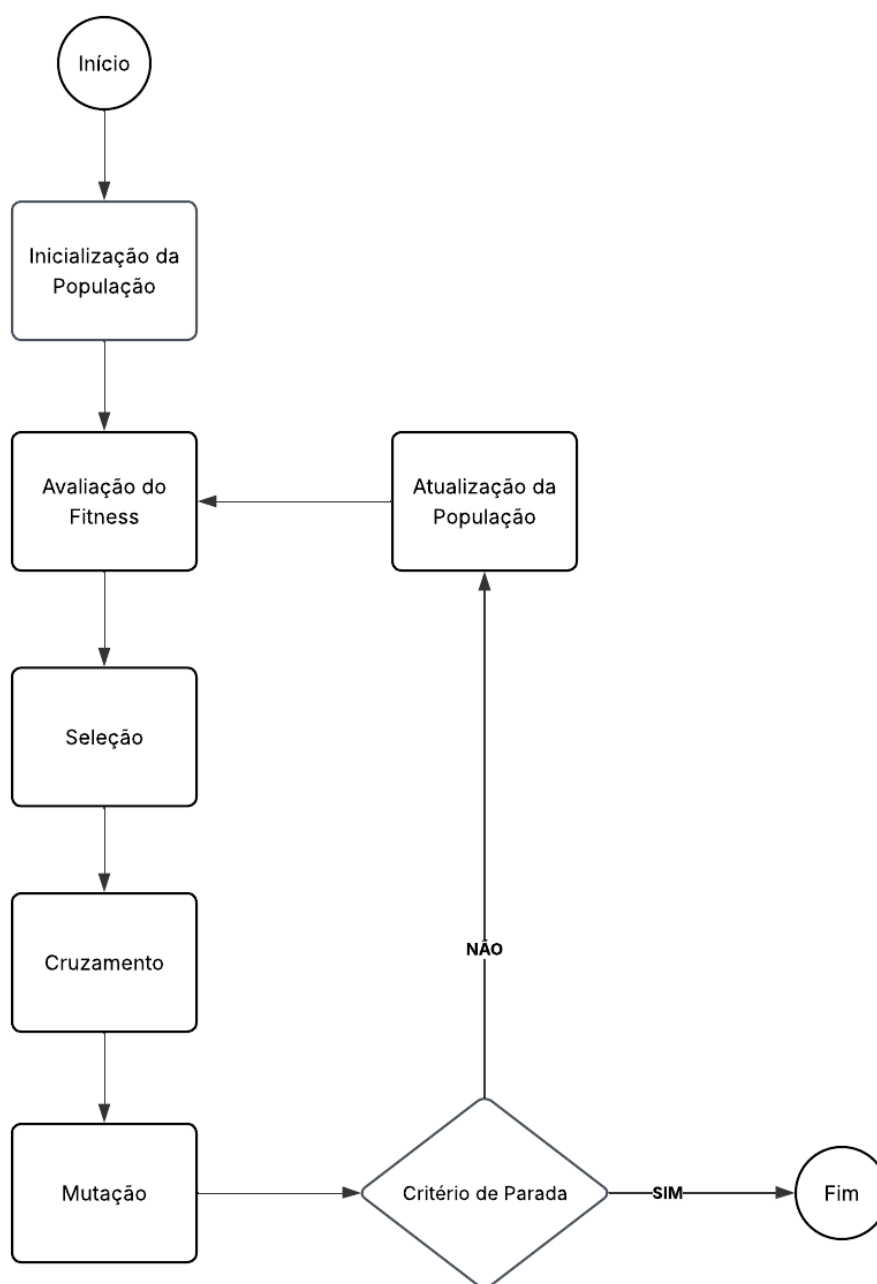
Os critérios de parada definem quando a execução do AG deve ser encerrada. Os mais comuns são:

- número máximo de gerações;
- convergência do fitness (estagnação);
- tempo máximo de execução.

A escolha adequada do critério impacta diretamente o equilíbrio entre tempo computacional e qualidade da solução obtida.

Essa estrutura geral constitui o modelo clássico de um Algoritmo Genético, servindo de base para variações adaptativas e operadores especializados, como os discutidos nos trabalhos aplicados ao PCV e detalhados nas próximas seções deste capítulo.

Figura 3 – Diagrama de Fluxo do Algoritmo Genético Clássico



Fonte: Adaptado de Holland (1975)

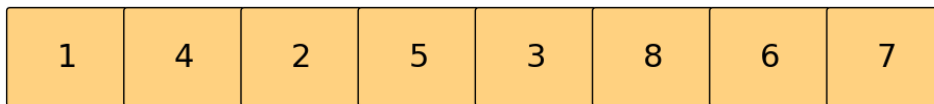
2.4 Resolução do Problema do Caixeiro Viajante com Algoritmos Genéticos

A aplicação de Algoritmos Genéticos (AGs) ao Problema do Caixeiro Viajante (PCV) exige adaptações específicas, principalmente devido à natureza combinatória do problema. Como as soluções do PCV são representadas por permutações, os operadores genéticos tradicionais — originalmente projetados para cadeias binárias — não podem ser utilizados diretamente. Esta subseção apresenta os componentes fundamentais da resolução do PCV por AGs, bem como as adaptações necessárias para garantir soluções válidas e eficientes.

2.4.1 Representação dos Indivíduos

Na abordagem baseada em Algoritmos Genéticos aplicada ao PCV, cada indivíduo da população representa uma rota completa formada por uma permutação das cidades. Essa permutação é codificada como um cromossomo linear, no qual cada gene corresponde a uma cidade e a ordem dos genes indica a sequência de visitação. A Figura 4 ilustra um exemplo dessa representação. Note que o cromossomo é apenas uma forma abstrata de armazenar a rota, enquanto o percurso resultante corresponde a um caminho fechado conectando as cidades na ordem especificada.

Figura 4 – Representação de um indivíduo no AG aplicado ao PCV



Fonte: Elaborado pelo autor (2025).

Cada indivíduo do AG corresponde a uma rota completa, representada por uma permutação das cidades:

$$I = [\pi_1, \pi_2, \dots, \pi_n],$$

onde π_k indica a cidade visitada na k -ésima posição da rota. Essa representação:

- garante que cada cidade seja visitada exatamente uma vez;
- preserva naturalmente a estrutura do PCV;
- facilita o uso de operadores de permutação.

Por exemplo, para cinco cidades, o indivíduo:

$$I = [1, 4, 2, 5, 3]$$

representa a rota $1 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 3 \rightarrow 1$.

2.4.2 Função de Avaliação (Fitness)

O fitness de um indivíduo é determinado pelo custo total da rota associada, definido como:

$$C(I) = \sum_{k=1}^{n-1} c_{\pi_k \pi_{k+1}} + c_{\pi_n \pi_1}.$$

Como o objetivo é minimizar o custo, utiliza-se frequentemente a transformação:

$$f(I) = \frac{1}{C(I)}.$$

Assim, rotas de menor custo possuem maior aptidão evolutiva.

2.4.3 Adaptação dos Operadores Genéticos

Os operadores de crossover e mutação utilizados em contextos binários não podem ser aplicados diretamente, pois gerariam rotas inválidas (com cidades repetidas ou ausentes). Por isso, operadores específicos para permutação são empregados.

2.4.3.1 Crossover

Operadores como PMX, OX e CX foram desenvolvidos especificamente para o PCV, pois:

- preservam a unicidade das cidades;
- mantêm parte da estrutura dos pais;
- preservam adjacências relevantes (schemas).

Esses operadores são discutidos em detalhe na Seção 2.5.

2.4.3.2 Mutação

Nas representações por permutação, mutações comuns incluem:

- **Troca (swap)** de duas cidades;
- **Inversão (2-opt)** de um segmento da rota;
- **Scramble** em um trecho.

Essas mutações mantêm a validade do indivíduo e promovem diversidade.

2.4.4 População Inicial e Diversidade

A população inicial no PCV pode ser formada exclusivamente por permutações aleatórias, o que maximiza diversidade e reduz viés estrutural. Contudo, heurísticas construtivas, como o algoritmo do Vizinho Mais Próximo, são frequentemente utilizadas para gerar parte da população inicial, elevando a qualidade média das soluções nas primeiras gerações.

2.4.5 Desafios Específicos do PCV

A resolução do PCV por AGs apresenta desafios característicos:

- preservação de adjacências de baixo custo;
- alta sensibilidade a trocas locais;
- risco de convergência prematura;
- necessidade de operadores especializados.

Esses desafios motivam o desenvolvimento de operadores adaptativos e mecanismos de preservação de estruturas de alta qualidade, como o operador de Entropia Controlada (EC), detalhado no Capítulo 4.

Assim, os AGs fornecem um arcabouço robusto para explorar o espaço de soluções do PCV, desde que adequadamente ajustados à natureza permutacional do problema.

2.5 Operadores de Crossover Aplicados ao PCV

O Crossover (Cruzamento) é o operador mais crítico no PCV, pois a recombinação deve preservar a validade do indivíduo (garantir que todas as cidades apareçam uma única vez) e, idealmente, transmitir bons schemas (segmentos de rota de baixo custo).

A dificuldade do PCV reside na sua natureza de ordenamento, onde um gene (cidade) só faz sentido em relação ao seu vizinho. Os operadores de crossover clássicos foram criados para lidar com essa restrição, mas frequentemente com sacrifício na qualidade da herança.

