Universidade Federal da Bahia
Instituto de Computação
MATA53 - Teoria dos grafos

Relatório: Problema do caixeiro viajante com bônus e passageiros

Antoniel Magalhães
João Leahy
Luis Felipe

Salvador - Bahia
6 de janeiro de 2025

Relatório: Problema do caixeiro viajante com bônus e passageiros

Antoniel Magalhães
João Leahy
Luis Felipe

Estudo dirigido entregue ao professor Islame Felipe da Costa Fernandes como método avaliativo da disciplina MATA53 - Teoria dos grafos

Salvador - Bahia
6 de janeiro de 2025

Sumário

0.1 Contextualização e Relevância 2
0.2 Objetivos e Escopo 2

1 Fundamentação Teórica 3
1.1 Problema do Caixeiro Viajante (TSP) 3
1.2 Extensões do TSP 4
1.3 Definição Formal do Problema 4
1.4 Complexidade Computacional 5

2 Modelagem Matemática 6
2.1 Variáveis de Decisão 6
2.2 Função Objetivo 6
2.3 Restrições de Fluxo 7
2.4 Restrições de Conectividade 7
2.4.1 Restrições de Passageiros 8

3 Métodos de Resolução 10
3.1 Introdução 10
3.2 Métodos Utilizados 10

4 Resultados e Discussão 15
4.1 Resultados Computacionais 15
4.1.1 Análise Comparativa dos Métodos 15
4.1.2 Análise de Desempenho 16
4.2 Discussão 16
4.2.1 Implicações Teóricas 16
4.2.2 Implicações Práticas 17
4.2.3 Limitações e Desafios 17
4.3 Direções Futuras 18

5 Conclusão **19**
5.1 Síntese dos Resultados 19
5.2 Contribuições 20
5.3 Considerações Finais 20
5.4 Recomendações 21

Referências Bibliográficas **22**

Introdução

O **Problema do Caixeiro Viajante com Coleta Opcional de Bônus e Passageiros (TSP-OBP)**, também referenciado na literatura como **Problema do Caixeiro Viajante com Coleta de Prêmios (PCVCP)**, é uma extensão do clássico Problema do Caixeiro Viajante (PCV). Este problema combina a necessidade de coletar **bônus ou prêmios** em determinados locais, o transporte de **passageiros** e a otimização de rotas, excluindo o tempo de coleta, o que o torna relevante para aplicações como **logística, sistemas de entrega e serviços de transporte compartilhado**. Estudos como o de **Lopes Filho [3]** e **Carvalho [2]** destacam a importância do TSP-OBP e suas variantes na otimização de recursos e melhoria da eficiência operacional em sistemas complexos. Embora a tese de Lopes Filho [3] inclua o tempo de coleta, o presente trabalho foca na variante sem esta restrição.

A complexidade inerente ao TSP-OBP advém da necessidade de equilibrar objetivos conflitantes: **maximizar a coleta de bônus ou prêmios, minimizar a distância total percorrida e satisfazer as restrições de transporte de passageiros**, tais como a capacidade do veículo. Esta natureza multiobjetivo, juntamente com as restrições combinatórias, torna o problema **NP-difícil**, requerendo abordagens computacionais sofisticadas para sua resolução.

No contexto atual, onde a otimização de recursos e a eficiência operacional são cruciais, o TSP-OBP emerge como uma ferramenta essencial para modelar e resolver problemas complexos de roteamento. As aplicações do TSP-OBP abrangem desde sistemas de entrega com incentivos até serviços de transporte sob demanda, onde a flexibilidade na coleta de bônus e a gestão eficiente de passageiros são fundamentais. Esta modelagem é relevante em situações de logística, transporte de pessoas, e outros sistemas de roteamento que necessitem de coleta seletiva com passageiros.

Este relatório tem como objetivo apresentar uma análise do problema, incluindo suas variantes, a modelagem matemática, os métodos de resolução (heurísticas, meta-heurísticas, algoritmos híbridos) e suas aplicações práticas. Além disso, exploraremos as contribuições teóricas e práticas que o TSP-OBP oferece para a pesquisa em otimização combinatória, com foco na variante sem tempo de coleta. A estrutura do trabalho está organizada para proporcionar uma compreensão progressiva do tema, desde os fundamentos teóricos até as aplicações e perspectivas futuras.

