

Trabalho Final: Aplicação e Comparação de Metaheurísticas

Disciplina: Metaheurísticas e Aplicações
Professor: Franklin Angelo Krukoski

Data de Divulgação: 03/05/2025
Data de Entrega: 24/05/2025

1 Visão Geral

Este trabalho final tem como objetivo consolidar os conhecimentos adquiridos nas aulas sobre metaheurísticas, aplicando três algoritmos específicos – **Otimização por Enxame de Partículas (PSO)**, **Busca Tabu (TS)** e **Algoritmos de Colônia de Formigas (ACO)** – para resolver um problema de otimização complexo escolhido dentre as opções fornecidas. O foco estará na implementação, experimentação, análise comparativa de desempenho e escalabilidade das metaheurísticas.

2 Escolha do Problema

O aluno deverá escolher **UM** dos três problemas de otimização listados abaixo para implementar e analisar as três metaheurísticas (PSO, TS, ACO).

2.1 Opção 1: Problema do Caixeiro Viajante com Janelas de Tempo (TSPWT - TSP with Time Windows)

- **Descrição:** Extensão do TSP clássico. Além de encontrar a rota mais curta que visita cada cidade uma vez, cada cidade i possui uma janela de tempo $[e_i, l_i]$ (earliest e latest arrival time) na qual a visita deve ocorrer. O veículo também tem tempos de viagem t_{ij} entre as cidades i e j e, opcionalmente, um tempo de serviço s_i em cada cidade.
- **Objetivo:** Minimizar a distância total percorrida (ou tempo total da rota).
- **Restrições Principais:**
 - Cada cidade é visitada exatamente uma vez.
 - A chegada na cidade i deve ocorrer dentro da janela $[e_i, l_i]$. Se chegar antes de e_i , o veículo deve esperar. Chegar após l_i torna a solução inviável.
 - A sequência da rota deve respeitar os tempos de viagem e serviço.

- **Entradas Típicas:** Coordenadas ou matriz de distância/tempo, lista de janelas de tempo $[e_i, l_i]$ para cada cidade, tempos de serviço s_i (opcional).
- **Saída:** A permutação das cidades (rota) que minimiza o custo e satisfaz todas as restrições de tempo.
- **Complexidade Adicional:** As janelas de tempo tornam o problema significativamente mais restrito e complexo que o TSP padrão. A viabilidade de uma rota depende não apenas da sequência, mas também dos tempos de chegada.

2.2 Opção 2: Problema de Escala de Funcionários com Maximização da Satisfação

- **Descrição:** Criar uma escala de trabalho semanal/mensal para um conjunto de funcionários, definindo quais turnos cada um trabalhará, de forma a atender às necessidades de cobertura de pessoal e, ao mesmo tempo, maximizar a satisfação geral dos funcionários, considerando suas preferências e restrições.
- **Objetivo:** Maximizar uma função que representa a satisfação total dos funcionários (pode ser baseada em preferências por turnos, dias de folga, sequências de trabalho, etc.) OU Minimizar a insatisfação/penalidades por violação de preferências.
- **Restrições Principais (Exemplos):**
 - Número mínimo/máximo de funcionários por turno/dia.
 - Qualificações necessárias para certos turnos.
 - Limites de horas de trabalho por funcionário (diário, semanal).
 - Restrições sobre sequências de turnos (e.g., descanso mínimo entre turnos).
 - Indisponibilidades ou preferências explícitas dos funcionários.
- **Entradas Típicas:** Lista de funcionários, lista de turnos, requisitos de cobertura, matriz de preferências/custos/satisfação de cada funcionário para cada turno/padrão, regras trabalhistas.
- **Saída:** Uma matriz ou estrutura de dados indicando qual funcionário trabalha em qual turno/dia.
- **Complexidade Adicional:** Lida com múltiplas restrições complexas e um objetivo que pode ser subjetivo (satisfação). A representação da solução pode ser desafiadora.

2.3 Opção 3: Problema de Roteamento em Arcos Capacitado (CARP - Capacitated Arc Routing Problem)

- **Descrição:** Determinar um conjunto de rotas de veículos de custo mínimo, partindo e retornando a um depósito central, para atender à demanda de um conjunto de **arcos (arestas ou links)** necessários em uma rede. Cada veículo tem uma capacidade limitada, e cada arco tem uma demanda associada. Diferente do VRP/TSP que roteia nós, o CARP roteia para servir arcos/arestas.

- **Objetivo:** Minimizar o custo total de viagem (distância ou tempo) de todos os veículos.
- **Restrições Principais:**
 - Todos os arcos com demanda positiva devem ser servidos exatamente uma vez por algum veículo.
 - Cada rota de veículo deve começar e terminar no depósito.
 - A demanda total acumulada em uma rota não pode exceder a capacidade do veículo.
- **Entradas Típicas:** Grafo da rede (nós, arcos, custos de travessia), localização do depósito, lista de arcos requeridos com suas demandas, capacidade dos veículos.
- **Saída:** Um conjunto de sequências de nós/arcos representando as rotas de cada veículo.
- **Complexidade Adicional:** A solução envolve tanto a seleção de quais arcos servir em cada rota quanto a sequência ótima para servi-los, respeitando a capacidade. A representação pode ser mais complexa que permutações simples.