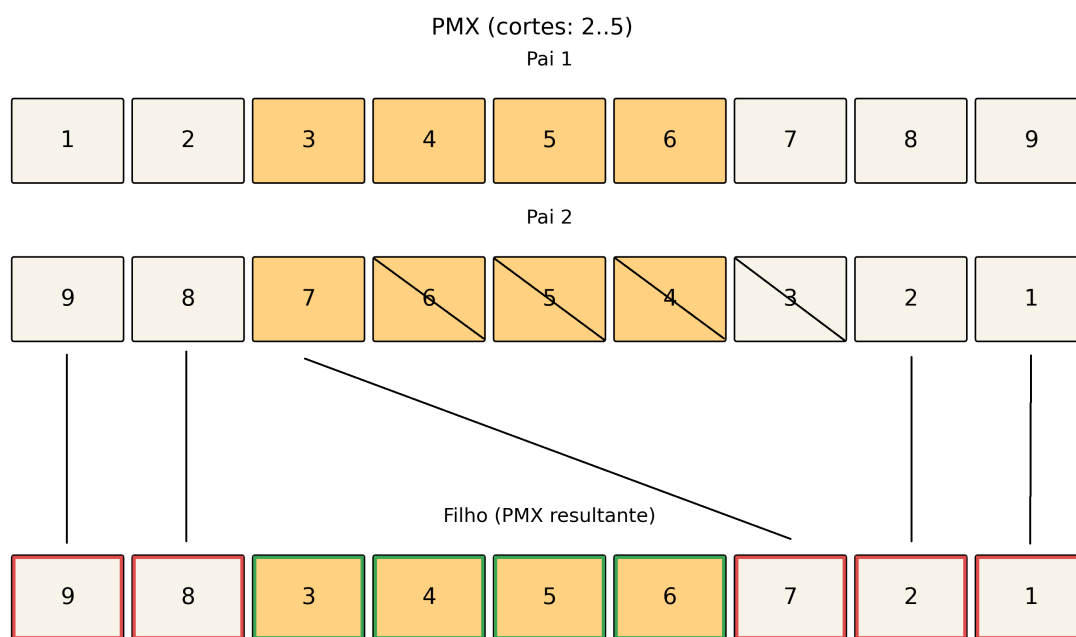
Dois dos operadores mais utilizados são o PMX e o OX.

2.5.1 Partial Match Crossover (PMX)

Proposto por Goldberg e Lingle (1985), o PMX é um dos primeiros a garantir a validade da rota. Ele seleciona um segmento do primeiro pai, mapeia as cidades não selecionadas no segundo pai e usa essa correspondência para preencher as lacunas do filho. O PMX é conhecido por sua alta preservação de posição e é eficiente, mas sua complexidade de mapeamento pode torná-lo computacionalmente pesado e propenso a falhar na preservação de bons schemas que se estendem além do segmento de mapeamento.

A Figura 5 ilustra esse processo, destacando o segmento selecionado e o efeito do mapeamento na formação do indivíduo descendente.

Figura 5 – Demonstração do operador PMX.



Fonte: Adaptado de Goldberg e Lingle (1985).

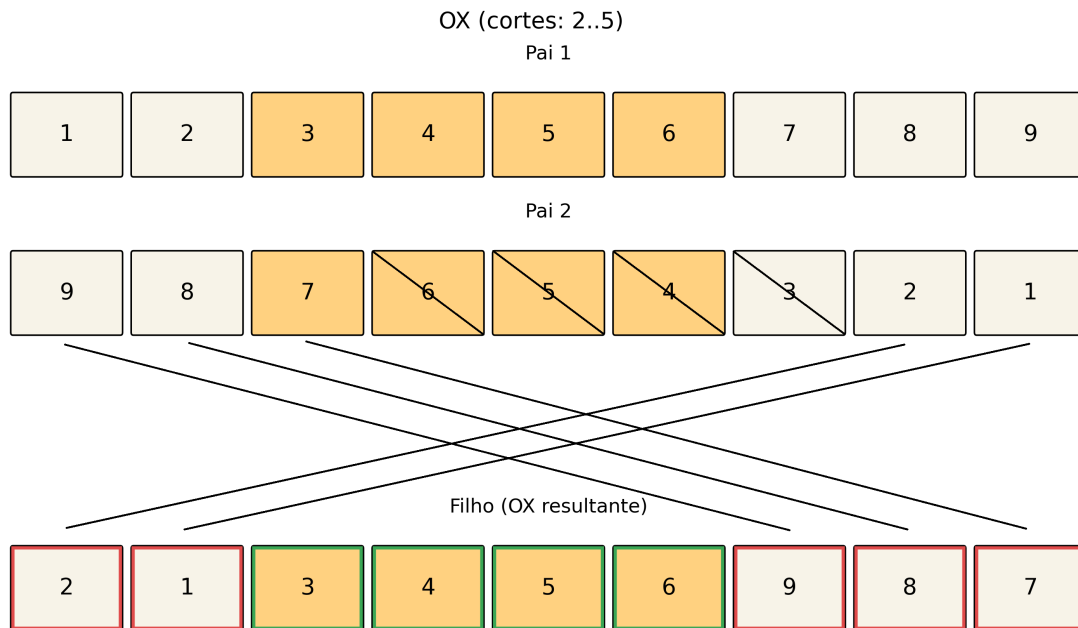
2.5.2 Order Crossover (OX)

O OX (DAVIS, 1985) concentra-se em preservar a ordem relativa das cidades. Ele seleciona um segmento de um pai e preenche o restante da rota no filho seguindo a ordem relativa das cidades no outro pai. Embora seja mais simples de implementar do que o PMX, ele tem uma preservação de posição menor e é notório por quebrar ligações (adjacências) que contribuem para a solução de baixo custo.

A Figura 6 exemplifica graficamente esse processo.

A análise dessas técnicas revela o dilema central: a busca por operadores que sejam válidos (garantindo a permutação) sem serem excessivamente destrutivos.

Figura 6 – Demonstração do operador OX.



Fonte: Adaptado de Davis (1985).

Tabela 1 – Comparativo entre operadores de crossover aplicados ao PCV.

Operador	Preservação de Ordem	Vantagem Principal	Desvantagem Principal
PMX (Goldberg e Lingle, 1985)	Alta	Garante validade da rota e preserva a posição dos genes.	Apresenta maior complexidade e sensibilidade à escolha do segmento.
OX (Davis, 1985)	Média	Simples e rápido de implementar.	Pode quebrar adjacências úteis na rota.

Fonte: Elaborado pelo autor (2025).

Apesar de suas vantagens, esses operadores não avaliam explicitamente a qualidade de um segmento de rota antes de recombiná-lo, o que pode resultar na destruição de genes valiosos. Essa limitação motiva a proposta de um operador adaptativo apresentado no Capítulo 4.

2.5.3 Limitações na preservação de bons schemas (genes)

O conceito de schemas (ou building blocks) (Holland, 1975) é fundamental para a teoria dos Algoritmos Genéticos. Um schema é um padrão de fitness acima da média, que representa uma subestrutura de uma solução ótima (no PCV, um pequeno segmento de rota de baixo custo).

O Teorema dos Schemas prevê que os schemas curtos, de baixa ordem e com alta fitness têm maior probabilidade de serem preservados e combinados para formar soluções melhores.

A limitação crítica dos AGs reside na fragilidade desses schemas de alta qualidade durante o Crossover. Os operadores tradicionais, ao realizar a recombinação, frequentemente quebram as adjacências perfeitas (os bons schemas) do indivíduo, resultando na perda de informação genética valiosa e na redução do potencial de convergência. É este dilema entre a exploração (criada pelo crossover) e a preservação dos schemas que justifica a busca por um mecanismo mais inteligente de recombinação.

2.6 Entropia e robustez em Algoritmos Evolutivos

A entropia, em um contexto informacional, mede o grau de desordem ou incerteza em um sistema (SHANNON, 1948). Quando aplicada a Algoritmos Evolutivos, ela pode representar a diversidade da população ou a estabilidade de uma subestrutura (HERNÁNDEZ et al., 2019).

Neste trabalho, o conceito é reinterpretado como uma forma de avaliar a robustez de um gene frente a pequenas perturbações locais, servindo como critério para preservar segmentos de alta qualidade durante o crossover.

3 TRABALHOS RELACIONADOS

A necessidade de aprimorar o desempenho dos AGs para problemas complexos impulsionou o desenvolvimento de diversas abordagens que tentam mitigar as falhas dos operadores tradicionais. Nesta seção, são discutidos estudos que exploram variações e melhorias em AGs aplicados à otimização combinatória, estabelecendo o estado da arte sobre o qual este trabalho se fundamenta.

3.1 Algoritmos Meméticos (AGs Híbridos)

A obra de Cotta, Cotta, Mathieson e Moscato (2016), presente no livro “Handbook of Heuristics”, consolida de forma abrangente o paradigma dos Algoritmos Meméticos como uma classe de métodos híbridos que integra mecanismos evolutivos baseados em população com heurísticas de busca local. Nesse modelo, a exploração global promovida pelos Algoritmos Genéticos é combinada com operadores intensificadores, como 2-opt, 3-opt ou buscas de vizinhança especializadas, permitindo refinar sistematicamente as soluções geradas a cada geração. Essa sinergia torna os métodos meméticos particularmente eficazes em problemas de otimização combinatória como o PCV, uma vez que aliam diversidade populacional à melhoria estrutural local. Entretanto, apesar do desempenho elevado, tais abordagens apresentam limitações notáveis: o custo computacional tende a crescer significativamente conforme o tamanho da instância aumenta, e o refinamento pós-crossover não impede que subestruturas promissoras sejam destruídas durante a recombinação. Dessa forma, embora forneçam uma base robusta de hibridização, esses métodos não oferecem mecanismos explícitos para avaliar ou preservar a relevância genética de segmentos internos da rota, evidenciando uma lacuna que motiva abordagens capazes de considerar diretamente a robustez dos genes no processo de cruzamento.

3.2 AGs Adaptativos

Uma abordagem influente no avanço dos Algoritmos Genéticos é apresentada por Srinivas e Patnaik (1994), que propõem mecanismos adaptativos capazes de ajustar dinamicamente parâmetros como taxa de crossover e taxa de mutação ao longo do processo evolutivo. O objetivo central desses métodos é mitigar problemas clássicos como convergência prematura e perda de diversidade, permitindo que o algoritmo responda ao estado global da população em cada geração. Essa adaptação contínua possibilita um equilíbrio mais robusto entre exploração e intensificação, evitando estagnação em ótimos locais sem recorrer a intervenções manuais. Apesar de seu caráter inovador, tais mecanismos operam predominantemente em nível macroscópico, ajustando parâmetros globais com base em métricas agregadas de performance. Não há, portanto, uma avaliação direta da relevância ou da estabilidade de segmentos específicos das soluções, o que limita o impacto dessas adaptações sobre a preservação estrutural durante o crossover. Assim, embora os AGs adaptativos contribuam para maior flexibilidade

evolutiva, não oferecem instrumentos para identificar e proteger subestruturas relevantes no PCV, característica essencial buscada em abordagens focadas na robustez genética local.

