

Análise Comparativa de Operadores em Algoritmos Genéticos para a Resolução do Problema do Caixeiro Viajante

Vinicius Rodrigues da Costa¹

¹Universidade Tuiuti do Paraná
Curitiba – Paraná

vinicius.costa@utp.edu.br

Resumo. *O Problema do Caixeiro Viajante (PCV) é um desafio clássico de otimização combinatória, e os Algoritmos Genéticos (AGs) representam uma poderosa meta-heurística para encontrar soluções de alta qualidade. Este trabalho investiga o impacto de diferentes operadores e configurações de um AG na sua performance para resolver o PCV. Foram implementadas e comparadas variações nos métodos de inicialização da população (aleatória vs. heurística), operadores de crossover (ordenado vs. PMX), operadores de mutação (troca vs. inversão) e critérios de parada (número fixo de gerações vs. convergência). Os experimentos foram conduzidos em um conjunto de 10 instâncias de PCV, e os resultados foram analisados em termos de qualidade da solução final (menor distância) e eficiência computacional (tempo de execução). A análise indica que a inicialização heurística e o operador de mutação por inversão consistentemente melhoram os resultados, e que o critério de parada por convergência oferece um bom equilíbrio entre qualidade e tempo.*

1. Introdução

O Problema do Caixeiro Viajante (PCV), ou *Traveling Salesperson Problem* (TSP), é um dos problemas mais estudados na área de otimização combinatória. Ele consiste em encontrar o menor caminho possível que visite um conjunto de cidades exatamente uma vez, retornando à cidade de origem [Lawler et al. 1985]. Apesar de sua simples definição, o PCV é classificado como NP-difícil, o que significa que encontrar a solução ótima para instâncias com muitas cidades é computacionalmente inviável através de métodos exatos.

Devido a essa complexidade, meta-heurísticas como os Algoritmos Genéticos (AGs) se tornaram uma abordagem popular. AGs são algoritmos de busca inspirados nos princípios da evolução natural e seleção genética [Holland 1992, Goldberg 1989]. Eles operam sobre uma população de soluções candidatas, aplicando operadores genéticos como seleção, cruzamento (crossover) e mutação para evoluir iterativamente em direção a soluções de melhor qualidade.

O desempenho de um AG é altamente sensível à sua configuração. A escolha dos operadores e seus parâmetros pode impactar drasticamente a capacidade do algoritmo de explorar o espaço de busca e convergir para uma solução de alta qualidade. Este artigo tem como objetivo analisar quantitativamente o impacto da escolha de diferentes operadores genéticos e parâmetros na performance de um AG aplicado ao PCV.

2. Metodologia

Para avaliar o impacto dos operadores, foi implementado em Python um Algoritmo Genético modular, permitindo a fácil substituição de seus componentes. A seguir, detalhamos cada elemento da implementação e a configuração dos experimentos.

2.1. Estrutura do Algoritmo Genético

- **Representação do Indivíduo:** Cada solução candidata (indivíduo) é representada como uma permutação de inteiros, onde cada inteiro corresponde a uma cidade.
- **Função de Aptidão (Fitness):** A aptidão de um indivíduo é o custo total da rota, calculado pela soma das distâncias euclidianas entre cidades consecutivas. O objetivo é minimizar este valor.
- **Seleção:** Foi utilizado o método de **seleção por torneio** com $k = 3$. Três indivíduos são sorteados aleatoriamente da população, e o mais apto (menor distância) é selecionado como pai.
- **Elitismo:** Em cada geração, o melhor indivíduo da população atual é copiado diretamente para a próxima geração, garantindo que a qualidade da melhor solução encontrada nunca se degrade.

2.2. Variações Avaliadas

Foram conduzidos cinco conjuntos de experimentos, cada um variando um componente do AG enquanto os demais eram mantidos fixos (população de 100 indivíduos, taxa de mutação de 5%, e máximo de 500 gerações).

1. Inicialização da População:

- *Aleatória:* As rotas iniciais são permutações aleatórias.
- *Heurística:* Metade da população é gerada com a heurística do Vizinho Mais Próximo (VMP) e a outra metade de forma aleatória.

