

Meta-heurísticas para o Problema do Caixeiro Viajante

VNS, GRASP e Algoritmo Genético

André Costa Batista

ELE634 Laboratório de Sistemas II
Escola de Engenharia
Universidade Federal de Minas Gerais

05 de setembro de 2025

- 1 O Problema do Caixeiro Viajante (PCV)
- 2 Variable Neighborhood Search (VNS)
- 3 Greedy Randomized Adaptive Search Procedure (GRASP)
- 4 Algoritmo Genético (AG)
- 5 Conclusão

O Problema do Caixeiro Viajante (PCV)

O Problema do Caixeiro Viajante (PCV)

Definição do Problema

Definição

Dado um conjunto de cidades e as distâncias entre cada par delas, o objetivo é encontrar a rota mais curta possível que visita cada cidade exatamente uma vez e retorna à cidade de origem.

- É um dos problemas de otimização combinatória mais famosos e estudados.
- Pertence à classe de problemas **NP-difícil**, o que significa que encontrar a solução ótima exata para um grande número de cidades é computacionalmente inviável.
- Por isso, usamos **meta-heurísticas** para encontrar soluções de alta qualidade em um tempo razoável.

Variable Neighborhood Search (VNS)

Princípio Fundamental

A ideia central do VNS é a exploração sistemática de diferentes **estruturas de vizinhança** para evitar ficar preso em ótimos locais.

- Proposto por Mladenović e Hansen em 1997.
- Baseia-se no fato de que um **ótimo local** em uma estrutura de vizinhança não é, necessariamente, um ótimo local em outra.
- Ao mudar a vizinhança, a busca pode escapar de vales e explorar outras regiões do espaço de soluções.

O VNS alterna sistematicamente entre três fases:

1. Perturbação (Shaking)

A partir da solução atual (s), gera-se uma solução vizinha aleatória (s') em uma **vizinhança** k . A busca começa com $k = 1$.

2. Busca Local

A solução vizinha (s') é melhorada usando um método de **busca local**, resultando em uma nova solução (s''). Geralmente, a busca local usa uma vizinhança mais simples (ex: $k = 1$).

3. Movimento (Move)

- Se a solução melhorada (s'') for melhor que a solução original (s), ela é aceita como a nova solução e a busca retorna para a primeira vizinhança ($k \leftarrow 1$).
- Caso contrário, a busca avança para a próxima estrutura de vizinhança ($k \leftarrow k + 1$).

Para o PCV, uma solução é uma rota (permutação de cidades). As vizinhanças são geradas por movimentos:

- **Troca (Swap):**

- Troca a posição de duas cidades na rota.
- Ex: $[A, B, C, D, E] \rightarrow [A, D, C, B, E]$

- **Inserção (Insertion):**

- Remove uma cidade e a insere em outra posição.
- Ex: $[A, B, C, D, E] \rightarrow [A, C, D, B, E]$

- **Inversão (2-opt):**

- Inverte um segmento da rota entre duas cidades.
- Ex: $[A, B, C, D, E] \rightarrow [A, D, C, B, E]$

Greedy Randomized Adaptive Search Procedure (GRASP)

Princípio Fundamental

O GRASP é uma meta-heurística *multi-start*, onde cada iteração consiste em duas fases: uma de **construção semi-gulosa** e outra de **busca local**.

- O componente-chave é a **Lista Restrita de Candidatos (RCL)**.
- A RCL armazena os melhores candidatos para o próximo passo da construção da solução.
- Um elemento é **sorteado aleatoriamente** da RCL, introduzindo diversidade na busca.
- O parâmetro α (0 a 1) controla o quão “gulosa” ou “aleatória” é a construção.

Fluxo de uma Iteração GRASP

1 Fase de Construção:

- Começa com uma rota vazia ou com uma cidade aleatória.
- Repetidamente, constrói a **RCL** com as cidades vizinhas de menor custo (mais próximas).
- Sorteia uma cidade da RCL e a adiciona à rota.
- Continua até que todas as cidades tenham sido visitadas.

2 Fase de Busca Local:

- A solução construída é usada como ponto de partida para uma busca local (ex: 2-opt).
- O objetivo é refinar a solução, encontrando um ótimo local próximo.

O algoritmo repete essas duas fases por um número definido de iterações e retorna a **melhor solução** encontrada.

Algoritmo Genético (AG)

Princípio Fundamental

O Algoritmo Genético (AG) é uma técnica de busca inspirada na **teoria da evolução de Charles Darwin** e nos princípios da genética.

- Opera sobre uma **população** de soluções candidatas.
- Evolui essa população ao longo de **gerações** para encontrar soluções cada vez melhores.
- Utiliza operadores inspirados na biologia: **Seleção**, **Cruzamento** (Crossover) e **Mutação**.

- **Indivíduo (Cromossomo):** Representa uma solução. No PCV, é uma rota (ex: [3, 1, 4, 2, 0]).
- **População:** Um conjunto de indivíduos (um conjunto de rotas).
- **Função de Aptidão (Fitness):** Avalia a qualidade de um indivíduo. No PCV, é o inverso da distância total da rota. (*Rotas mais curtas são mais aptas*).
- **Seleção:** Escolhe os indivíduos mais aptos (pais) para se reproduzirem. (Ex: Seleção por Torneio).
- **Cruzamento (Crossover):** Combina o material genético de dois pais para criar filhos. (Ex: Crossover Ordenado).
- **Mutação:** Introduce pequenas alterações aleatórias nos filhos para garantir diversidade. (Ex: Mutação de Troca).
- **Elitismo:** Garante que os melhores indivíduos de uma geração passem para a próxima.

Processo Evolutivo

- ① **Inicialização:** Cria-se uma população inicial de soluções aleatórias (rotas).
- ② **Loop de Gerações:** Para cada geração:
 - **Avaliação:** Calcula-se a aptidão de cada indivíduo.
 - **Seleção:** Pais são selecionados com base na aptidão.
 - **Reprodução:** Operadores de Crossover e Mutação são aplicados para gerar uma nova população de descendentes.
 - **Substituição:** A nova população substitui a antiga (frequentemente mantendo os melhores através do Elitismo).
- ③ **Término:** O algoritmo para após um número definido de gerações ou outro critério, retornando o melhor indivíduo encontrado.

Conclusão

- **VNS**, **GRASP** e **AG** são meta-heurísticas tradicionais para resolver problemas combinatórios como o PCV.
- **VNS** foca na mudança sistemática de vizinhanças para escapar de ótimos locais.
- **GRASP** equilibra a construção gulosa com aleatoriedade, refinando soluções com busca local em um processo multi-start.
- **Algoritmo Genético** utiliza uma abordagem populacional, imitando a evolução natural para explorar o espaço de busca.
- A escolha da melhor abordagem depende das características do problema, do tempo computacional disponível e da qualidade de solução desejada.
- Outras meta-heurísticas, como **Simulated Annealing**, **Ant Colony Optimization** e **Particle Swarm Optimization**, também são opções viáveis para o PCV.

Referências Bibliográficas

-  Gaspar-Cunha, A., Takahashi, R., & Antunes, C. H. (2012). *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra/Coimbra University Press.
-  Gendreau, M., & Potvin, J. Y. (Eds.). (2010). *Handbook of Metaheuristics* (2nd ed.). Springer.
-  Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097-1100.
-  Feo, T. A., & Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2), 109-133.
-  Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.