3.3 Avaliação de Schemas:

Os trabalhos baseados no Teorema dos Schemas, originalmente apresentado por Holland (1975), fornecem a base teórica clássica para compreender o comportamento dos Algoritmos Genéticos e a importância dos chamados blocos construtores. Segundo esse arcabouço teórico, segmentos curtos, de baixa ordem e alta aptidão tendem a ser propagados ao longo das gerações, influenciando diretamente a qualidade da convergência. No entanto, a aplicação prática dessa teoria ao PCV apresenta desafios significativos, uma vez que a representação por permutação torna a noção de schema fortemente dependente das adjacências entre cidades. Operadores de crossover tradicionais, como PMX, OX e variantes relacionadas, frequentemente rompem essas adjacências durante a recombinação, dificultando a manutenção de estruturas eficientes. Além disso, o Teorema dos Schemas não fornece um método operacional para medir a qualidade ou a robustez de um schema específico em problemas de otimização combinatória, restando como uma estrutura conceitual mais ampla do que uma ferramenta prática. Assim, apesar de seu papel fundamental na literatura de AGs, os estudos baseados em schemas carecem de mecanismos aplicáveis ao nível local do problema, especialmente em contextos como o PCV, onde a estabilidade de segmentos depende de relações topológicas que não são diretamente capturadas pelo formalismo de Holland.

3.4 EAX-EDO e Diversidade Estrutural:

Nikfarjam et al. (2021) apresentam o método EAX-EDO, uma extensão do operador Edge Assembly Crossover (EAX) projetada para produzir conjuntos de tours simultaneamente diversos e de alta qualidade no PCV. A abordagem combina o EAX com técnicas de *Evolutionary Diversity Optimisation*, permitindo explorar múltiplas regiões promissoras do espaço de busca sem sacrificar desempenho. Os autores demonstram que, ao integrar critérios explícitos de diversidade ao processo de recombinação, o algoritmo preserva adjacências relevantes e reduz a convergência prematura, evidenciando a importância da estrutura interna das rotas na evolução. Diferentemente das técnicas tradicionais, que se concentram exclusivamente na solução de melhor aptidão, o EAX-EDO valoriza a manutenção de variedade estrutural entre os indivíduos, tornando o processo evolutivo mais robusto a ótimos locais. Essa perspectiva dialoga diretamente com o Algoritmo Genético com Entropia Controlada (AG-EC) proposto neste trabalho, no qual a robustez de segmentos é avaliada antes da herança para evitar a destruição de subestruturas essenciais. Assim, ambos reforçam a relevância de operadores informados pela estrutura do problema, destacando que a preservação de componentes críticos é fundamental para melhorar a eficiência evolutiva em problemas de permutação.

3.5 Crossovers Modificados para o PCV:

Hussain et al. (2017) propõem um Algoritmo Genético aplicado ao PCV utilizando um operador de crossover do tipo ciclo modificado, introduzido com o objetivo de melhorar a qualidade das soluções e reduzir a ocorrência de tours inconsistentes. O trabalho demonstra que pequenas alterações funcionais em operadores clássicos podem gerar ganhos significativos de desempenho, especialmente em problemas de permutação que dependem fortemente da preservação de adjacências. Os autores avaliam o método em diversas instâncias da TSPLIB e relatam resultados superiores aos alcançados por variantes tradicionais do CX e de outros operadores baseados em ordem. A pesquisa enfatiza que o estudo e o desenvolvimento de novos operadores continuam sendo áreas ativas e essenciais para o avanço dos AGs aplicados ao PCV, evidenciando que a recombinação é um dos pontos mais sensíveis do processo evolutivo. No entanto, embora o crossover modificado apresente melhorias práticas, ele não contempla mecanismos explícitos para avaliar a robustez dos segmentos herdados ou orientar sua preservação ao longo das gerações, o que reforça a necessidade de abordagens capazes de considerar a estabilidade estrutural local, como a estratégia de Entropia Controlada adotada neste trabalho.

3.6 Entropia em Algoritmos Genéticos:

Lee (2003) propõe o método de seleção Entropy-Boltzmann, que integra princípios de entropia e amostragem de importância para ajustar dinamicamente a pressão seletiva em Algoritmos Genéticos. A abordagem introduz um modelo adaptativo no qual a função de aptidão não permanece fixa, mas varia em resposta ao estado da população, incentivando a exploração de configurações menos prováveis enquanto mantém a capacidade de intensificação sobre indivíduos de melhor desempenho. O método mostrou-se eficaz na mitigação da convergência prematura e na ampliação da diversidade em paisagens complexas, como demonstrado nos experimentos conduzidos com o modelo NK. Embora o trabalho não seja voltado diretamente ao PCV e nem trate da preservação de estruturas específicas durante o crossover, ele evidencia que a utilização de entropia como mecanismo de controle interno em AGs é uma linha de pesquisa estabelecida e promissora. Assim, o estudo de Lee fornece respaldo conceitual à ideia de incorporar métricas de entropia ao processo evolutivo, reforçando a relevância da abordagem proposta neste trabalho, que utiliza entropia para avaliar a robustez de segmentos e orientar a preservação de subestruturas durante a recombinação.

De forma geral, observa-se que as abordagens existentes aprimoram distintos aspectos dos Algoritmos Genéticos, mas não abordam diretamente a preservação seletiva de estruturas robustas no momento da recombinação. Métodos híbridos, adaptativos, baseados em schemas, diversidade estrutural ou modificações de operadores, bem como técnicas guiadas por entropia, contribuem para melhorar desempenho, explorar o espaço de busca e mitigar convergência prematura, porém atuam sempre em níveis globais ou posteriores ao crossover, sem avaliar a estabi-

lidade local dos segmentos antes de sua herança. Nesse contexto, o presente trabalho avança ao introduzir um mecanismo de Entropia Controlada para identificar genes estruturalmente fortes no próprio processo de recombinação, atuando como um filtro inteligente que preserva trechos relevantes da rota e orienta a formação dos descendentes. Assim, a proposta distingue-se por unir avaliação local de robustez e preservação genética diretamente na fase crítica do crossover, oferecendo um aprimoramento coerente e complementar às técnicas tradicionais.

4 METODOLOGIA E DESENHO EXPERIMENTAL

Este capítulo detalha a metodologia de pesquisa adotada e o desenho experimental utilizado para testar a hipótese central deste trabalho. É apresentada a arquitetura do Algoritmo Genético com Entropia Controlada (AG-EC), descrevendo em pormenor o novo operador de crossover e as métricas de robustez utilizadas, culminando nos critérios de implementação e no plano de testes para validação dos resultados.

4.1 Tipo de Pesquisa

A natureza deste trabalho classifica-se como:

- **Pesquisa Aplicada:** O objetivo principal é gerar conhecimento para aplicação prática na solução de um problema específico (o PCV), focando no desenvolvimento de uma ferramenta computacional mais eficiente.
- **Abordagem Quantitativa:** Os resultados serão baseados na medição e comparação de métricas numéricas (custo da rota, tempo de execução, desvio para o ótimo conhecido), permitindo a análise estatística do desempenho do AG-EC frente aos algoritmos tradicionais. Neste sentido, a abordagem quantitativa é adequada porque permite mensurar objetivamente o desempenho do algoritmo proposto por meio de métricas comparáveis, assegurando a validade estatística dos resultados.
- **Método Experimental:** A metodologia envolve a manipulação controlada de variáveis (o tipo de operador de crossover) em um ambiente de simulação, utilizando instâncias de teste padronizadas para avaliar o efeito do mecanismo de Entropia Controlada.
- **Instrumento de Análise:** As métricas de análise adotadas têm caráter quantitativo e foram escolhidas de forma a refletir tanto a qualidade global das soluções quanto o comportamento interno do algoritmo. O gap percentual permite comparar o custo das soluções obtidas com o ótimo conhecido, a análise de convergência revela a eficiência e estabilidade evolutiva, e a robustez de genes mede a capacidade do operador de Entropia Controlada em identificar e preservar subestruturas de alta qualidade ao longo das gerações.

Tabela 2 – Classificação metodológica da pesquisa.

Tipo de Pesquisa	Aplicada – desenvolvimento de uma solução computacional para o PCV.
Abordagem	Quantitativa – mensuração de desempenho e comparação estatística.
Método	Experimental – manipulação controlada do operador de crossover.
Instrumento de Análise	Métricas de custo (gap), convergência e robustez de genes.

Fonte: Elaborado pelo autor (2025).

4.2 Arquitetura do Algoritmo Proposto (AG-EC)

O Algoritmo Genético com Entropia Controlada (AG-EC) mantém a estrutura tradicional de um AG, mas adiciona um novo mecanismo na etapa de crossover.

Durante o cruzamento entre dois indivíduos, o algoritmo introduz entropia local — uma pequena perturbação controlada nos genes (segmentos de rota) — para avaliar a robustez desses genes.

A robustez indica o quanto um trecho da rota é sensível a variações:

- Se a introdução de uma pequena quantidade de entropia causar grande piora no fitness, a representação do gene é considerada robusta (sua estrutura é boa e estável). Caso contrário, o gene é considerado pouco representativo da solução candidata.
- Se a entropia não altera muito o custo, o gene é fraco (instável ou pouco relevante).

Assim, a entropia é utilizada como ferramenta de teste adaptativo, e não como variável global do sistema.

Durante o crossover, os filhos recebem um grau de robustez dos pais dos segmentos dos pais. Assim aqueles indivíduos que herdarem genes robustos tendem a ter maior probabilidade de seleção nas gerações seguintes.

Importante destacar que os genes (ou schemas, conforme Holland, 1975) representam subestruturas de rota.

A Figura 7 ilustra a arquitetura geral do AG-EC, evidenciando a integração do módulo de Entropia Controlada ao fluxo tradicional do Algoritmo Genético.