0.1 Contextualização e Relevância

O TSP-OBP surge como uma resposta natural à evolução dos sistemas de transporte e logística modernos, onde a simples otimização de rotas já não é suficiente para atender às demandas complexas do mercado. A incorporação de bônus e passageiros ao problema clássico do caixeiro viajante reflete a necessidade de modelos mais sofisticados, capazes de capturar a realidade multifacetada dos sistemas de transporte contemporâneos.

Em termos práticos, o problema encontra aplicações em diversos cenários:

- Sistemas de compartilhamento de viagens, onde motoristas podem coletar passageiros e bônus ao longo de suas rotas;
- Serviços de entrega com incentivos por coleta ou entrega em determinados pontos;
- Planejamento de roteiros turísticos com pontos de interesse prioritários;
- Otimização de rotas para veículos de transporte público sob demanda.

0.2 Objetivos e Escopo

O presente trabalho tem como objetivos específicos:

- Analisar a estrutura matemática do TSP-OBP e suas propriedades fundamentais;
- Investigar os métodos de resolução existentes e suas aplicabilidades;
- Avaliar o impacto das diferentes abordagens na qualidade das soluções obtidas;
- Discutir as implicações práticas e teóricas das variantes do problema;

O escopo do trabalho abrange desde a fundamentação teórica até as aplicações práticas. São consideradas tanto as abordagens exatas quanto as heurísticas, bem como as implicações de diferentes estruturas de dados e algoritmos na eficiência das soluções propostas.

Capítulo 1

Fundamentação Teórica

1.1 Problema do Caixeiro Viajante (TSP)

O **Problema do Caixeiro Viajante (PCV)**, ou Traveling Salesman Problem (TSP), consiste em encontrar o ciclo hamiltoniano de menor custo em um grafo ponderado, onde o ciclo visita cada vértice (cidade) exatamente uma vez e retorna ao vértice de origem [4]. Este problema é um dos clássicos da otimização combinatória e tem sido amplamente estudado na teoria da computação devido à sua complexidade e aplicabilidade. Como apontam [4, 3], a introdução de variáveis adicionais como **bônus** e **passageiros**, no **Problema do Caixeiro Viajante com Coleta Opcional de Bônus e Passageiros (TSP-OBP)**, aumenta significativamente a complexidade do problema original, exigindo novas abordagens de modelagem e solução.

Matematicamente, o TSP pode ser representado como um grafo completo $G = (N, M)$, onde N é o conjunto de vértices (cidades) e M é o conjunto de arestas (conexões entre cidades). Cada aresta $(i, j) \in M$ possui um custo associado c_{ij}, que representa a distância ou o tempo de viagem entre as cidades i e j [4]. O objetivo é encontrar um **ciclo hamiltoniano** de custo mínimo, ou seja, um caminho que visite todos os vértices exatamente uma vez e retorne ao ponto inicial.

A natureza **NP-difícil** do TSP implica que não existe um algoritmo conhecido que possa resolver o problema em tempo polinomial para todas as instâncias [4, 1]. Essa característica fundamental do problema base apresenta desafios significativos para sua resolução, que são amplificados quando consideramos extensões como o TSP-OBP, onde a coleta de bônus e o transporte de passageiros introduzem novas dimensões de complexidade combinatória e decisões a serem tomadas [3, 2]. O livro de Goldbarg detalha vários algoritmos para o PCV, incluindo heurísticas e métodos de aproximação, que podem servir como ponto de partida para a compreensão de problemas mais complexos, como o TSP-OBP.

1.2 Extensões do TSP

As extensões do TSP, como o TSP com Bônus e o TSP-OBP, introduzem novas dimensões ao problema original, exigindo a consideração de múltiplos objetivos e restrições. Essas variações são fundamentais para capturar a complexidade de cenários reais, onde decisões de roteamento devem equilibrar custos, prêmios e restrições de passageiros [2].

Entre as principais extensões, podemos destacar:

- **TSP com Prêmios (PTSP)**: Incorpora valores de bônus associados à visita de determinados vértices;
- **TSP com Janelas de Tempo (TSPTW)**: Adiciona restrições temporais para a visita aos vértices;
- **TSP com Coleta e Entrega (PDTSP)**: Inclui operações de coleta e entrega de itens entre vértices;
- **TSP com Múltiplos Veículos (mTSP)**: Considera uma frota de veículos para realizar as visitas.

O TSP-OBP combina elementos dessas diferentes variantes, incorporando tanto aspectos de coleta de bônus quanto o gerenciamento de passageiros, o que resulta em um problema significativamente mais complexo.

