

Relatório Científico: Otimização do Problema do Caixeiro Viajante com Algoritmos Genéticos

D. Cavalcanti¹

¹Universidade Tuiuti do Paraná
Curitiba – PR

diego.cavalcanti@utp.br

Resumo. Este trabalho apresenta a aplicação de Algoritmos Genéticos (AGs) para resolver o Problema do Caixeiro Viajante (TSP), com o objetivo de analisar o impacto de diferentes configurações dos operadores genéticos no desempenho do algoritmo. Foram testadas combinações de operadores de crossover (um ponto, dois pontos e uniforme), taxas de mutação (1%, 5% e 10%), métodos de inicialização da população (aleatório e heurístico) e critérios de parada (número fixo de gerações e convergência). Os resultados mostraram que todos os tipos de crossover obtiveram soluções de qualidade semelhantes, mas o crossover de dois pontos, combinado com inicialização heurística, taxa de mutação de 5% e critério de parada por convergência, proporcionou melhor equilíbrio entre desempenho e tempo de execução. Conclui-se que a escolha adequada dos operadores pode melhorar significativamente a eficiência do AG na resolução de problemas combinatórios como o TSP.

1. Introdução

O Problema do Caixeiro Viajante (Traveling Salesman Problem - TSP) é um dos problemas mais estudados da teoria da otimização combinatória. O desafio consiste em encontrar o menor caminho possível que permita a um caixeiro visitar uma lista de cidades exatamente uma vez e retornar à cidade de origem. É um problema NP-difícil, o que implica que não existe algoritmo conhecido capaz de resolvê-lo em tempo polinomial para qualquer instância. Algoritmos Genéticos (AGs) são técnicas metaheurísticas baseadas nos mecanismos da seleção natural e da genética biológica. Eles se mostram promissores para resolver problemas de otimização complexos como o TSP, pois podem explorar eficientemente grandes espaços de solução e evitar mínimos locais por meio de operações como mutação e crossover.

2. Descrição do Algoritmo Genético

O Algoritmo Genético (AG) implementado neste trabalho segue uma estrutura clássica baseada em populações de soluções representadas como permutações das cidades do Problema do Caixeiro Viajante (TSP). A seguir, descreve-se cada etapa do algoritmo:

1. Inicialização da população:

- *Aleatória*: cada indivíduo é uma permutação gerada de forma randômica;
- *Heurística (Nearest Neighbor)*: a rota começa por uma cidade aleatória e segue sempre para a cidade mais próxima ainda não visitada.

2. **Avaliação de fitness:** a aptidão de cada indivíduo é definida pela soma das distâncias entre cidades consecutivas, incluindo o retorno à cidade inicial. Soluções com menor distância total são consideradas melhores.
3. **Seleção:** é utilizado o método de torneio com três candidatos. A cada iteração, três indivíduos são sorteados e o de melhor fitness é selecionado como pai.
4. **Crossover:** dois pais geram dois filhos com base em operadores que garantem a validade das permutações. Foram testados:
 - Um ponto: metade do primeiro pai combinada com os genes restantes do segundo;
 - Dois pontos: um segmento intermediário fixo herdado de um dos pais, completado com os genes do outro;
 - Uniforme: cada gene é herdado aleatoriamente de um dos pais, com consistência na ordem.
5. **Mutação:** com uma taxa configurável (1%, 5% ou 10%), dois genes da permutação são trocados para manter diversidade na população.
6. **Critério de parada:**
 - *Número fixo de gerações:* o algoritmo é executado por uma quantidade pré-definida (ex: 1000 gerações);
 - *Convergência:* o algoritmo encerra quando não há melhoria no melhor indivíduo após 50 gerações consecutivas.

Essa estrutura permite que o AG explore eficientemente o espaço de busca, equilibrando exploração e intensificação, e é particularmente adequada para problemas como o TSP devido à sua natureza combinatória e ao grande número de soluções possíveis.