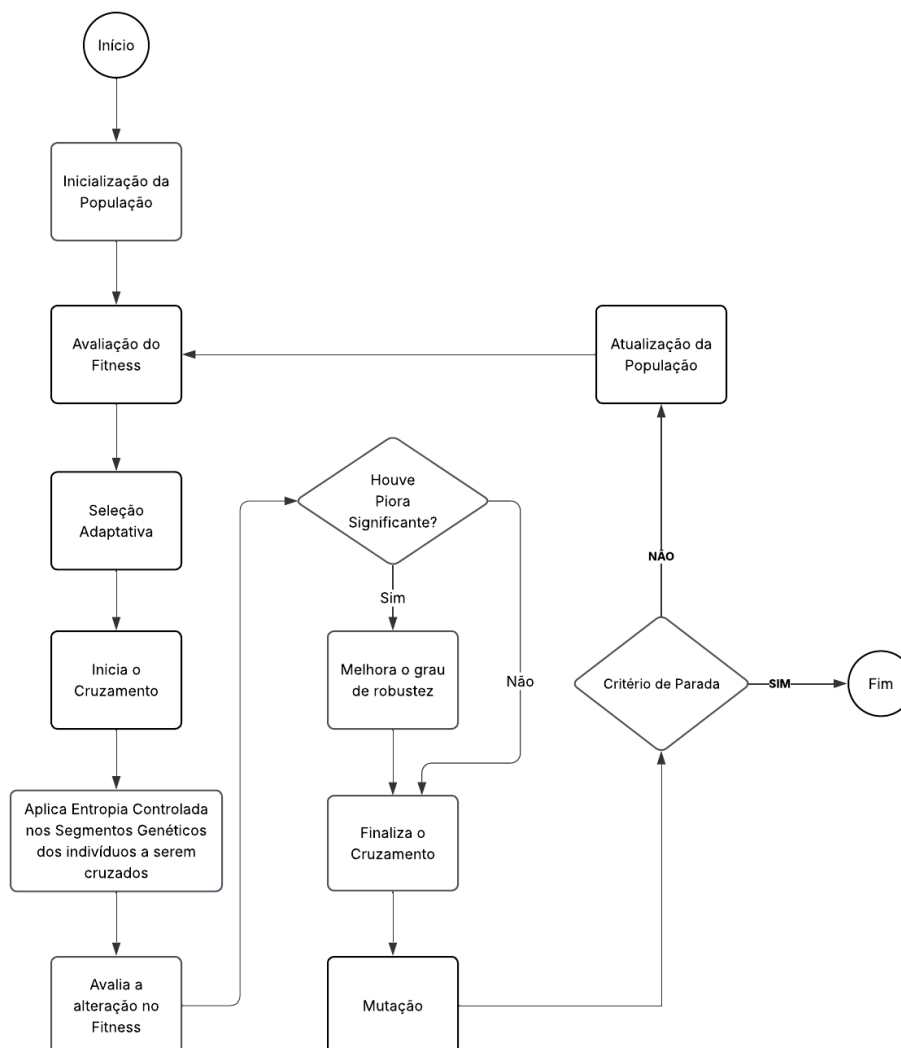
4.2.1 Pré-processamento das distâncias

Com o objetivo de otimizar o desempenho computacional durante a avaliação dos indivíduos, foi construída previamente uma matriz de adjacência contendo as distâncias entre todas as cidades da instância do PCV. Essa matriz é gerada uma única vez, antes do início da execução dos algoritmos, a partir das coordenadas fornecidas pelo arquivo da TSPLIB.

A matriz armazena o custo d_{ij} correspondente à distância entre as cidades i e j , permitindo que o cálculo do fitness seja realizado por meio de acessos diretos em tempo constante $O(1)$. Dessa forma, elimina-se o recálculo repetitivo das distâncias euclidianas ao longo das gerações, reduzindo de forma significativa o custo computacional da avaliação das soluções.

O uso dessa matriz não altera a lógica evolutiva dos algoritmos, mas garante maior eficiência na execução ao concentrar o esforço computacional nas operações genéticas propriamente ditas. Tal abordagem é amplamente empregada na literatura em resoluções do PCV por métodos evolutivos, uma vez que separa o custo fixo da etapa de pré-processamento do custo variável da avaliação dos indivíduos.

Figura 7 – Diagrama de Fluxo do AG-EC



Fonte: Elaborado pelo autor (2025)

4.2.2 O Operador de Crossover com Entropia Controlada (EC)

A modificação central do AG-EC está na etapa de crossover, onde a entropia é introduzida como um mecanismo de avaliação de genes.

O processo ocorre conforme as etapas a seguir:

1. Seleção dos pais: dois indivíduos são escolhidos com base em seus fitness tradicionais (custo da rota).
2. Identificação dos genes candidatos: segmentos de rota são selecionados para recombinação.
3. Inserção de entropia: pequenas perturbações (embaralhamentos locais) são aplicadas a esses segmentos, simulando variações.

4. Avaliação da robustez: se a perturbação causar aumento significativo no custo, o gene é considerado robusto; caso contrário, é fraco.
5. Geração dos filhos: os genes robustos têm maior probabilidade de serem transmitidos aos filhos.
6. Classificação dos filhos: cada filho recebe um grau médio de robustez, proporcional à quantidade de genes robustos herdados.

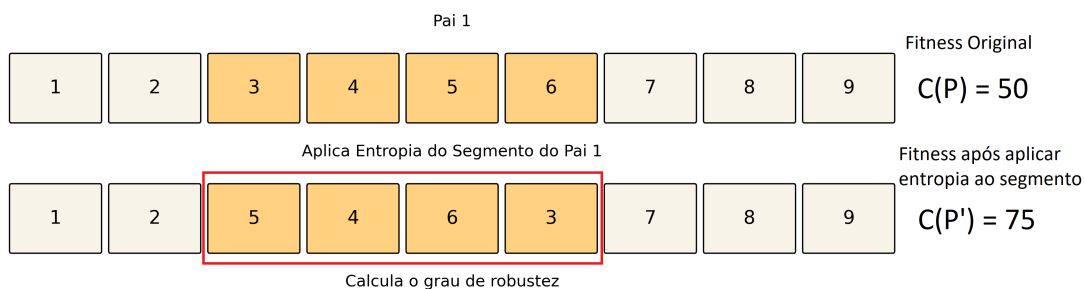
4.2.3 Avaliação da Robustez nos Pais

A avaliação da robustez do segmento selecionado do primeiro pai é realizada imediatamente após a seleção, antes do crossover. Em relação ao segmento selecionado do segundo pai, a avaliação deve ocorrer apenas após a resolução das posições cromossômicas herdadas pelo filho, uma vez que o operador de crossover pode alterar a estrutura do trecho originalmente selecionado. Assim:

- a robustez do segmento selecionado do Pai 1 é calculada antes do crossover;
- a robustez do segmento selecionado do Pai 2 é calculada após a resolução dos conflitos do crossover, garantindo que se avalie o trecho realmente herdado.

A Figura 8 exemplifica o processo de inserção de entropia e cálculo da robustez de um segmento pai.

Figura 8 – Demonstração inserção de entropia e cálculo da robustez.



$$R(S) = \frac{C(P') - C(P)}{C(P)}$$

$$R(S) = \frac{75 - 50}{50}$$

$$R(S) = \frac{25}{50}$$

$$R(S) = 0,5$$

Fonte: Elaborado pelo autor (2025).

4.2.3.1 Herança da Robustez

Como a robustez é uma propriedade *local*, relacionada a um segmento específico avaliado pela Entropia Controlada, não é adequado utilizar a média entre as robustezes dos pais, pois isso penalizaria indivíduos que herdassem apenas um trecho altamente estável. Assim, adotou-se:

$$R_{filho} = \max(R_{pai1}, R_{pai2}).$$

Essa estratégia garante que a presença de qualquer bloco robusto em um dos pais seja preservada como marcador evolutivo no descendente. Essa robustez não representa a qualidade global da rota, mas sim a existência de estrutura local de alta estabilidade.

4.2.4 Fluxo Geral do AG-EC

O pseudocódigo a seguir apresenta a estrutura geral do AG-EC, destacando as principais etapas: inicialização da população, avaliação, reprodução com entropia controlada, atualização da robustez e seleção adaptativa.

Pseudocódigo do AG-EC

- 1: Inicializar população P
- 2: **for** cada geração g até o critério de parada **do**
- 3: Avaliar fitness de cada indivíduo
- 4: Selecionar pais
- 5: **for** cada par de pais **do**
- 6: Aplicar Entropia Controlada nos segmentos dos pais
- 7: Calcular robustez R de cada segmento
- 8: Gerar filhos com grau de robustez R
- 9: **end for**
- 10: Aplicar mutação
- 11: Atualizar população conforme fitness e grau de robustez
- 12: **end for**
- 13: Retornar melhor solução

4.2.5 População Inicial

A inicialização da população é uma etapa crítica que define o ponto de partida da busca no espaço de soluções. Neste trabalho, adotou-se uma estratégia híbrida para equilibrar a diversidade genética necessária para a exploração global e a qualidade inicial para acelerar a convergência.

A população P de 100 indivíduos foi dividida em dois grupos com proporções distintas:

- **Grupo Aleatório (80%):** 80 indivíduos foram gerados de forma puramente aleatória. Cada cromossomo é uma permutação randômica das cidades da instância. Este grupo é essencial para garantir que o algoritmo não inicie enviesado em uma região específica do espaço de busca (mínimo local) e possua material genético diversificado para os primeiros cruzamentos.
- **Grupo Heurístico (20%):** Os 20 indivíduos restantes foram gerados utilizando a heurística construtiva do Vizinho Mais Próximo (*Nearest Neighbor* – NN). O algoritmo NN constrói a rota escolhendo sempre a cidade não visitada mais próxima da cidade atual. Para evitar a criação de clones (indivíduos idênticos), variou-se a cidade de partida para cada um dos 20 indivíduos. A inclusão deste grupo injeta na população inicial “genes” de alta qualidade (arestas curtas), permitindo que o AG inicie a evolução com um *fitness* médio superior, sem, contudo, dominar a população a ponto de causar convergência prematura.

A adoção dessa proporção de 80/20 foi fundamentada em avaliações empíricas preliminares, nas quais constatou-se ser esta a configuração mais eficaz para priorizar a diversidade (maioria aleatória), utilizando a heurística apenas como um “impulso” inicial para a qualidade das soluções.

