

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Luciano Ermelindo de Almeida

Problema de Orientação Clusterizado com Subgrupos

Belo Horizonte
2025

Luciano Ermelindo de Almeida

Problema de Orientação Clusterizado com Subgrupos

Versão Final

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Douglas Guimarães Macharet

Belo Horizonte
2025

Almeida, Luciano Ermelindo de

A447p Problema de orientação clusterizado com subgrupos
[recurso eletrônico] /Luciano Ermelindo de Almeida –
2025.

1 recurso online (112 f. il., color.) : pdf.

Orientador: Douglas Guimarães Macharet.

Dissertação (Mestrado) - Universidade Federal de
Minas Gerais, Instituto de Ciências Exatas, Departamento
de Ciência da Computação.

Referências: f. 91-96.

1. Computação – Teses. 2. Robótica- Teses. 3. Otimização
combinatória. – Teses. 4. Problema de otimização clusterizado
– Teses. 5. Problema de roteamento de veículos - Teses.
6. Programação linear – Teses. I. Macharet, Douglas Guimarães.
II. Universidade Federal de Minas Gerais, Instituto de Ciências
Exatas, Departamento de Ciência da Computação. III. Título.

CDU 519.6*82.9(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Problema de Orientação Clusterizado com Subgrupos

LUCIANO ERMELINDO DE ALMEIDA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

A handwritten signature in black ink, reading "Douglas Guimarães Macharet".

PROF. DOUGLAS GUIMARÃES MACHARET - Orientador
Departamento de Ciência da Computação - UFMG

A handwritten signature in blue ink, consisting of stylized initials.

PROF. CRISTIANO ARBEX VALLE
Departamento de Ciência da Computação - UFMG

A handwritten signature in black ink, reading "Thiago F. Noronha".

PROF. THIAGO FERREIRA DE NORONHA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 21 de novembro de 2025.

*Dedico este trabalho Àquele de quem provêm todas as coisas,
e a quem tudo devolvo com gratidão.*

Agradecimentos

Agradeço, primeiramente, a Deus, pelas misericórdias que se renovam dia após dia, mesmo no meio das tempestades. Sua graça foi meu sustento em todos os momentos desta jornada.

À minha mãe, Delma, à minha irmã, Nilma, e à minha sobrinha, Valentina — minha base, meu alicerce, minha família. Obrigado por estarem sempre presentes, com amor e apoio incondicional. Família é a base desta vida, e sou profundamente grato por vocês. Aos meus primos, agradeço pelos momentos de leveza e descontração que tornaram essa caminhada mais leve.

Aos colegas do VeRLab, minha gratidão pela parceria ao longo do percurso. Em especial, Maurício, Aline e Alysson — obrigado pelos cafés, pelas conversas e pela torcida nos momentos decisivos.

Ao meu orientador, professor Douglas, agradeço pela paciência, pelos conselhos e pelo direcionamento ao longo do trabalho. Continue sendo este bom orientador, Douglas, sua orientação fez a diferença.

À UFMG, ao DCC e ao VeRLab, deixo meu reconhecimento pela formação acadêmica de excelência que recebi. Agradeço também à banca avaliadora, pela leitura atenta, pelas contribuições valiosas e pela generosidade no processo de avaliação.

A todos que, de alguma forma, fizeram parte desta conquista, o meu sincero muito obrigado.

*“Pois os nossos sofrimentos leves e momentâneos estão produzindo para nós
uma glória eterna que pesa mais do que todos eles.”*
(2 Coríntios 4:17)

Resumo

O planejamento de rotas é fundamental para otimizar processos logísticos e aumentar a eficiência no deslocamento de pessoas, mercadorias e robôs móveis. Em cenários onde não é possível visitar todos os destinos, torna-se necessário priorizar atendimentos com base na relação custo-benefício. O Orienteering Problem (OP) é um modelo clássico que busca maximizar a recompensa obtida dentro de um limite de tempo ou distância. A evolução das aplicações práticas desse problema deu origem a variantes como o Clustered Orienteering Problem (COP), no qual todos os pontos de um agrupamento devem ser visitados para se obter a recompensa, e o Set Orienteering Problem (SOP), que permite visitar no máximo um ponto por grupo. No entanto, essas formulações não representam adequadamente cenários que envolvem alternativas de visita com diferentes níveis de esforço e recompensa dentro de uma mesma região.