3. Metodologia

O algoritmo genético foi implementado na linguagem Python, utilizando as bibliotecas *numpy*, *pandas* e *csv* para manipulação de dados e análise de resultados. O processo de execução do AG seguiu as seguintes etapas:

- **Inicialização da população:** duas abordagens foram testadas — uma aleatória, gerando permutações puramente randômicas das cidades, e outra heurística, baseada na vizinhança mais próxima (Nearest Neighbor);
- **Avaliação de fitness:** a aptidão de cada indivíduo foi calculada com base na distância total percorrida no tour;
- **Seleção:** os indivíduos mais aptos foram escolhidos por meio do método de torneio, que avalia subconjuntos aleatórios da população;
- **Crossover:** foram testados três métodos de recombinação: crossover de um ponto, crossover de dois pontos e crossover uniforme;
- **Mutação:** realizada por meio da troca de duas cidades dentro de um indivíduo, com taxas de mutação ajustadas em 1%, 5% e 10%;
- **Critério de parada:** o algoritmo foi executado com dois critérios distintos: um número fixo de gerações (1000) e uma condição de convergência baseada na ausência de melhorias por 50 gerações consecutivas.

O experimento foi automatizado para testar todas as combinações entre os parâmetros mencionados, utilizando instâncias do TSP com tamanhos variados. Os resultados foram registrados no arquivo `results.csv` para posterior análise quantitativa.

4. Análise dos Resultados Experimentais

Os resultados obtidos a partir das execuções do algoritmo genético foram analisados com base em duas métricas principais: o tempo de execução e a qualidade da solução (comprimento total do tour). As análises foram realizadas a partir dos dados registrados no arquivo `results.csv`, que contém os resultados para todas as combinações de operadores genéticos e configurações testadas.

A seguir, destacam-se as principais observações:

- **Qualidade das Soluções:** As variações entre os operadores de crossover (um ponto, dois pontos, uniforme) apresentaram pouca diferença na qualidade final das soluções. A distância média encontrada se manteve praticamente constante para cada instância, indicando que todos os métodos foram eficazes em encontrar boas soluções.
- **Desempenho Temporal:** O tempo de execução foi impactado significativamente pelos critérios de parada e pelo método de inicialização. A inicialização heurística, por gerar soluções melhores desde o início, permitiu uma convergência mais rápida. Já o critério de parada por convergência, quando comparado a um número fixo de gerações, resultou em menor tempo médio sem prejuízo da qualidade.
- **Taxas de Mutação:** As taxas mais altas (10%) aumentaram a variabilidade, mas não garantiram melhor desempenho. Em alguns casos, resultaram em maior tempo de execução devido ao maior número de alterações nos indivíduos e instabilidade na convergência.
- **Configuração Ideal:** A combinação que apresentou melhor equilíbrio entre tempo e qualidade foi: inicialização heurística, crossover de dois pontos, taxa de mutação de 5% e critério de parada por convergência.

Esses resultados demonstram que, embora os operadores genéticos apresentem desempenhos similares em qualidade de solução, suas combinações influenciam significativamente a eficiência do algoritmo. A escolha adequada dos parâmetros é, portanto, essencial para alcançar bons resultados com tempo de execução reduzido.

5. Precisão da Solução

Para avaliar a qualidade das soluções encontradas pelo algoritmo genético, foi calculado o desvio percentual em relação ao valor ótimo conhecido ou estimado para cada instância do TSP. A Tabela 1 apresenta o desvio médio percentual obtido com as diferentes configurações de operadores genéticos.