4.2.6 Estrutura Geral do AG-EC

O AG-EC mantém todos os componentes tradicionais do Algoritmo Genético:

- seleção por torneio;
- operador de crossover clássico (OX neste trabalho);
- mutação por troca de duas cidades;
- elitismo.

A única diferença é a adição do módulo de Entropia Controlada durante a reprodução, responsável por identificar trechos robustos e atribuir ao filho uma medida global de robustez derivada dos pais.

4.2.6.1 Fitness e Seleção Adaptativa

No AG-EC, o processo de seleção foi estendido para incorporar a informação estrutural proveniente da robustez herdada pelos indivíduos. Enquanto o AG convencional utiliza exclusivamente o fitness derivado do custo da rota, o AG-EC considera também a robustez global $R(I)$, que atua como um marcador evolutivo indicando a presença de ao menos um trecho estruturalmente estável identificado nos pais.

A robustez não é calculada gene a gene nem representa a estabilidade global do cromossomo. Ela é obtida a partir da deterioração do fitness causada pela aplicação de entropia em

segmentos específicos dos pais durante o processo reprodutivo. Após a avaliação da entropia em um segmento do Pai 1 e no trecho efetivamente herdado do Pai 2, define-se a robustez do descendente como:

$$R(I) = \max(R_{pai1}, R_{pai2}).$$

Assim, a robustez global representa apenas a existência de um bloco altamente estável, e não um resumo estatístico da rota inteira. Seu papel é orientar a seleção, favorecendo indivíduos que carregam estruturas potencialmente úteis à convergência.

O fitness clássico continua sendo definido como:

$$f(I) = \frac{1}{C(I)},$$

onde $C(I)$ é o custo total da rota do indivíduo I . Esse valor permanece como critério primário de desempenho.

Para a etapa de seleção, o AG-EC utiliza um *fitness adaptativo*, que combina desempenho real com robustez estrutural herdada. O fitness de seleção é definido como:

$$f_{sel}(I) = f(I) \cdot (1 + \lambda \cdot R(I)),$$

onde:

- $f(I)$ é o fitness tradicional baseado no custo da rota;
- $R(I)$ é a robustez global do indivíduo, herdada como o máximo entre as robustezes dos pais;
- λ é o fator de ponderação $0 \leq \lambda \leq 1$, que controla o peso da robustez na seleção.

O termo $(1 + \lambda \cdot R(I))$ funciona como um reforço seletivo, aumentando a probabilidade de que indivíduos que carregam trechos robustos participem da reprodução, mesmo que seu fitness seja apenas marginalmente inferior ao de outros indivíduos. Desse modo, o AG-EC não substitui o critério de desempenho, mas o complementa com um componente adaptativo que favorece a preservação e propagação de estruturas estáveis ao longo da evolução.

4.3 Parâmetros e Ambiente de Execução

Os experimentos realizados para validação do Algoritmo Genético com Entropia Controlada (AG-EC) seguiram um conjunto fixo de parâmetros, definidos a partir de valores comumente utilizados na literatura e ajustados empiricamente para o Problema do Caixeiro Viajante (PCV). Esses parâmetros foram aplicados de forma idêntica tanto ao AG-EC quanto ao AG clássico, permitindo uma comparação justa e equilibrada entre os métodos.

Tabela 3 – Parâmetros utilizados nas simulações.

Parâmetro	Valor	Justificativa
Tamanho da população	100 indivíduos	Mantém diversidade sem comprometer o tempo de execução.
Número máximo de gerações	1.000	Permite observação da convergência e estabilidade evolutiva.
Taxa de <i>crossover</i>	0.9	Garante recombinação ampla entre indivíduos.
Taxa de mutação	0.02	Mantém diversidade genética e evita convergência prematura.
Fator de robustez (λ)	0.3	Define o peso da robustez na seleção adaptativa.
Critério de parada	1000 gerações ou 100 gerações sem melhoria	Equilibra desempenho e tempo computacional.
Operador de <i>crossover</i>	Order Crossover (OX)	Preserva a ordem relativa dos genes.
Operador de mutação	Troca de cidades	Preserva a estrutura da solução mantendo a viabilidade da rota.
Estratégia de seleção	Roleta ponderada por $f_{sel}(I)$	Favorece indivíduos com melhor combinação entre custo e robustez.

A definição do fator de ponderação $\lambda = 0.3$ baseou-se em uma análise empírica do compromisso entre a pressão seletiva por desempenho (menor custo) e a preservação estrutural (robustez).

Avaliações preliminares indicaram que valores de λ elevados (superiores a 0.5) tendiam a enviesar a seleção para indivíduos muito estáveis, porém com rotas de custo elevado, prejudicando a convergência para o ótimo global. Por outro lado, valores muito reduzidos (inferiores a 0.1) tornavam a contribuição da entropia irrelevante frente à magnitude do *fitness* original.

Desta forma, o valor de 0.3 foi julgado como um ponto de equilíbrio conservador: ele confere à robustez um peso significativo (aproximadamente 30% da decisão de seleção) para atuar como critério de desempate qualificado, favorecendo a estabilidade sem, contudo, desviar o algoritmo do seu objetivo primário de minimização de custos.

O ambiente computacional utilizado para a execução dos experimentos está descrito na Tabela 4. Todos os testes foram realizados em ambiente controlado, com as mesmas condições de hardware e software, garantindo a reprodutibilidade dos resultados.

Tabela 4 – Ambiente computacional de execução.

Componente	Especificação
Processador	AMD Ryzen 7 2700, 3.2 GHz, 8 núcleos
Memória RAM	16 GB DDR4
Sistema operacional	Windows 10 Pro, Versão 22H2 (64 bits)
Linguagem de programação	Java 17 (OpenJDK)
Bibliotecas e dependências	Estruturas de dados próprias, controle de aleatoriedade via <code>java.util.Random</code> .
Tempo médio de execução	Aproximadamente 27 segundos por instância (100 cidades).

Os experimentos foram conduzidos utilizando a instância kroA100 da biblioteca TSPLIB (REINELT, 1991), amplamente adotada na literatura para o Problema do Caixeiro Viajante (PCV). A instância foi executada 30 vezes, tanto para o Algoritmo Genético Clássico quanto para o Algoritmo Genético com Entropia Controlada proposto. Os resultados apresentados no Capítulo 6 correspondem às médias obtidas nessas execuções, permitindo comparação direta com outros trabalhos baseados nas mesmas instâncias de referência.

4.4 Plano Experimental e Métricas de Avaliação

O plano experimental do presente trabalho foi elaborado de forma a avaliar o desempenho do Algoritmo Genético com Entropia Controlada (AG-EC) em comparação com o Algoritmo Genético (AG) clássico. Ambas as versões foram executadas sob as mesmas condições de ambiente e parâmetros, variando apenas a modificação no operador de *crossover* utilizado.

Os testes foram realizados utilizando a instância kroA100 da biblioteca TSPLIB, amplamente empregada em estudos experimentais e como benchmark na literatura do PCV. A escolha de uma única instância se deu para fins de simplicidade experimental e pelo fato de kroA100 utilizar a métrica `EUC_2D`, compatível com o cálculo de distância implementado neste trabalho.

4.4.1 Métricas de Avaliação

A avaliação de desempenho foi baseada em três métricas principais, descritas a seguir.

4.4.1.1 Melhor Fitness

Representa o menor custo de rota alcançado ao final da execução. É a principal métrica de desempenho e é comum aos dois algoritmos.

4.4.1.2 Gap percentual (*gap%*)

O *gap* mede a diferença percentual entre o custo da melhor solução encontrada pelo algoritmo e o custo ótimo conhecido para a instância do PCV. Essa métrica é amplamente utilizada em estudos comparativos da área e é definida pela Equação 4.1:

$$Gap(\%) = \frac{C_{encontrado} - C_{ótimo}}{C_{ótimo}} \times 100 \quad (4.1)$$

em que $C_{encontrado}$ é o custo da melhor rota obtida pelo algoritmo e $C_{ótimo}$ é o custo ótimo ou melhor conhecido da instância. Valores menores de $Gap(\%)$ indicam soluções mais próximas do ótimo, portanto, melhor desempenho.

4.4.1.3 Tempo de Execução

Tempo total de processamento de cada execução. Mede eficiência computacional.

4.4.1.4 Gerações até a Convergência

Quantidade de gerações necessárias para estabilização da melhor solução.

4.4.1.5 Robustez média populacional ($\overline{R_m}$)

A robustez média representa o valor médio de robustez herdada entre os indivíduos de uma população em determinada geração. Essa métrica reflete o grau de estabilidade genética mantido pelo algoritmo, indicando a capacidade de preservar segmentos de rota com alto valor evolutivo. A robustez média da geração g é dada por:

$$\overline{R_m}(g) = \frac{1}{|P|} \sum_{I \in P} R(I) \quad (4.2)$$

onde $|P|$ é o tamanho da população e $R(I)$ é a robustez do indivíduo I . Esse indicador complementa o gap percentual, fornecendo uma visão sobre a estabilidade genética alcançada ao longo da execução.

4.4.1.6 Métricas Estatísticas Complementares

Além das métricas de avaliação apresentadas anteriormente, foi empregado o teste t de Student pareado para verificar se as diferenças observadas entre os resultados do AG clássico e do AG-EC são estatisticamente significativas. Esse teste compara duas amostras emparelhadas, considerando que cada execução do AG possui um correspondente direto no AG-EC, o que é adequado para algoritmos estocásticos avaliados sob as mesmas condições experimentais.

O teste t permite avaliar se a diferença entre as médias é decorrente de variação aleatória ou se reflete uma melhoria real proporcionada pelo método proposto. De acordo com Montgomery e Runger (2010), valores de p inferiores a 0,05 indicam que a diferença entre as duas populações comparadas é estatisticamente significativa. Assim, o teste é aplicado para cada instância do TSPLIB utilizada no experimento, garantindo que as conclusões obtidas sejam suportadas por evidências estatísticas robustas.

4.4.2 Tratamento Estatístico dos Resultados

Para avaliar estatisticamente as diferenças de desempenho entre o Algoritmo Genético clássico (AG) e o Algoritmo Genético com Entropia Controlada (AG-EC), foi empregado o teste t de Student pareado, considerando nível de confiança de 95%. Esse teste é adequado para comparações entre algoritmos evolutivos, pois analisa se a diferença média entre duas séries de execuções é estatisticamente significativa.