Para lidar com essa limitação, esta dissertação propõe o Clustered Orienteering Problem with Subgroups (COPS), uma nova variante que generaliza o COP e o SOP ao introduzir subgrupos hierárquicos dentro dos grupos (*clusters*). Nesse modelo, cada subgrupo oferece uma recompensa condicional à visita de todos os seus pontos, sendo permitido selecionar no máximo um subgrupo por grupo. O objetivo é determinar a rota que maximize a recompensa total, respeitando as restrições de orçamento.

A proposta tem aplicações em logística, inspeção, monitoramento ambiental e, especialmente, no planejamento de rotas para robôs móveis. Para resolver o COPS, foram desenvolvidas duas abordagens: uma formulação exata em Programação Linear Inteira (ILP), adequada para instâncias menores, e uma heurística baseada em Busca Tabu, eficiente para instâncias de maior escala.

Os resultados experimentais demonstram que a formulação ILP é viável apenas para instâncias pequenas e médias, reforçando a necessidade de métodos aproximativos em cenários mais complexos. O COPS-TABU apresentou desempenho competitivo em termos de qualidade das recompensas obtidas, superando ou igualando benchmarks adaptados do COP e do SOP. Além disso, os experimentos confirmam a aplicabilidade do modelo em cenários com clusterização complexa, rotas não circulares, múltiplos destinos e ambientes tridimensionais.

Palavras-chave: planejamento de rota; problema de orientação; problema de orientação agrupado; problema de orientação por conjuntos; problema de orientação agrupado com subgrupos; busca tabu; programação linear inteira.

Abstract

Route planning plays a crucial role in optimizing logistics processes and enhancing the efficiency of the movement of people, goods, and mobile robots. In scenarios where visiting all destinations is not feasible, it becomes necessary to prioritize visits based on a cost–benefit trade-off. The Orienteering Problem (OP) is a classical optimization model that seeks to maximize the total collected reward within a given time or distance constraint. The evolution of practical applications has led to variants such as the Clustered Orienteering Problem (COP), in which all points within a cluster must be visited to obtain its reward, and the Set Orienteering Problem (SOP), which allows visiting at most one point per group. However, these formulations fail to adequately represent scenarios involving alternative options with varying levels of effort and reward within the same region.

To address this limitation, this dissertation introduces the Clustered Orienteering Problem with Subgroups (COPS), a novel variant that generalizes both COP and SOP by incorporating hierarchical subgroups within clusters. In this formulation, each subgroup provides a conditional reward that is granted upon visiting all its points, while at most one subgroup can be selected per cluster. The objective is to determine the route that maximizes the total collected reward subject to a budget constraint.

The proposed model has potential applications in logistics, inspection, environmental monitoring, and particularly in route planning for mobile robots. To solve COPS, two complementary approaches were developed: an exact Integer Linear Programming (ILP) formulation suitable for small instances, and a Tabu Search-based heuristic designed for larger-scale problems.

Experimental results show that the ILP approach is computationally feasible only for small and medium-sized instances, highlighting the need for heuristic or metaheuristic techniques for larger problems. The COPS-TABU method demonstrated competitive performance in terms of reward quality compared to adapted COP and SOP benchmarks, while also validating the applicability of the proposed model to complex clustering configurations, non-circular routes, multi-destination scenarios, and three-dimensional environments.

Keywords: route planning; orienteering problem; clustered orienteering problem; set orienteering problem; clustered orienteering problem with subgroups; tabu search; integer linear programming.