Tabela 1. Precisão média da solução: desvio percentual em relação ao ótimo

Instância	Crossover	Inicialização	Mutação	Desvio (%)
tsp_1.csv	Dois pontos	Heurística	5%	0.00%
tsp_2.csv	Um ponto	Aleatória	1%	1.25%
tsp_3.csv	Uniforme	Heurística	10%	0.67%
tsp_4.csv	Dois pontos	Heurística	5%	0.00%
tsp_5.csv	Uniforme	Aleatória	5%	1.10%

Os resultados mostram que a abordagem heurística combinada com crossover de dois pontos e taxa de mutação moderada (5%) alcançou soluções muito próximas do

ótimo, com desvios inferiores a 1%. Isso demonstra a capacidade do algoritmo genético de se aproximar de soluções ótimas mesmo em instâncias de média complexidade.

6. Resultados

Foram executadas várias combinações de parâmetros do Algoritmo Genético para resolver a instância `tsp_1.csv`. Os principais achados experimentais incluem:

- Todas as combinações retornaram o mesmo melhor valor de solução: **123,66**;
- A abordagem com **critério de parada por convergência** foi consideravelmente mais rápida (por exemplo, **0,02s**) em comparação ao uso de **número fixo de gerações** (por exemplo, **0,50s**);
- O método **heurístico de inicialização** apresentou convergência ligeiramente mais rápida do que a abordagem aleatória.

Esses dados indicam que a qualidade da solução é estável independentemente das configurações testadas, mas o tempo de execução varia de forma significativa em função dos operadores genéticos escolhidos.

7. Discussão

Com base nos resultados experimentais, é possível realizar uma análise crítica sobre o impacto dos diferentes operadores genéticos e critérios utilizados:

- **Inicialização:** A estratégia heurística baseada na vizinhança mais próxima não proporcionou melhorias na qualidade final das soluções, mas contribuiu para uma convergência mais rápida do algoritmo, reduzindo o tempo total de execução;
- **Crossover:** Os três métodos de recombinação (um ponto, dois pontos e uniforme) apresentaram desempenhos semelhantes em termos de qualidade da solução. No entanto, o crossover de dois pontos demonstrou ser ligeiramente mais estável, gerando soluções consistentes com menor variabilidade;
- **Mutação:** Aumentar a taxa de mutação para valores mais altos (como 10%) levou a uma maior diversidade populacional e exploração do espaço de busca, mas não resultou necessariamente em soluções melhores. Em alguns casos, isso aumentou o tempo de execução devido à instabilidade na convergência;
- **Critério de parada:** O uso da convergência dinâmica (ausência de melhorias em um número fixo de gerações) mostrou-se mais eficiente em termos de tempo de execução, sem prejuízo da qualidade das soluções finais.

Esses pontos reforçam a importância da escolha cuidadosa dos operadores e critérios em algoritmos evolutivos, especialmente quando se busca um equilíbrio entre desempenho computacional e qualidade de solução.

8. Conclusão

Este trabalho demonstrou a eficácia dos Algoritmos Genéticos na resolução do Problema do Caixeiro Viajante (TSP), especialmente em instâncias de pequena e média escala. Através da experimentação com diferentes configurações de operadores genéticos, foi possível avaliar o impacto individual de cada parâmetro sobre o desempenho do algoritmo.

A melhor configuração observada durante os testes foi composta por:

- Inicialização heurística;
- Crossover de dois pontos;
- Taxa de mutação de 5%;
- Critério de parada por convergência.

Essa combinação apresentou um bom equilíbrio entre tempo de execução e qualidade da solução, atingindo consistentemente os melhores resultados com menor custo computacional.

Como sugestões para trabalhos futuros, destaca-se a possibilidade de:

- Aplicar algoritmos híbridos, combinando AGs com outras metaheurísticas como Simulated Annealing ou GRASP;
- Explorar AGs multiobjetivo para considerar simultaneamente múltiplos critérios, como tempo de execução, estabilidade e robustez da solução;
- Ampliar os testes para instâncias maiores e mais complexas do TSP, avaliando a escalabilidade da abordagem.

Os resultados reforçam o potencial dos AGs como ferramenta flexível e eficiente para problemas combinatórios, desde que adequadamente parametrizados.