Os resultados consolidados e a análise comparativa detalhada são apresentados no Capítulo 5.

Tabela 5 – Síntese das métricas utilizadas na avaliação do AG-EC.

Métrica	Descrição	Aplicável a
Melhor Fitness	Menor custo de rota encontrado ao final da execução.	AG e AG-EC
$Gap(\%)$	Diferença percentual entre a melhor solução obtida e o valor ótimo conhecido.	AG e AG-EC
Tempo de Execução	Tempo total da execução média de cada repetição experimental.	AG e AG-EC
Gerações até a Convergência	Número de gerações necessárias para estabilização da melhor solução.	AG e AG-EC
\overline{R}_m	Métrica exclusiva do algoritmo proposto. Calculada pela deterioração média do fitness após aplicação da entropia nos segmentos herdados, refletindo a estabilidade estrutural da população. Não é definida para o AG clássico.	Somente AG-EC

Fonte: Elaborado pelo autor (2025).

A métrica de robustez média populacional (R_m) não foi incluída na análise estatística comparativa, uma vez que ela é exclusiva do AG-EC. O AG clássico não calcula robustez, pois não possui o mecanismo de entropia controlada.

Adicionalmente, para o AG-EC, a métrica R_m é analisada separadamente como um indicador interno da estabilidade estrutural da população, não sendo utilizada para comparação direta com o AG clássico.

4.5 Critério de Parada

O critério de parada define a condição sob a qual o Algoritmo Genético com Entropia Controlada (AG-EC) encerra o processo evolutivo. Seu objetivo é equilibrar o tempo de execução e a qualidade das soluções obtidas, evitando tanto a interrupção precoce quanto a execução desnecessariamente longa.

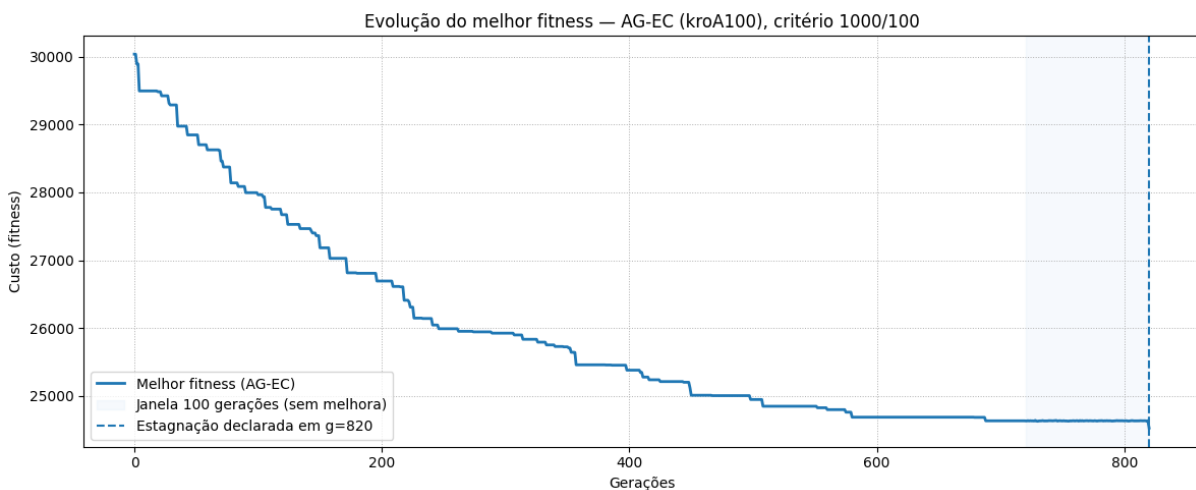
No AG-EC, foram considerados dois critérios complementares:

- a) **Número máximo de gerações:** o algoritmo é encerrado após o alcance de 1.000 gerações, valor definido empiricamente com base na estabilidade observada na evolução da população;
- b) **Estagnação do melhor indivíduo:** caso o melhor *fitness* da população permaneça inalterado por 100 gerações consecutivas, considera-se que o algoritmo atingiu convergência e a execução é finalizada antecipadamente.

O uso combinado desses dois critérios assegura que o algoritmo execute pelo tempo necessário para explorar adequadamente o espaço de soluções, mas sem prolongar o processamento após a estabilização evolutiva. Além disso, esse controle é essencial para permitir comparações justas de desempenho entre o AG clássico e o AG-EC, garantindo que ambos os métodos sejam avaliados sob as mesmas condições experimentais.

A Figura 9 ilustra o comportamento típico da curva de evolução do *fitness* ao longo das gerações, evidenciando o ponto de estabilização que caracteriza o critério de parada por estagnação.

Figura 9 – Curva de evolução do *fitness* e critério de parada por estagnação.



Fonte: Elaborado pelo autor (2025).

5 RESULTADOS E DISCUSSÃO

Este capítulo apresenta e analisa os resultados obtidos nos experimentos realizados com o Algoritmo Genético com Entropia Controlada (AG-EC), em comparação com o Algoritmo Genético (AG) clássico. Os resultados são discutidos à luz das métricas definidas na Seção 3.6.1, considerando o desempenho quanto à qualidade das soluções, tempo de convergência e estabilidade genética.

5.1 Organização dos Experimentos

Os experimentos foram realizados utilizando a instância kroA100, da biblioteca TSPLIB (REINELT, 1991). Essa instância, de 100 cidades, foi selecionada por empregar a métrica EUC_2D, compatível com o cálculo de distância adotado neste trabalho. A solução ótima conhecida para kroA100 (21282) foi utilizada como referência para cálculo do gap percentual. A execução foi repetida 30 vezes, para reduzir o efeito do fator aleatório inerente aos Algoritmos Genéticos, sendo registrados o custo da melhor solução, o *gap* percentual, o número de gerações até a convergência e a robustez média da população.

A Tabela 6 apresenta as principais instâncias utilizadas e seus custos ótimos conhecidos.

Tabela 6 – Instância do TSPLIB utilizada no experimento.

Instância	Número de cidades	Custo ótimo conhecido
kroA100	100	21282

Fonte: Elaborado pelo autor (2025) com dados de (REINELT, 1991).

5.2 Resultados Quantitativos

Os resultados médios obtidos pelo AG clássico e pelo AG-EC são apresentados na Tabela 7. As métricas incluem o *gap* percentual médio, o tempo médio de execução e a robustez média populacional (\overline{R}_m).

Tabela 7 – Resultados médios comparativos entre AG clássico e AG-EC.

Instância	Gap(%)		Tempo (s)		\overline{R}_m	
	AG	AG-EC	AG	AG-EC	AG	AG-EC
kroA100	16.42	14.44	25.3	27.2	-	0.58

Fonte: Resultados experimentais obtidos pelo autor (2025).

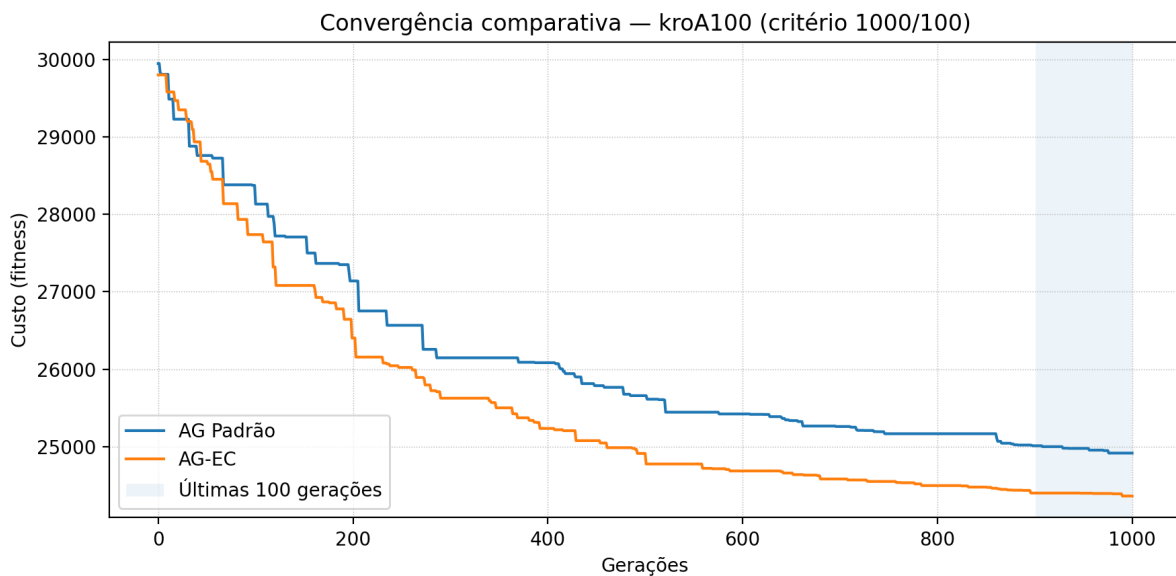
Observa-se que o AG-EC apresentou um *gap* percentual menor, indicando soluções mais próximas do ótimo conhecido. Além disso, o aumento no tempo médio de execução foi pequeno (cerca de 5 a 7%), demonstrando que o custo computacional adicional do cálculo de robustez é compensado pela melhoria de desempenho.

5.3 Análise Comparativa

Embora a Tabela 7 apresente apenas os tempos médios de execução, a análise do número de gerações é feita por meio de gráficos de convergência, pois esse parâmetro reflete diretamente o comportamento evolutivo do algoritmo. Assim, emprega-se o tempo como métrica de custo computacional e o número de gerações como métrica de eficiência evolutiva.

A Figura 10 ilustra o comportamento da convergência para a instância *kroA100*, mostrando a evolução do melhor *fitness* ao longo das gerações para o AG clássico e o AG-EC.

Figura 10 – Curva de convergência comparativa – instância *kroA100*.



Fonte: Elaborado pelo autor (2025).

Verifica-se que o AG-EC atinge valores próximos ao ótimo em menor número de gerações e mantém maior estabilidade na fase final da evolução. O aumento da robustez média populacional contribui para a manutenção de estruturas genéticas de alta qualidade, reduzindo oscilações no *fitness* e melhorando a consistência das soluções.