Lista de Figuras

1.1	Exemplo de um conjunto de pontos de vista utilizados para reconstrução 3D [Kolev et al., 2009].	21
1.2	Resultado do planejamento de trajetória para a reconstrução 3D de diferentes prédios, conforme [N. Smith et al., 2018].	22
1.3	Exemplo de aplicação do COPS no contexto de HRI [Silva, 2025].	23
1.4	Exemplo de sete clusters com subgrupos distintos. A recompensa associada aos subgrupos é representada por suas cores. O cluster com um único vértice é o ponto inicial/final, e a linha azul é uma rota possível encontrada pelo algoritmo desenvolvido neste trabalho.	24
2.1	Exemplo de seis clusters com subgrupos distintos. A recompensa associada aos subgrupos é representada por suas cores. O cluster 1 é o ponto inicial/final, e a linha azul é uma rota possível encontrada pelo algoritmo desenvolvido neste trabalho.	28
3.1	Solução para um <i>Traveling Salesman Problem</i> (Problema do Caixeiro Viajante, TSP), cujo custo total é minimizado.	32
3.2	Esportista em uma competição de orientação por [North East Orienteering Association, 2025].	37
3.3	Um exemplo de como os clientes são organizados em grupos no <i>Clustered Orienteering Problem</i> (COP). O lucro é coletado somente se todos os pontos de interesse (PoIs) forem visitados.	42
3.4	<i>Set Orienteering Problem</i> (SOP). No máximo um ponto de interesse de cada cluster deve ser visitado para se obter a recompensa associada àquele grupo, por [Pěnička et al., 2019].	43
3.5	Grafo formado pela sequência de conjuntos no <i>Variable Neighborhood Search for the Set Orienteering Problem</i> (VNS-SOP) [Pěnička et al., 2019].	45
3.6	Exemplo de teste rápido para limite de orçamento por [Pěnička et al., 2019].	46
3.7	<i>Teams Orienteering Problem</i> (TOP) - Neste Exemplo dois veículos realizam duas rotas alcançando a recompensa associada a cada um dos vértices visitados por [Bouchakhchoukha e Akeb, 2014].	47
3.8	Comparação entre o OP e o DOP. À esquerda planejamento de caminho sem restrição de curvatura, à direita caminho com restrição de curvatura por [Pěnička et al., 2017].	47

3.9	<i>Minimal Exposure Dubins Orienteering Problem (MEDOP)</i> - Regiões avermelhadas devem ser evitadas pelo veículo por [Macharet et al., 2021].	48
3.10	Exemplo do COP como caso particular do COPS. Neste exemplo, cada cluster é formado por somente um subgrupo contendo todos os vértices do cluster. . .	54
3.11	Exemplo do SOP como caso particular do COPS. Neste exemplo, todos os subgrupos são formados por um único vértice, e a rota da solução seleciona no máximo um subgrupo por cluster.	55
5.1	Exemplo ilustrativo de uma instância do problema COPS.	66
5.2	SOP formato - pontos pretos representam os vértices, áreas coloridas representam os subgrupos (a cor depende da recompensa correspondente) e linha cinza representa o cluster. Neste caso, cada subgrupo contém apenas um vértice. Como o COPS escolhe apenas um subgrupo em cada cluster, alcançamos um resultado idêntico ao SOP. A linha em azul é o caminho encontrado pelo COPS-TABU.	69
5.3	COP formato - pontos pretos representam os vértices, áreas coloridas representam os subgrupos (a cor depende da recompensa correspondente) e linha cinza representa o cluster. Neste caso, cada cluster contém apenas um subgrupo com vários vértices. Já que o COPS visita todos os vértices dentro de um subgrupo, conseguimos assim um resultado idêntico ao COP. A linha em azul é o caminho encontrado pelo COPS-TABU.	70
5.4	Exemplos de aplicação do COPS com dois clusters clientes e 4 subgrupos clientes, dois subgrupos para cada cluster. O algoritmo COPS-TABU escolhe diferentes caminhos para diferentes valores de $T_{\max}$, sempre escolhendo caminhos que alcançam maior recompensa possível. O subgrupo mais externo possui recompensa de 4, enquanto o subgrupo mais interno possui recompensa igual a 3.	74
5.5	Exemplos de aplicação do COPS aumentando progressivamente o número de clusters mantendo o mesmo orçamento $T_{\max} = 150$. Cada cluster cliente possui 3 subgrupos, com recompensas de 60, 70 e 100.	76
5.6	Exemplos de aplicação do COPS Não Circular. Cada cluster cliente possui 2 subgrupos, possuindo as recompensas de 6 e 10. E, o cluster final possui 4 subgrupos, o algoritmo escolhe o endpoint de acordo com o orçamento da viagem.	79
5.7	Exemplos de aplicação do COPS em 3 dimensões.	80
5.8	Análise de desempenho da ILP com o aumento do número de vértices em diferentes configurações de clusters e subgrupos.	82
5.9	Análise de desempenho do COPS-TABU com o aumento do número de vértices em diferentes configurações de clusters e subgrupos.	83