3 Requisitos de Implementação

- **Linguagens Permitidas:** O código deve ser desenvolvido em **R**, **Python** ou **Matlab**. A escolha é livre, mas deve ser consistente para todas as metaheurísticas.
- **Metaheurísticas:** Implementar **PSO**, **TS** e **ACO** para o problema escolhido.
 - **Adaptações:** Para PSO, será necessário adaptar o algoritmo para o domínio combinatório do problema (e.g., usando Random Keys, PSO discreto, ou outra técnica apropriada). Detalhe a adaptação realizada.
 - **Componentes:** As implementações devem refletir claramente os conceitos vistos em aula (e.g., estrutura de vizinhança e lista tabu/aspiração para TS, feromônio/heurística/construção para ACO, etc.).
- **Código Fonte:** O código deve ser claro, bem comentado e organizado (e.g., funções separadas para cálculo de custo, operadores, etc.). Inclua um arquivo README ou comentários no início explicando como executar o código e quais são os principais parâmetros.
- **Instâncias do Problema:** Utilize instâncias de benchmark conhecidas para o problema escolhido (se disponíveis e adequadas) ou gere instâncias com diferentes tamanhos para análise de escalabilidade. Forneça as instâncias utilizadas junto com o código ou indique a fonte.

4 Análise Experimental e Comparativa

O núcleo do trabalho é a análise comparativa das metaheurísticas implementadas. O aluno deve:

- **Parâmetros:** Investigar (mesmo que brevemente) o efeito de alguns parâmetros chave de cada metaheurística no desempenho (e.g., tamanho da população/enxame, taxas de GA/PSO, duração tabu, parâmetros α, β, ρ do ACO). Justifique a escolha final dos parâmetros usados na comparação principal.
- **Métricas de Desempenho:** Comparar os algoritmos usando pelo menos duas métricas:
 - **Qualidade da Solução:** Melhor valor da função objetivo encontrado, média ao longo de múltiplas execuções (devido à natureza estocástica).
 - **Esforço Computacional:** Tempo de execução (CPU time).
- **Escalabilidade:** Executar os algoritmos em instâncias de tamanhos variados (e.g., pequeno, médio, grande) para analisar como a qualidade da solução e o tempo de execução escalam com o tamanho do problema para cada metaheurística. Apresente os resultados graficamente (e.g., custo vs tamanho, tempo vs tamanho).
- **Discussão:** Analisar criticamente os resultados, discutindo os pontos fortes e fracos de cada metaheurística no contexto do problema escolhido e dos resultados obtidos. Qual algoritmo pareceu mais adequado e por quê? Houve dificuldades na adaptação de algum deles?

5 Entregáveis

O trabalho final deverá ser entregue através de uma pasta compartilhada no Dropbox (detalhes abaixo) contendo:

1. **Código Fonte:** Uma pasta contendo todo o código implementado (arquivos ‘.R’, ‘.py’, ou ‘.m’), incluindo as instâncias de teste utilizadas ou instruções claras para obtê-las. O código deve ser executável. Inclua um arquivo ‘README.md’ (ou similar) com instruções de execução.
2. **Relatório Técnico:** Um documento em formato PDF (preferencialmente gerado a partir de LaTeX, Word ou similar) contendo:
 - Introdução (apresentação do problema e objetivos).
 - Descrição detalhada do problema escolhido.
 - Descrição das metaheurísticas implementadas e das adaptações realizadas (especialmente para PSO). Detalhe a representação, vizinhança, operadores, memória, etc., de cada uma.
 - Metodologia Experimental (instâncias usadas, parâmetros testados/escolhidos, número de execuções, métricas coletadas, ambiente computacional).

- Resultados e Análise (tabelas e gráficos comparando qualidade da solução, tempo de execução e escalabilidade).
- Discussão e Conclusões (interpretação dos resultados, comparação crítica, dificuldades encontradas, sugestões futuras).
- Referências.

6 Critérios de Avaliação

O trabalho será avaliado com base nos seguintes critérios:

- Correta implementação das metaheurísticas e adaptações (30%).
- Qualidade e profundidade da análise experimental e comparativa (incluindo escalabilidade e parâmetros) (30%).
- Clareza, organização e completude do relatório técnico (incluindo gráficos e tabelas) (30%).
- Qualidade e organização do código fonte (comentários, clareza, executabilidade) (10%).

7 Colaboração e Compartilhamento

- **Dropbox:** Cada grupo deverá criar uma pasta no Dropbox ou Google Drive e compartilhá-la com o professor (email: franklin.krukoski@gmail.com) até 10/05/2025. O nome da pasta deve seguir o padrão: ‘Metaheurísticas/TrabalhoFinal/SeuNome’ (ou ‘/NomesGrupo’). Todo o desenvolvimento e os entregáveis finais devem ser colocados nesta pasta. O professor poderá acompanhar o progresso através da pasta compartilhada.
- **Integridade Acadêmica:** O trabalho é individual (ou em grupo, conforme definido). Espera-se que todo o código e texto do relatório sejam originais. A utilização de bibliotecas externas para funções auxiliares (e.g., leitura de dados, plots) é permitida, mas o núcleo das metaheurísticas deve ser implementado pelo aluno/grupo. Qualquer fonte externa utilizada deve ser devidamente citada. Casos de plágio resultarão em nota zero.

8 Prazos e Contato

- **Data Limite para Escolha do Problema:** 05/05/2025
- **Data Limite para Criação da Pasta Dropbox:** 10/05/2025
- **Data Final de Entrega (Código + Relatório):** 24/05/2025
- **Dúvidas:** Entrar em contato com o professor via franklin.krukoski@gmail.com .

Bom trabalho!