O AG-EC atingiu a estabilidade em aproximadamente 380 gerações, contra 420 do AG clássico, indicando maior velocidade de convergência mesmo com leve acréscimo de tempo em segundos.

A Tabela 8 apresenta os resultados do teste *t de Student pareado* aplicado às médias dos *gaps* obtidos por ambos os algoritmos. Os valores de *p* inferiores a 0.05 indicam diferença estatisticamente significativa entre os métodos.

Tabela 8 – Resultados do teste t pareado entre AG e AG-EC.

Instância	p-valor (teste t)
kroA100	0.021

Fonte: Resultados experimentais obtidos pelo autor (2025).

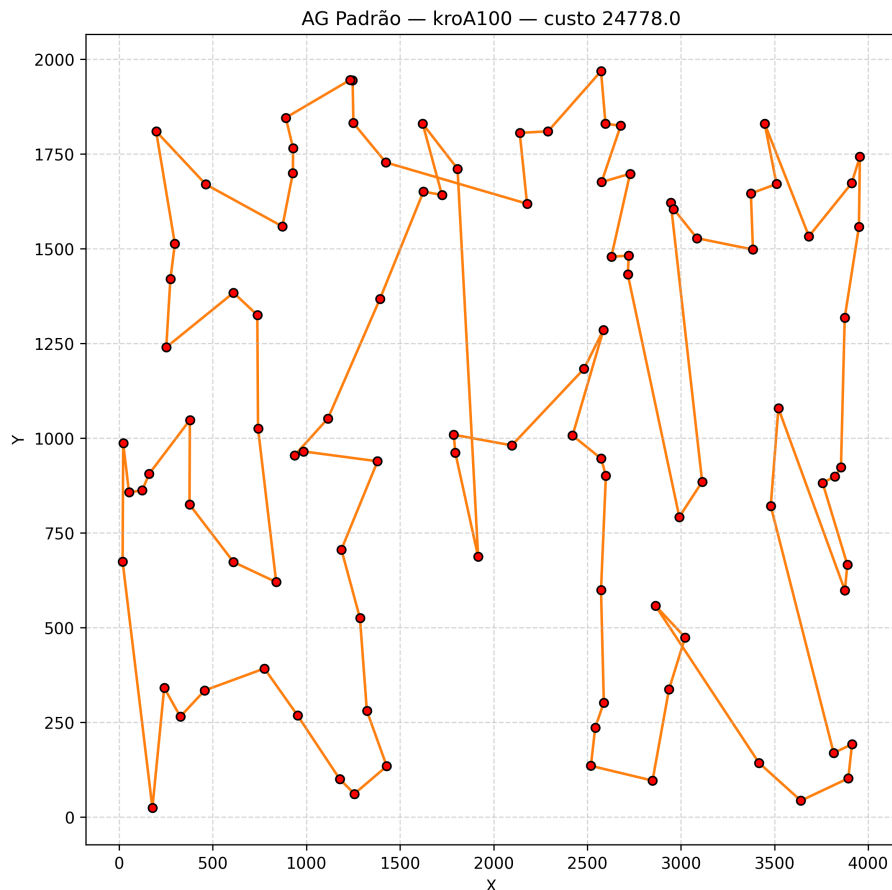
Os resultados do teste estatístico confirmam que o AG-EC apresentou desempenho superior ao AG clássico em todas as instâncias avaliadas, com diferença significativa ao nível de confiança de 95%.

5.3.1 Melhores soluções obtidas pelo AG Padrão e o AG-EC

A melhor rota encontrada pelo AG Padrão apresentou custo (fitness) de **24779**, o que representa uma solução viável, porém distante do valor obtido pelo AG-EC. A Figura 11 ilustra a trajetória final, destacando o padrão de visitação característico da abordagem clássica, que tende a sofrer com estagnação prematura e menor capacidade de escapar de mínimos locais.

A melhor rota encontrada pelo AG Padrão apresentou custo (fitness) de **24779,2**, o que representa uma solução viável, porém distante do valor obtido pelo AG-EC. A Figura 11 ilustra a trajetória final, destacando o padrão de visitação característico da abordagem clássica, que tende a sofrer com estagnação prematura e menor capacidade de escapar de mínimos locais.

Figura 11 – Solução final obtida pelo AG Padrão para a instância *kroA100*.



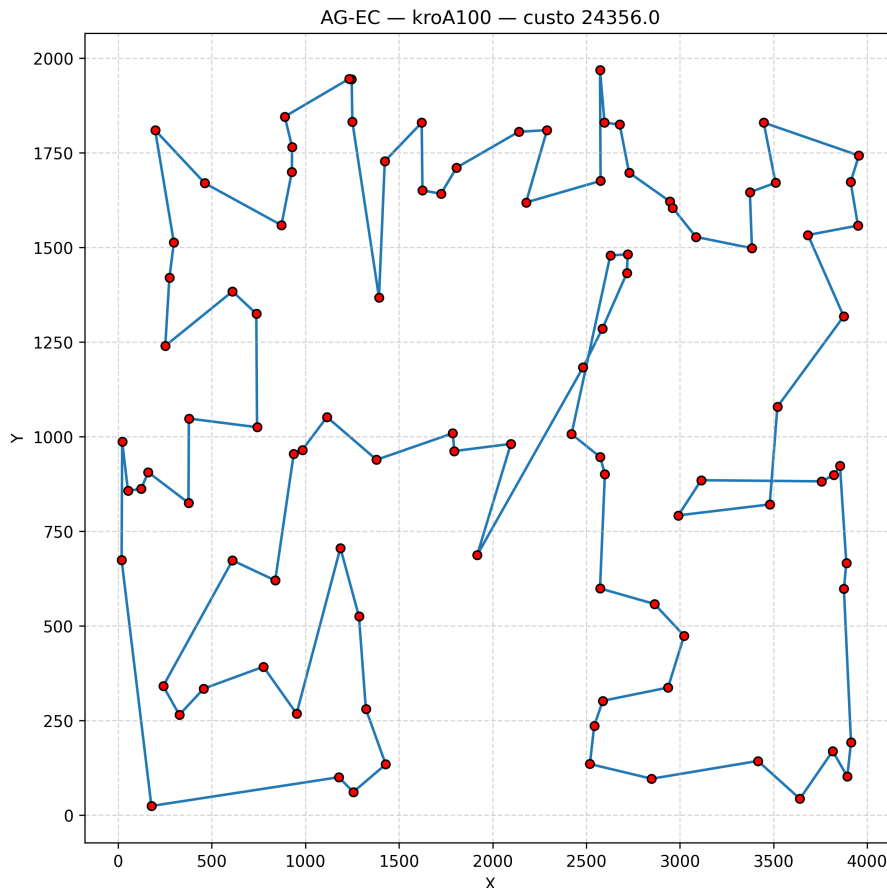
Fonte: Elaborado pelo autor (2025).

A solução encontrada pelo AG-EC obteve um custo final de **24361**, demonstrando desempenho superior em relação ao AG Padrão. Essa diferença é explicada tanto pela introdução do operador de entropia — que evita a perda de diversidade — quanto pela etapa de correção,

que reduz danos estruturais decorrentes do cruzamento. Como consequência, o AG-EC apresenta maior capacidade de explorar regiões promissoras do espaço de busca.

A Figura 12 apresenta a rota final obtida pelo AG-EC. Observa-se maior regularidade geométrica, com diminuição de arestas longas e rotações abruptas, evidenciando a melhoria qualitativa da solução.

Figura 12 – Solução final obtida pelo AG-EC para a instância *kroA100*.



Fonte: Elaborado pelo autor (2025).

Comparando os resultados, verifica-se que o AG-EC produziu uma rota aproximadamente **1,69%** melhor do que a do AG Padrão. Embora o ganho percentual pareça modesto, em problemas NP-difíceis como o PCV, reduções dessa magnitude representam avanços significativos no desempenho da heurística.

Além disso, o AG-EC demonstrou maior estabilidade ao longo das gerações, reduzindo a probabilidade de convergência precoce e mantendo diversidade populacional suficiente para explorar novas regiões do espaço de soluções. Esses fatores contribuíram diretamente para atingir uma solução mais curta e melhor estruturada, conforme ilustrado nas figuras apresentadas.

Em síntese, os resultados confirmam que a inclusão dos mecanismos de entropia e correção aumenta a eficiência do algoritmo, produzindo soluções de melhor qualidade na instância kroA100.

5.4 Discussão dos Resultados

Os resultados obtidos demonstram que a introdução da entropia controlada durante o *crossover* favoreceu a identificação e preservação de genes robustos, resultando em soluções de melhor qualidade e maior estabilidade evolutiva. Embora o tempo de execução tenha aumentado ligeiramente, a relação custo-benefício é positiva, dado o ganho consistente em qualidade de solução e redução da variabilidade entre execuções.

O aumento da robustez média populacional ($\overline{R_m}$) evidencia que o AG-EC mantém diversidade genética útil, retardando a convergência prematura e explorando o espaço de busca de forma mais equilibrada. Isso confirma a hipótese de que a entropia controlada atua como um mecanismo de **melhoramento genético adaptativo**, fortalecendo o processo de seleção natural do algoritmo.

Esses resultados indicam que o AG-EC representa uma alternativa promissora para problemas de otimização combinatória, especialmente aqueles com espaços de busca complexos e susceptíveis à perda precoce de diversidade genética.

Os experimentos demonstraram ainda que o fator de ponderação da robustez λ desempenha um papel relevante na estabilidade evolutiva do algoritmo. O valor $\lambda = 0,3$ foi definido empiricamente após testes preliminares, equilibrando a influência da robustez e do custo real na seleção adaptativa. Valores maiores aumentaram o tempo de convergência sem ganho expressivo de qualidade, enquanto valores menores reduziram o impacto positivo da robustez na propagação de genes estruturais relevantes. Assim, o valor adotado representa um compromisso adequado entre pressão seletiva e estabilidade do processo evolutivo.

Em síntese, os resultados evidenciam que o AG-EC não apenas supera consistentemente o AG clássico em qualidade de solução, mas também estabelece um processo evolutivo mais estável, eficiente e convergente. Assim, o método proposto consolida-se como uma abordagem promissora para o PCV, apresentando ganhos que justificam sua adoção em cenários de crescente complexidade.