5.10	Análise do comportamento do tempo de execução do algoritmo COPS-TABU ao aumentar o número de vértices para diferentes tamanhos de subgrupos e clusters.	85
B.1	Divisão de regiões (clusters) a serem atendidas no cenário 1. Em cada região podemos ter vértices prioritários, secundários e terciários.	103
B.2	Rotas para diversos orçamentos no cenário 1.	104
B.3	Outras Rotas para o cenário 1.	105
B.4	Rotas para diversos orçamentos no cenário 2.	106
B.5	Rotas para diversos orçamentos no cenário 3.	108

Lista de Tabelas

3.1	Relações estruturais entre o COPS e suas variantes particulares.	55
5.1	Comparação dos resultados do SOP original MASOP [Archetti et al., 2018] e VNS-SOP [Pěnička et al., 2019], com as nossa formulações em ILP e COPS-TABU. A análise considera a Recompensa (R) e Tempo de execução (T) em segundos.	68
5.2	Comparação dos resultados do COP original COP-TABU [Angelelli et al., 2014], com as nossa formulações em ILP e COPS-TABU. A análise considera a Recompensa (R) e Tempo de execução (T) em segundos.	71
5.3	Avaliação do ILP e do COPS-TABU variando o orçamento do robô, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.	74
5.4	Avaliação do ILP e do COPS-TABU variando a quantidade de clusters e mantendo o orçamento $T_{\max}$ constante, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.	77
5.5	Avaliação do ILP e do COPS-TABU variando o orçamento do robô em um cenário onde o cluster final contém diferentes endpoints, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.	78
5.6	Avaliação do ILP e do COPS-TABU em um cenário com 3 dimensões, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.	81
B.1	Resultados obtidos para diferentes valores de orçamento ($T_{\max}$) no cenário proposto. A tabela apresenta a recompensa total obtida (R), o comprimento da rota (L) e observações sobre quais subgrupos foram atendidos em cada caso.	103

List of Algorithms

1	Busca Tabu para o COPS	60
---	----------------------------------	----

Lista de Símbolos

c	Cluster
$\mathcal{C}$	Conjunto de Clusters
e	Aresta
$\mathcal{E}$	Conjunto de arestas
$\mathcal{G}$	Grafo não direcionado
L	Distância de viagem do robô
p	Nó (vértice) da rota
$\mathcal{P}$	Rota Final
r	Recompensa por visitar o subgrupo
R	Recompensa Total alcançada
s	Subgrupo
$\mathcal{S}$	Conjunto de Subgrupos
t	Custo de viajar pela aresta
T	Tempo de execução
$T_{\max}$	Orçamento de viagem do robô
v	Vértice
$\mathcal{V}$	Conjunto de Vértices

Lista de Acrônimos

APORA *Adaptable Probabilistic Object Reconstruction Algorithm* (Algoritmo Adaptável de Reconstrução Probabilística de Objetos)

BFS *Breadth-First Search* (busca em largura)

CCOP *Clustered Coverage Orienteering Problem*

CluTOP *Clustered Team Orienteering Problem*

COP *Clustered Orienteering Problem*

COPCDP *Categorized Orienteering Problem with Count-Dependent Profits*

COPS *Clustered Orienteering Problem with Subgroups*

COPTW *Cooperative OP with Time Windows*

DOP *Dubins Orienteering Problem* (Problema de Orientação Dubins)

ILP *Integer Linear Programming* (Programação Linear Inteira)

KP *Knapsack Problem* (Problema da Mochila)

MASOP *Matheuristic for the SOP*

MCTOPMTW *Multi-Constraint TOP with Multiple Time Windows*

MEDOP *Minimal Exposure Dubins Orienteering Problem*

MILP *Mixed Integer Linear Programming* (Programação Linear Inteira Mista)

MPOPTW *Multi-Path OP with Time Windows*

MuPOPTW *Multi-Period OP with Multiple Time Windows*

OP *Orienteering Problem*

OPN *Orienteering Problem with Neighborhoods*

OPTW *Orienteering Problem with Time Windows*

PoI ponto de interesse

SfM *Structure from motion* (Estrutura de movimento)

SOP *Set Orienteering Problem*

SOPTW *Stochastic OP with Time Windows*

STOPTW *Set Team Orienteering Problem with Time Windows*

SVCIRP *Problema de Roteamento de Inventário Cíclico com Veículo Único*

TDOP *Time Dependent Orienteering Problem*

TDTOPTW *Time Dependent Team Orienteering Problem with Time Windows*

TOARP *Team Orienteering Arc Routing Problem*

TOP *Teams Orienteering Problem*

TSP *Traveling Salesman Problem* (Problema do Caixeiro Viajante)