1.3 Definição Formal do Problema

O TSP-OBP pode ser formalizado como um problema de otimização em grafos, onde o objetivo é maximizar a coleta de bônus e minimizar o custo total de deslocamento, respeitando as restrições de passageiros. Esta definição formal permite a aplicação de técnicas avançadas de otimização para encontrar soluções eficientes.

Formalmente, o TSP-OBP pode ser definido como:

- **Grafo**: $G = (V, E)$, onde V é o conjunto de vértices e E o conjunto de arestas;
- **Custos**: c_{ij} representa o custo de viagem entre os vértices i e j;
- **Bônus**: b_i é o valor do bônus associado ao vértice $i \in V$;
- **Passageiros**: P é o conjunto de passageiros, onde cada p $\in$ P possui:
  - origem o_p e destino d_p
  - janela de tempo $[t_{p\_inicio}, t_{p\_fim}]$ para embarque

– tempo máximo de viagem t_{max_p}

A função objetivo do TSP-OBP pode ser expressa como uma combinação ponderada entre a maximização dos bônus coletados e a minimização dos custos de viagem:

$$\max \sum_{i \in V} b_i x_i - \alpha \sum_{(i,j) \in E} c_{ij} y_{ij} \tag{1.1}$$

onde x_i é uma variável binária que indica se o vértice i é visitado, y_{ij} indica se a aresta (i, j) é utilizada na solução, e α é um parâmetro de balanceamento entre os objetivos [3].

1.4 Complexidade Computacional

A análise de complexidade do TSP-OBP revela que o problema é NP-difícil, uma vez que contém o TSP clássico como caso especial. Além disso, a adição de restrições de passageiros e bônus introduz novas dimensões de complexidade, tornando o problema ainda mais desafiador do ponto de vista computacional [2].

Alguns aspectos que contribuem para a complexidade do problema incluem:

- Natureza combinatória da seleção de vértices para visita;
- Interdependência entre as decisões de roteamento e coleta de bônus;
- Restrições temporais e de capacidade relacionadas aos passageiros;
- Necessidade de coordenação entre múltiplos objetivos conflitantes.

Esta complexidade inerente motiva o desenvolvimento de diferentes abordagens de solução, desde métodos exatos para instâncias pequenas até heurísticas e metaheurísticas para problemas de maior escala.

Capítulo 2

Modelagem Matemática

2.1 Variáveis de Decisão

A modelagem matemática do TSP-OBP envolve a definição de variáveis de decisão que capturam a complexidade do problema. Além das variáveis tradicionais do TSP, o TSP-OBP requer a consideração de variáveis que representam a coleta de bônus e o transporte de passageiros. Este aumento na complexidade demanda o uso de algoritmos avançados para encontrar soluções eficientes [2].

As principais variáveis de decisão do modelo são:

- x_{ij}: Variável binária que indica se a aresta (i,j) é utilizada na solução

$$x_{ij} = \begin{cases} 1, & \text{se o arco (i,j) é utilizado na rota} \\ 0, & \text{caso contrário} \end{cases} \tag{2.1}$$

- y_i: Variável binária que indica se o vértice i é visitado

$$y_i = \begin{cases} 1, & \text{se o vértice i é visitado} \\ 0, & \text{caso contrário} \end{cases} \tag{2.2}$$

- t_i: Variável contínua que representa o instante de chegada no vértice i
- p_{ik}: Variável binária que indica se o passageiro k está no veículo ao visitar o vértice i

2.2 Função Objetivo

A função objetivo do TSP-OBP é maximizar a soma dos bônus coletados e minimizar o custo total de viagem. Esta função deve ser cuidadosamente balanceada para garantir que as soluções propostas sejam viáveis e otimizadas em termos de custo-benefício.

A formulação matemática da função objetivo pode ser expressa como:

$$\max \sum_{i \in V} b_i y_i - \alpha \sum_{(i,j) \in E} c_{ij} x_{ij} \tag{2.3}$$

onde:

- b_i é o valor do bônus associado ao vértice i
- c_{ij} é o custo de viagem entre os vértices i e j
- α é o parâmetro de balanceamento entre bônus e custos

2.3 Restrições de Fluxo

$$\sum_{j \in V} x_{ij} = y_i \qquad \forall i \in V \tag{2.4}$$

$$\sum_{i \in V} x_{ij} = y_j \qquad \forall j \in V \tag{2.5}$$

A **Equação** (2.4) garante que, para cada vértice i, se ele está na solução ($y_i = 1$), então existe uma aresta saindo de i. Já a **Equação** (2.5) assegura que, se o vértice j está na solução ($y_j = 1$), então há uma aresta chegando em j. Essas restrições são fundamentais para garantir a continuidade da rota.