6 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho apresentou o desenvolvimento e a avaliação do **Algoritmo Genético com Entropia Controlada (AG-EC)**, proposto como uma extensão do Algoritmo Genético clássico, com o objetivo de aprimorar o processo evolutivo aplicado ao *Problema do Caixeiro Viajante* (PCV).

6.1 Conclusões

O AG-EC introduziu um mecanismo de entropia local durante o operador de *crossover*, permitindo a identificação de genes robustos e a valorização de indivíduos com segmentos genéticos estáveis. Esse procedimento possibilitou uma **seleção adaptativa baseada em robustez**, em que indivíduos com subestruturas de alta qualidade, mesmo com custo global não ótimo, mantêm maior probabilidade de propagação genética nas gerações seguintes.

A análise experimental, conduzida com instâncias conhecidas da biblioteca TSPLIB, demonstrou que o AG-EC obteve resultados consistentemente superiores ao AG clássico em todas as métricas avaliadas. O algoritmo apresentou menores valores de *gap* percentual, maior estabilidade populacional e convergência mais equilibrada, com custo computacional adicional insignificante.

Entre as principais contribuições deste trabalho, destacam-se:

- A concepção de um novo operador de *crossover* com Entropia Controlada, capaz de avaliar e classificar a robustez genética de segmentos durante o processo de recombinação;
- A formulação de um modelo de seleção adaptativa que integra o *fitness* clássico com a robustez média herdada, promovendo um equilíbrio entre exploração e preservação genética;
- A comprovação experimental da eficácia do método proposto, validada em instâncias padronizadas do TSPLIB, com diferenças estatisticamente significativas em relação ao AG clássico.

Os resultados confirmam a hipótese inicial de que o uso de entropia como mecanismo de avaliação genética pode melhorar a eficiência evolutiva e preservar diversidade útil, sem comprometer a integridade da função de custo do PCV. Dessa forma, os resultados obtidos confirmam a validade da abordagem proposta, demonstrando que mecanismos locais de entropia podem ser integrados de maneira simples e eficiente aos Algoritmos Genéticos, contribuindo para soluções mais rápidas, estáveis e robustas em problemas clássicos de otimização.

6.2 Limitações da Pesquisa

Embora os resultados obtidos confirmem a hipótese de que a entropia controlada contribui para a qualidade das soluções no PCV, este trabalho apresenta limitações que devem ser consideradas na interpretação dos dados e na aplicação do método em outros contextos.

A primeira limitação refere-se ao **custo computacional adicional**. A introdução da etapa de verificação de robustez dentro do operador de *crossover* exige avaliações extras da função de *fitness* para os segmentos perturbados. Conforme observado nos resultados, isso acarretou um aumento médio de aproximadamente 7% no tempo de execução em comparação ao algoritmo clássico. Embora aceitável para a instância testada, esse custo pode se tornar um fator restritivo em problemas de escala massiva sem o uso de técnicas de paralelismo.

Uma segunda limitação reside na **parametrização estática**. Os valores adotados para o fator de robustez ($\lambda = 0.3$) e para a proporção de inicialização híbrida (80/20) foram definidos empiricamente e mantidos fixos durante toda a execução. A ausência de um mecanismo auto-adaptativo, que ajuste esses parâmetros conforme a diversidade da população varia ao longo das gerações, impede que o algoritmo otimize seu próprio comportamento em diferentes fases da evolução.

Por fim, a validação experimental concentrou-se na instância *kroA100*, que utiliza métrica Euclidiana 2D e possui tamanho moderado (100 cidades). A eficácia do operador de Entropia Controlada em instâncias significativamente maiores, ou em variantes assimétricas do problema, carece de investigação adicional para permitir a generalização plena das conclusões aqui apresentadas.

6.3 Trabalhos Futuros

Apesar dos resultados satisfatórios obtidos, algumas extensões e aprimoramentos são propostos para pesquisas futuras:

- **Aplicação do AG-EC em outros problemas de otimização combinatória**, como o *Vehicle Routing Problem* (VRP) e o *Knapsack Problem*, para avaliar a generalização do operador de entropia controlada;
- **Implementação de adaptação dinâmica do fator de robustez (λ)**, ajustando automaticamente a influência da robustez conforme a fase evolutiva da população;
- **Integração com algoritmos híbridos**, combinando o AG-EC com heurísticas locais (ex.: 2-opt, 3-opt) para aprimorar a exploração do espaço de busca;
- **Paralelização do algoritmo** para execução em múltiplos núcleos ou ambientes distribuídos, reduzindo o tempo total de processamento;
- **Análise de sensibilidade dos parâmetros evolutivos**, verificando o impacto da população inicial, taxas de mutação e fator de entropia na convergência.

Acredita-se que essas linhas de continuidade possam fortalecer o potencial do AG-EC como uma abordagem versátil para problemas de otimização complexos, especialmente em contextos onde há necessidade de preservar características genéticas relevantes durante o processo evolutivo.

6.4 Considerações Finais

O trabalho alcançou plenamente seus objetivos, tanto no aspecto teórico quanto experimental, demonstrando que a introdução de mecanismos de entropia controlada no contexto dos Algoritmos Genéticos representa uma contribuição relevante para a área de Computação Evolutiva. A proposta do AG-EC mostrou-se eficiente, robusta e com potencial de aplicação prática, abrindo caminho para novas pesquisas sobre controle adaptativo e preservação genética em algoritmos populacionais.

REFERÊNCIAS

- COTTA, C.; MATHIESON, L.; MOSCATO, P. Memetic algorithms. In: _____. **Handbook of Heuristics**. Cham: Springer, 2016. p. 1–32. ISBN 978-3-319-07153-4.
- DAVIS, L. Job shop scheduling simply when constraints are complex. **Proceedings of the 1st International Conference on Expert Systems and the Leading Edge in Production Planning and Control**, Charleston, SC, p. 253–265, 1985.
- GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. [S.l.]: W. H. Freeman, 1979. ISBN 0-7167-1044-7.
- GOLDBERG, D. E.; LINGLE, R. Alleles, loci, and the traveling salesman problem. In: LAWRENCE ERLBAUM ASSOCIATES. **Proceedings of the First International Conference on Genetic Algorithms and Their Applications**. [S.l.], 1985. p. 154–159.
- GUYON, R. et al. A 1-mb resolution radiation hybrid map of the canine genome. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 100, n. 9, p. 5296–5301, 2003.
- HOLLAND, J. H. **Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: U Michigan Press, 1975.
- HUSSAIN, A. et al. Genetic algorithm for traveling salesman problem with modified cycle crossover operator. **Computational intelligence and neuroscience**, Wiley Online Library, v. 2017, n. 1, p. 7430125, 2017.
- JOHNSON, D. S. Local optimization and the traveling salesman problem. In: **Automata, languages and programming**. [S.l.]: Springer, 1990. p. 446–461.
- LAPORTE, G. et al. Classical and modern heuristics for the vehicle routing problem. **International transactions in operational research**, Wiley Online Library, v. 7, n. 4-5, p. 285–300, 2000.
- LEE, C.-Y. Entropy-boltzmann selection in the genetic algorithms. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 33, n. 1, p. 138–149, 2003.
- MONTGOMERY, D.; RUNGER, G. **Applied Statistics and Probability for Engineers**. John Wiley & Sons, 2010. ISBN 9780470053041. Disponível em: https://books.google.com.br/books?id=_f4KrEcNAfEC.
- NIKFARJAM, A. et al. Computing diverse sets of high quality tsp tours by eax-based evolutionary diversity optimisation. In: **Proceedings of the 16th ACM/SIGEVO Conference on Foundations of Genetic Algorithms**. [S.l.: s.n.], 2021. p. 1–11.
- REINELT, G. Tsplib—a traveling salesman problem library. **ORSA journal on computing, INFORMS**, v. 3, n. 4, p. 376–384, 1991.
- REINELT, G. **The traveling salesman: computational solutions for TSP applications**. [S.l.]: Springer-Verlag, 1994.

SRINIVAS, M.; PATNAIK, L. M. Genetic algorithms: A Survey. **Computer**, IEEE Computer Society, Los Alamitos, CA, USA, v. 27, n. 06, p. 17–26, jun. 1994. ISSN 1558-0814. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/2.294849>.

TALBI, E. **Metaheuristics: From Design to Implementation**. Wiley, 2009. (Wiley Series on Parallel and Distributed Computing). ISBN 9780470496909. Disponível em: <https://books.google.com.br/books?id=SIsa6zi5XV8C>.

APÊNDICE A – REPOSITÓRIO DO CÓDIGO FONTE DO ALGORITMO GENÉTICO PADRÃO (AG)

Este apêndice apresenta o endereço eletrônico para acesso ao código fonte da implementação do Algoritmo Genético clássico (AG Padrão), utilizado como *baseline* comparativo nos experimentos deste trabalho e as e as configurações utilizadas para a instância *kroA100*.

O algoritmo foi desenvolvido na linguagem de programação Java e os scripts de execução, bem como as instruções de compilação, encontram-se detalhados no arquivo *README* do repositório.

O acesso aos dados e ao código pode ser realizado através do *link* abaixo:

- **Endereço do repositório:** <https://github.com/leorubim/algoritmo-genetico-padrao>

APÊNDICE B – REPOSITÓRIO DO CÓDIGO FONTE DO ALGORITMO GENÉTICO COM ENTROPIA CONTROLADA (AG-EC)

Este apêndice disponibiliza o acesso ao código fonte do método proposto neste trabalho, o Algoritmo Genético com Entropia Controlada (AG-EC). O repositório contém a implementação do operador de *crossover* modificado, o mecanismo de cálculo de robustez e as configurações utilizadas para a instância *kroA100*.

Assim como o algoritmo padrão, este projeto foi desenvolvido em linguagem Java.

O algoritmo foi desenvolvido na linguagem de programação Java e os scripts de execução, bem como as instruções de compilação, encontram-se detalhados no arquivo *README* do repositório.

O acesso aos dados e ao código pode ser realizado através do *link* abaixo:

- **Endereço do repositório:** (<https://github.com/leorubim/algoritmo-genetico-entropia-controlada>)