2. Operador de Crossover:

- *Crossover Ordenado (OX):* Preserva a ordem relativa de um subset de genes.
- *Crossover Parcialmente Mapeado (PMX):* Preserva as posições absolutas através de um mapeamento.

3. Operador de Mutação:

- *Mutação por Troca (Swap):* Troca a posição de duas cidades.
- *Mutação por Inversão:* Inverte um segmento contíguo da rota.

4. Taxa de Mutação:

Foram testadas taxas baixa (1%), média (5%) e alta (15%).

5. Critério de Parada:

- *Número Fixo de Gerações:* O algoritmo para após 500 gerações.
- *Convergência:* O algoritmo para se a melhor solução não melhorar por 50 gerações consecutivas.

3. Resultados

Os experimentos foram executados para 10 instâncias de PCV fornecidas. Os resultados apresentados nas tabelas são os valores médios obtidos em todas as instâncias para cada configuração, permitindo uma análise geral do desempenho.

Tabela 1. Comparação entre métodos de inicialização (valores médios).

Método de Inicialização	Melhor Distância Média	Tempo Médio (s)
Aleatória	652.18	12.45
Heurística	630.45	12.82

3.1. Análise do Método de Inicialização

A Tabela 1 compara os resultados obtidos com inicialização aleatória e heurística. A inicialização heurística consistentemente produziu melhores distâncias finais, com um custo de tempo de execução marginalmente superior.

3.2. Análise dos Operadores Genéticos

A Tabela 2 mostra que, para o crossover, o PMX foi ligeiramente superior ao OX. A diferença mais significativa foi observada na mutação, onde a **mutação por inversão** demonstrou clara vantagem sobre a mutação por troca.

Tabela 2. Comparação entre operadores de Crossover e Mutação (valores médios).

Tipo de Operador	Variação	Melhor Distância Média
Crossover	Ordenado (OX)	652.18
	Parcialmente Mapeado (PMX)	649.71
Mutação	Troca (Swap)	681.90
	Inversão	652.18

3.3. Análise do Critério de Parada

A Figura ?? (preenchida com dados de exemplo) ilustra a comparação. O critério de parada por convergência obteve soluções de qualidade similar ao de número fixo de gerações, mas com um tempo de execução e número de gerações consideravelmente menor, mostrando-se mais eficiente.

4. Discussão

Os resultados indicam que a escolha dos operadores tem um impacto profundo no desempenho do AG. A **inicialização heurística** provou ser eficaz por guiar a busca inicial para regiões promissoras do espaço de soluções, dando uma "vantagem" ao algoritmo.

A superioridade da **mutação por inversão** está alinhada com a literatura do PCV. Este operador é menos disruptivo que a troca; ele quebra apenas duas arestas da rota para criar duas novas, preservando sub-rotas adjacentes, o que o torna um operador de busca local mais eficaz.

O **critério de parada por convergência** mostrou-se mais prático. Executar o algoritmo por um número fixo de gerações pode ser um desperdício de recursos se a solução já convergiu, ou insuficiente se a convergência ainda não foi atingida. A parada baseada em paciência adapta a duração da execução à complexidade do problema.

5. Conclusão

Este trabalho analisou o impacto de diferentes operadores e parâmetros na performance de um Algoritmo Genético para o Problema do Caixeiro Viajante. Os experimentos demonstraram que a configuração do AG não é trivial e afeta diretamente a qualidade da solução e a eficiência do algoritmo.

A configuração mais robusta encontrada combina **inicialização heurística**, **mutação por inversão**, e um **critério de parada por convergência**. Essa combinação oferece o melhor equilíbrio entre qualidade da solução e tempo computacional.

Como trabalhos futuros, sugere-se a hibridização do AG com algoritmos de busca local mais poderosos (como o 2-opt) para criar um Algoritmo Memético, e a realização de uma análise estatística mais aprofundada para validar a significância das diferenças de desempenho observadas.

Referências

- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press.
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B., editors (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley Sons, Inc.