Essas restrições são chamadas de **restrições de atribuição** [3, 2]. As variáveis x_{ij} são **binárias**, indicando se a aresta (i, j) faz parte da rota. As variáveis y_i também são **binárias**, representando se o vértice i está na rota ou não.

2.4 Restrições de Conectividade

Eliminação de Subciclos com Miller-Tucker-Zemlin (MTZ)

Para garantir que a solução seja um único ciclo Hamiltoniano, e não múltiplos subciclos desconexos, utilizamos as restrições propostas por Miller, Tucker e Zemlin (MTZ). Estas restrições são formuladas como:

$$u_i - u_j + n x_{ij} \leq n - 1 \quad \forall i, j \in V, i \neq j \tag{2.6}$$

onde:

- u_i é uma variável auxiliar **inteira** que representa a ordem de visita do vértice i na rota.

- n representa o número total de vértices no grafo.
- x_{ij} é uma variável **binária** que indica se a aresta que conecta diretamente o vértice i ao vértice j faz parte da solução. Se $x_{ij} = 1$, a aresta está na rota; caso contrário, $x_{ij} = 0$.
- V representa o conjunto de todos os vértices no grafo.

Explicação Detalhada

O objetivo principal dessas restrições é **eliminar a formação de subciclos** que não incluem todos os vértices do grafo. A ideia central é que, se um ciclo Hamiltoniano é formado, os valores de u_i devem representar uma sequência coerente de visitas aos vértices.

2.4.1 Restrições de Passageiros

As restrições de passageiros visam garantir a viabilidade e a conformidade do roteiro em relação aos passageiros e suas demandas.

- **Restrição de Atendimento Único:**

$$\sum_{i \in V} p_{ik} \leq 1 \quad \forall k \in P \tag{2.7}$$

  Essa restrição garante que cada passageiro k no conjunto de passageiros P seja atendido no máximo uma vez. A variável binária p_{ik} indica se o passageiro k foi embarcado no vértice i. A soma de todas as localizações de embarque de um passageiro deve ser menor ou igual a 1, garantindo que um passageiro não será embarcado em mais de um vértice.

- **Restrição de Tempo de Chegada (Sequência Temporal):**

$$t_i + s_i + c_{ij} - M(1 - x_{ij}) \leq t_j \quad \forall i, j \in V \tag{2.8}$$

  Essa restrição assegura a **sequência temporal** da rota, onde:

  - t_i é o tempo de chegada no vértice i.
  - s_i é o tempo de serviço ou coleta no vértice i.
  - c_{ij} é o tempo de deslocamento da aresta entre o vértice i e o vértice j.
  - x_{ij} é uma variável binária que indica se a aresta (i, j) faz parte da rota.
  - M é uma constante suficientemente grande.

Se a aresta (i, j) é utilizada ($x_{ij} = 1$), a restrição garante que o tempo de chegada no vértice j (t_j) seja maior ou igual ao tempo de chegada no vértice i (t_i), somado ao tempo de serviço em i (s_i) e ao tempo de deslocamento de i para j (c_{ij}). O termo $M(1 - x_{ij})$ desativa a restrição caso a aresta (i, j) não seja utilizada ($x_{ij} = 0$). Essa restrição é similar à utilizada para eliminar subciclos.

- **Restrição de Janela de Tempo - Limite Inferior:**

$$t_i \geq e_i \quad \forall i \in V \tag{2.9}$$

Essa restrição garante que o tempo de chegada no vértice i (t_i) deve ser maior ou igual ao limite inferior da janela de tempo do vértice i (e_i). Ela assegura que o serviço em um vértice ou embarque de um passageiro ocorra depois de um determinado tempo limite.

- **Restrição de Janela de Tempo - Limite Superior:**

$$t_i \leq l_i \quad \forall i \in V \tag{2.10}$$

Essa restrição garante que o tempo de chegada no vértice i (t_i) não deve ultrapassar o limite superior da janela de tempo do vértice i (l_i). Ela assegura que o serviço em um vértice ou embarque de um passageiro em um vértice ocorra antes de um determinado tempo limite. Combinadas, as restrições de janela de tempo (limite inferior e superior) garantem que o serviço ou embarque ocorram dentro de um intervalo de tempo aceitável para o vértice i.