VNS *Variable Neighborhood Search* (Busca em Vizinhança Variável)

VNS-SOP *Variable Neighborhood Search for the Set Orienteering Problem*

VRP *Vehicle Routing Problem* (Problema de Roteamento de Veículo)

Sumário

1	Introdução	19
1.1	Contextualização	19
1.2	Motivação	20
1.3	Problema	23
1.4	Contribuições da Pesquisa	25
1.5	Estrutura da Dissertação	26
2	Formalização do Problema	27
2.1	Representação Matemática e Notação	29
3	Trabalhos Relacionados	32
3.1	Problema do Caixeiro Viajante (TSP)	32
3.2	Problema da Mochila	35
3.3	<i>Orienteering Problem</i> (OP)	37
3.4	Principais variantes do <i>Orienteering Problem</i>	41
3.5	Aplicações práticas do OP e de suas variantes	48
3.6	Contextualização do trabalho	52
4	Propostas de Solução	56
4.1	Formulação em Programação Linear Inteira	56
4.2	Heurística baseada em Busca Tabu	59
5	Resultados	65
5.1	Exemplo Ilustrativo do COPS	66
5.2	Comparação com Cenários Clássicos do COP e SOP	67
5.3	Casos de Estudo em Cenários Próprios	72
5.4	Análise de performance ILP	81
5.5	Análise de performance da Heurística	83
6	Conclusões e Trabalhos Futuros	86
6.1	Análise Crítica das Limitações do COPS	87
6.2	Trabalhos Futuros	90
	Bibliografia	91

Apêndice A Métodos automáticos para definição de clusters e subgrupos utilizando Linguagem Natural e LLM	97
Apêndice B Exemplos de Aplicação do Modelo COPS	101

Capítulo 1

Introdução

Atualmente, o planejamento de rotas é uma ferramenta indispensável para otimizar processos logísticos e aumentar a eficiência no transporte de pessoas, mercadorias e, cada vez mais, no deslocamento de robôs móveis em ambientes industriais e públicos. Diante da crescente demanda por agilidade e da pressão por redução de custos, definir rotas mais eficientes permite economizar tempo, combustível, energia e recursos operacionais, impactando diretamente na competitividade das organizações.

Mais do que traçar caminhos curtos, um planejamento eficaz deve considerar a priorização de atendimentos com base no custo-benefício de cada deslocamento. Isso significa identificar os grupos ou regiões com maior demanda, urgência ou retorno financeiro, garantindo que os recursos sejam alocados de forma estratégica, com o maior impacto possível para o negócio.

Assim, o planejamento de rotas deixa de ser uma tarefa meramente operacional e assume um papel estratégico. Ele orienta decisões sobre onde, quando e como deslocar recursos, promovendo o uso inteligente da frota e maximizando os resultados com o menor custo possível.

Este trabalho propõe um paradigma de planejamento de rotas que considera as relações de agrupamento e exclusividade entre diferentes pontos de interesse (PoIs), bem como as recompensas e prioridades de atendimento associadas a cada um. O modelo leva em conta um limite de gasto de energia durante o deslocamento, buscando maximizar o ganho total obtido ao final do percurso.

1.1 Contextualização

Vehicle Routing Problems (Problemas de Roteamento de Veículos, VRPs) desempenham um papel central em diversas áreas da Ciência da Computação, com aplicações relevantes em logística, transporte, robótica e sistemas autônomos. Dentre esses problemas, destaca-se o *Traveling Salesman Problem* (Problema do Caixeiro Viajante, TSP),

um dos problemas clássicos de otimização combinatória, cujo objetivo é determinar a menor rota possível que visita um conjunto de cidades exatamente uma vez, retornando ao ponto de origem. Apesar de ser um problema NP-difícil, o TSP possui ampla aplicabilidade em cenários reais e tem servido de base para o desenvolvimento de diversas variantes e extensões.

Uma generalização importante do TSP é o *Orienteering Problem* (OP) [Golden et al., 1987], que incorpora restrições de orçamento (como tempo ou distância máxima permitida) e associa recompensas às cidades visitadas. O objetivo do OP é determinar uma rota que maximize a recompensa total obtida, respeitando o limite orçamentário. Essa formulação é especialmente útil em contextos onde não é possível visitar todos os destinos disponíveis, exigindo uma seleção estratégica dos pontos mais vantajosos.