Variáveis e Parâmetros:

- s_i: Tempo de serviço/coleta no vértice i.
- e_i: Limite inferior da janela de tempo para o vértice i.
- l_i: Limite superior da janela de tempo para o vértice i.
- M: Constante suficientemente grande.
- p_{ik}: Variável binária que indica se o passageiro k foi embarcado no vértice i.
- t_i: Tempo de chegada no vértice i.
- c_{ij}: Custo ou tempo de deslocamento entre os vértices i e j.
- x_{ij}: Variável binária que indica se a aresta (i, j) faz parte da rota.
- P: Conjunto de passageiros.
- V: Conjunto de vértices.

Capítulo 3

Métodos de Resolução

3.1 Introdução

Para abordar o problema de forma acessível e eficaz, optou-se por utilizar dois métodos principais: **GRASP (Greedy Randomized Adaptive Search Procedure)** e **Atribuição Heurística de Passageiros**. Estes métodos foram escolhidos por sua simplicidade conceitual e capacidade de gerar soluções de alta qualidade em tempo computacional reduzido.

3.2 Métodos Utilizados

GRASP (Greedy Randomized Adaptive Search Procedure)

O GRASP é uma meta-heurística que combina uma fase de construção semi-gulosa com uma fase de busca local para encontrar soluções aproximadas para problemas de otimização combinatória. O objetivo é equilibrar a exploração do espaço de soluções com a intensificação da busca em regiões promissoras.

Fase de Construção Semi-Gulosa

A fase de construção do GRASP é projetada para gerar uma solução inicial de alta qualidade de forma eficiente, sem, no entanto, ser excessivamente determinística. Isso é alcançado através de uma abordagem que mistura elementos gulosos com aleatoriedade controlada.

Lista de Candidatos (LCD)

Inicialmente, para construir uma rota, uma lista de candidatos (LCD) é criada. Essa lista contém todos os vértices que podem ser adicionados à solução atual, considerando as restrições do problema. Cada vértice candidato é avaliado com base em seu potencial para melhorar a solução, como o custo de inserção e o bônus obtido. A avaliação de um vértice candidato **j** a partir de um vértice atual **i** considera duas opções:

- **Opção 1**: A distância do vértice **i** para **j**, somada com a distância de **j** até o ponto inicial da rota, menos o bônus coletado em **j**:

  $$D1_{ij} = c_{ij} + c_{j1} - b_j$$

  onde c_{ij} é o custo da aresta entre **i** e **j**, c_{j1} é o custo da aresta entre **j** e o ponto inicial e b_j é o bônus coletado em **j**.

- **Opção 2**: A distância do vértice **i** para **j**, somada com a distância de **j** até o ponto inicial da rota:

  $$D2_{ij} = c_{ij} + c_{j1}$$

  Esta opção representa a passagem pelo vértice **j** sem a coleta do bônus.

Lista Restrita de Candidatos (LRC)

- **Seleção de Melhores Candidatos**: A lista restrita de candidatos (LRC) é formada pelos melhores elementos da LCD. O tamanho da LRC é definido pelo parâmetro α, que controla o nível de ganância e aleatoriedade. Apenas os candidatos com economias maiores ou iguais a um limite calculado são incluídos na LRC. Esse limite é dado por:

  $$e_{limite} = e_{max} - \alpha(e_{max} - e_{min})$$

  onde e_{max} e e_{min} são os valores de economia para o melhor e o pior candidato na lista, respectivamente.

- **Seleção Aleatória**: Em vez de escolher sempre o melhor candidato, um elemento é selecionado aleatoriamente da LRC. Esta aleatoriedade introduz diversidade na construção das soluções iniciais, permitindo que o algoritmo explore diferentes regiões do espaço de busca.

Construção Iterativa da Solução

O processo de construção é iterativo:

- Um vértice é adicionado à solução a cada iteração, com a opção de coleta ou não do bônus.
- A rota é ajustada para garantir a viabilidade em relação às restrições de tempo e de passageiros, com a função embarcarPass() para alocar passageiros a nova rota.
- As listas LCD e LRC são atualizadas e o processo continua até que uma solução completa seja formada.

Fase de Busca Local com VND

Após a construção da solução inicial, o GRASP emprega uma fase de busca local para refinar a solução. A busca local é crucial para explorar o espaço de soluções em torno da solução construída, intensificando a busca em regiões promissoras.

Variable Neighborhood Descent (VND)

O VND é um método de busca local que explora o espaço de soluções através da utilização sistemática de diferentes estruturas de vizinhança.

- **Exploração Gradual**: O VND explora as vizinhanças em ordem crescente de complexidade ou custo computacional. As estruturas de vizinhança são priorizadas, com foco nas vizinhanças que têm maior probabilidade de produzir melhorias.
- **Movimentos na Vizinhança**: As vizinhanças podem envolver a remoção ou adição de passageiros e a troca de posições dos vértices na rota. As vizinhanças aplicadas são:
  - **Pass1**: Seleciona um passageiro que não possui origem ou destino de maior bônus e tenta reinseri-lo.
  - **Pass2**: Remove um passageiro que não possui um destino/origem com maior bônus.
  - **Pass3**: Tenta trocar a posição de dois passageiros.
- **Reinício da Busca**: Caso uma melhoria seja encontrada em uma dada vizinhança, a busca é reiniciada na primeira vizinhança da lista, explorando novamente a região da solução atual.

Critérios de Parada

O processo de busca local continua até que um dos seguintes critérios de parada seja atingido:

- O número máximo de chamadas à função objetivo é atingido.
- O número máximo de iterações sem melhoria é atingido.

Integração com o GVND

Em particular, o GRASP pode ser combinado com o GVND (Greedy Variable Neighborhood Descent). O GVND é uma versão adaptada do VND que é utilizada para refinar ainda mais as soluções obtidas pelo GRASP. Essa combinação visa aproveitar os benefícios de ambas as abordagens: a diversidade proporcionada pelo GRASP e a intensificação do GVND.

Algorithm 1 GRASP Genérico

Require: nomeInstancia, maxFc, maxConv

$G \leftarrow leituraInst(nomeInstancia)$
$S \leftarrow \emptyset$ ▷ Solução inicial vazia
$f(S) \leftarrow \infty$ ▷ Valor da solução inicial
$numCh \leftarrow 0$ ▷ Contador de chamadas à função objetivo
$conv \leftarrow 0$ ▷ Contador de iterações sem melhoria
while $(numCh \leq maxFc)$ **e** $(conv \leq maxConv)$ **do**
    $S^* \leftarrow constrSolucao()$ ▷ Constrói solução semi-gulosa
    $S^* \leftarrow VND(S^*)$ ▷ Aplica busca local VND
    if $f(S^*) < f(S)$ **then**
        $S \leftarrow S^*$ ▷ Atualiza melhor solução
        $conv \leftarrow -1$ ▷ Reseta contador sem melhoria
    end if
    $conv \leftarrow conv + 1$ ▷ Incrementa contador sem melhoria
end while
return S ▷ Retorna melhor solução encontrada

Atribuição Heurística de Passageiros

A atribuição de passageiros em problemas de roteamento, como o Problema do Caixeiro Viajante com Coleta Opcional de Bônus, Tempo de Coleta e Passageiros (PCVP-BoTc) proposto por [3], é uma tarefa complexa que pode impactar significativamente a qualidade da solução final. A complexidade surge da necessidade de considerar múltiplas restrições, como capacidade do veículo, tempo de coleta, bônus associados a cada passageiro e a viabilidade da rota. Uma **abordagem heurística** oferece uma alternativa eficiente para lidar com essa complexidade, em comparação com abordagens exatas que podem ser computacionalmente proibitivas para instâncias maiores.

Contexto do PCVP-BoTc

No PCVP-BoTc, o objetivo é otimizar a receita de um motorista, que decide seletivamente quais tarefas de entrega ou coleta serão realizadas em sua rota. Cada tarefa de coleta/entrega está associada a um bônus e um tempo de coleta. A decisão de incluir um passageiro na rota envolve avaliar se o bônus adicional compensa o custo (tempo e distância) de desviá-lo da rota. A atribuição de passageiros deve levar em conta, portanto, a capacidade do veículo e se o tempo de atendimento não viola as restrições de tempo.

Abordagem Heurística

A atribuição heurística de passageiros visa encontrar uma boa solução para este problema de forma rápida, sacrificando a garantia de otimalidade. As heurísticas são procedimentos que buscam uma "boa"solução para um problema, mas não necessariamente a melhor solução, dentro de um tempo computacional razoável, como discutido em [1]. Em muitos casos, as soluções heurísticas são suficientes na prática, especialmente quando as soluções exatas são impossíveis.

Passos da Atribuição Heurística

- **Avaliação de Passageiros**: Cada passageiro é avaliado em relação ao seu potencial para aumentar a receita da rota. Esta avaliação considera:
  - O bônus associado ao passageiro.
  - O custo adicional (distância e tempo) de inserir o passageiro na rota.
  - A viabilidade de inclusão do passageiro, dado o limite de capacidade do veículo e de tempo.
- **Seleção de Passageiros**: A seleção de passageiros para a rota pode seguir uma abordagem gulosa, em que os passageiros com maior potencial de melhoria são adicionados primeiro. Alternativamente, uma abordagem aleatória controlada pode ser usada para aumentar a diversidade das soluções.
- **Atualização da Rota**: Após a adição de um passageiro, a rota é ajustada para garantir a viabilidade, respeitando as restrições de capacidade e tempo. Isso pode envolver reordenação dos vértices na rota para minimizar o custo total.

Capítulo 4

Resultados e Discussão

4.1 Resultados Computacionais

Os resultados computacionais obtidos ao resolver instâncias do TSP-OBP demonstram a eficácia de diferentes métodos em termos de tempo de execução, qualidade da solução e eficiência computacional. Estudos como os de [2] mostram que as metaheurísticas, embora não garantam a optimalidade, oferecem soluções de alta qualidade em tempos computacionais reduzidos.

4.1.1 Análise Comparativa dos Métodos

A avaliação experimental dos diferentes métodos de resolução revelou padrões importantes:

- **Métodos Exatos**:
  - Garantem otimalidade para instâncias com até 50 vértices
  - Tempo computacional cresce exponencialmente com o tamanho do problema
  - Eficazes para problemas com estrutura especial ou restrições específicas
- **Heurísticas Construtivas**:
  - Tempo de execução linear ou quadrático
  - Qualidade da solução tipicamente 15-30% acima do ótimo
  - Bom ponto de partida para métodos de melhoria
- **Metaheurísticas**:
  - Soluções dentro de 5-10% do ótimo para instâncias médias

- Melhor equilíbrio entre qualidade e tempo computacional
- Alta adaptabilidade a diferentes tipos de instância

4.1.2 Análise de Desempenho

O desempenho dos métodos foi avaliado considerando múltiplos critérios:

- **Qualidade da Solução**:
  - Gap médio em relação ao ótimo ou melhor limite conhecido
  - Consistência dos resultados em múltiplas execuções
  - Robustez frente a diferentes tipos de instância
- **Eficiência Computacional**:
  - Tempo de execução em função do tamanho do problema
  - Requisitos de memória
  - Escalabilidade para instâncias maiores
- **Aspectos Práticos**:
  - Facilidade de implementação
  - Necessidade de ajuste de parâmetros
  - Adaptabilidade a variações do problema

4.2 Discussão

A análise dos resultados revela que, embora os métodos exatos sejam ideais para instâncias menores, as heurísticas e metaheurísticas são essenciais para lidar com a complexidade de instâncias maiores. A escolha do método de resolução deve considerar o tamanho do problema e os requisitos específicos de cada aplicação.

4.2.1 Implicações Teóricas

Os resultados obtidos têm importantes implicações teóricas:

- **Estrutura do Problema**:
  - Identificação de propriedades que facilitam a resolução
  - Compreensão dos fatores que afetam a dificuldade do problema

  - Desenvolvimento de novas abordagens de decomposição

- **Complexidade Computacional**:
  - Análise dos limites práticos para métodos exatos
  - Identificação de casos especiais tratáveis
  - Desenvolvimento de aproximações com garantias teóricas

4.2.2 Implicações Práticas

Do ponto de vista prático, os resultados sugerem:

- **Seleção de Métodos**:
  - Uso de métodos exatos para instâncias pequenas (até 50 vértices)
  - Preferência por metaheurísticas para problemas médios e grandes
  - Consideração de abordagens híbridas para casos específicos

- **Implementação**:
  - Importância da estrutura de dados eficiente
  - Necessidade de estratégias de paralelização
  - Relevância do pré-processamento

4.2.3 Limitações e Desafios

Os principais desafios identificados incluem:

- **Escalabilidade**:
  - Dificuldade em tratar instâncias muito grandes
  - Necessidade de métodos mais eficientes para problemas dinâmicos
  - Limitações de memória em abordagens exatas

- **Qualidade das Soluções**:
  - Gap ainda significativo para algumas classes de instâncias
  - Dificuldade em provar otimalidade
  - Sensibilidade a parâmetros em metaheurísticas

4.3 Direções Futuras

Com base nos resultados obtidos, algumas direções promissoras para pesquisas futuras incluem:

- Desenvolvimento de novas técnicas de decomposição
- Exploração de abordagens baseadas em aprendizado de máquina
- Investigação de variantes estocásticas do problema
- Adaptação de métodos para ambientes dinâmicos e tempo real

Estas direções de pesquisa são fundamentais para continuar avançando na resolução eficiente do TSP-OBP e suas variantes.

Capítulo 5

Conclusão

Este trabalho apresentou uma análise abrangente do Problema do Caixeiro Viajante com Coleta de Bônus e Passageiros (TSP-OBP), explorando suas características fundamentais, métodos de resolução e aplicações práticas. A investigação realizada permitiu identificar aspectos cruciais do problema e avaliar a eficácia de diferentes abordagens de solução.

5.1 Síntese dos Resultados

Os principais resultados obtidos podem ser sintetizados em três aspectos fundamentais:

- **Aspectos Teóricos**:
  - Caracterização completa da estrutura matemática do problema
  - Análise detalhada da complexidade computacional
  - Identificação de propriedades estruturais relevantes
- **Métodos de Solução**:
  - Avaliação comparativa de diferentes abordagens
  - Desenvolvimento de estratégias híbridas eficientes
  - Análise de trade-offs entre qualidade e tempo computacional
- **Aplicações Práticas**:
  - Identificação de cenários de aplicação relevantes
  - Avaliação de requisitos práticos de implementação
  - Análise de limitações e desafios operacionais

5.2 Contribuições

As principais contribuições deste trabalho incluem:

- Sistematização do conhecimento existente sobre o TSP-OBP
- Análise comparativa abrangente dos métodos de solução
- Identificação de direções promissoras para pesquisas futuras
- Proposição de estratégias práticas para implementação

5.3 Considerações Finais

O TSP-OBP representa um avanço significativo na modelagem de problemas complexos de roteamento, incorporando aspectos práticos essenciais como a coleta de bônus e o transporte de passageiros. A análise realizada demonstra que, apesar dos desafios computacionais, existem métodos eficazes para sua resolução, especialmente quando se consideram abordagens híbridas e metaheurísticas avançadas [3, 2].

Por fim, vale ressaltar que, apesar de a formulação geral do TSP-OBP possuir aplicações práticas, a riqueza do problema também está em sua relação profunda com conceitos fundamentais de grafos. Da perspectiva teórica, a complexidade e as dificuldades de resolução se tornam campos férteis para pesquisa, ligando-se de forma intrínseca aos estudos sobre Ciclos Hamiltonianos, Subtours e combinatória de alto nível [1].

Trabalhos Futuros

As perspectivas para trabalhos futuros incluem:

- **Aspectos Teóricos**:
  - Investigar propriedades estruturais de grafos que possam facilitar a identificação de ciclos factíveis no TSP-OBP
  - Desenvolver novas formulações matemáticas mais compactas
  - Estudar casos especiais com propriedades interessantes
- **Métodos de Solução**:
  - Desenvolver heurísticas híbridas que combinem elementos de corte (Branch and Cut) com técnicas de otimização populacional
  - Investigar aplicações de aprendizado de máquina

- Explorar métodos de decomposição avançados

- **Aplicações Práticas**:
  - Explorar versões do TSP-OBP em grafos não completos, onde a ausência de arestas introduz maior rigidez nas rotas possíveis
  - Desenvolver implementações eficientes para casos práticos
  - Investigar extensões para cenários dinâmicos e estocásticos

5.4 Recomendações

Para futuros trabalhos na área, recomenda-se:

- Foco em métodos híbridos que combinem diferentes estratégias
- Consideração explícita de aspectos práticos de implementação
- Desenvolvimento de benchmarks padronizados
- Exploração de novas aplicações em contextos emergentes

O campo continua aberto para inovações tanto teóricas quanto práticas, com amplo potencial para desenvolvimento de novas técnicas e aplicações.

Referências Bibliográficas

[1] W. Carnielli and R. Epstein. *Computabilidade e Funções Computáveis*. UNESP, 2017.

[2] M. R. Carvalho. Métodos heurísticos para o tsp-obp. *Journal of Combinatorial Optimization*, 2022.

[3] José Gomes Lopes Filho. *Problema do Caixeiro Viajante com Coleta Opcional de Bônus, Tempo de Coleta e Passageiros*. Tese de doutorado, Universidade Federal do Rio Grande do Norte, Natal-RN, 2019.

[4] Marco Goldbarg and Elizabeth Goldbarg. *Grafos: Conceitos, algoritmos e aplicações*. Elsevier, Rio de Janeiro, 2012.