Com a evolução das aplicações práticas e a necessidade de representar estruturas mais complexas, surgiram variantes do OP que consideram a organização dos PoIs em agrupamentos. Neste trabalho, utilizamos o termo *cluster* para nos referirmos a esses agrupamentos, adotando-o como termo técnico amplamente utilizado na literatura especializada, que preserva o significado exato no contexto do problema. Entre elas, destacam-se o *Clustered Orienteering Problem* (COP) [Angelelli et al., 2014], no qual existe uma relação de unidade entre os pontos de interesse de forma que a recompensa está associada ao grupo como um todo, e é obtida apenas se todas os PoIs do cluster forem visitadas, e o *Set Orienteering Problem* (SOP) [Archetti et al., 2018], onde a relação entre os PoIs do cluster é de exclusividade, de maneira que, para se obter a recompensa de um cluster no máximo um de seus PoIs deve ser visitado.

Neste contexto, insere-se o *Clustered Orienteering Problem with Subgroups* (COPS), uma nova variante proposta nesta dissertação. Essa formulação amplia as capacidades de modelagem das variantes anteriores ao introduzir subgrupos dentro de clusters, permitindo tanto a ideia de unidade quanto a de exclusividade entre os PoIs, dando assim maior flexibilidade na modelagem do problema de roteamento de veículos quando existem relações entre os pontos de interesse.

1.2 Motivação

Em diversos domínios aplicados — como logística, inspeção, monitoramento ambiental, reconstrução 3D e interação homem-robô — surge o desafio de planejar rotas que maximizem o ganho de informação, respeitando restrições de tempo, custo ou energia.

O problema clássico de roteirização OP, fornece uma base para esse tipo de otimização, mas assume que todos os pontos de interesse são igualmente acessíveis e inde-

pendentes. Essa simplificação raramente ocorre no mundo real, onde as tarefas tendem a se agrupar em regiões, contextos ou prioridades diferentes.

Por exemplo, no contexto da reconstrução 3D a partir de imagens, o processo de Structure from Motion (SfM) [Schönberger e Frahm, 2016] é utilizado para gerar modelos tridimensionais a partir de múltiplos pontos de vista de uma mesma cena ou objeto. Nesse processo, a captura de imagens de diferentes ângulos deve ser tratada como um conjunto único e correlacionado, no qual cada ponto de vista contribui parcialmente para a qualidade da reconstrução (Figura 1.1). No entanto, o ganho de informação só se torna significativo quando o conjunto completo de perspectivas é adquirido, uma vez que cada nova imagem ajuda a refinar a posição e a geometria dos pontos reconstruídos. Assim, surge a necessidade de modelar relações de dependência e complementaridade entre os pontos de interesse, de forma que o planejamento de rotas considere não apenas o valor individual de cada ponto, mas seu valor coletivo dentro de um grupo.

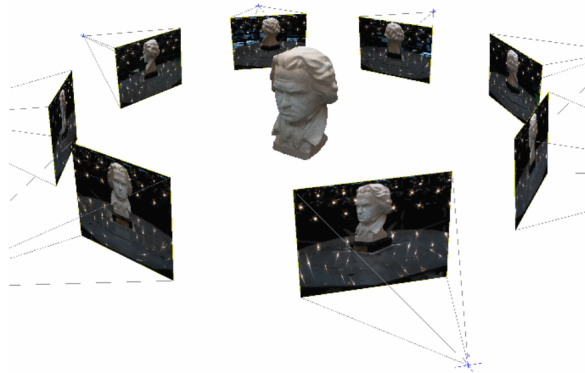


Figura 1.1: Exemplo de um conjunto de pontos de vista utilizados para reconstrução 3D [Kolev et al., 2009].

Além disso, podemos observar que a qualidade da reconstrução está diretamente relacionada ao conjunto de imagens adquiridas. Em outras palavras, conjuntos de imagens mais superficiais da região tendem a gerar reconstruções de menor qualidade, enquanto conjuntos mais abrangentes e detalhados resultam em reconstruções mais precisas. Por outro lado, ao considerar a trajetória de um drone, por exemplo, para capturar essas imagens, fica claro que, para obter um conjunto mais descritivo de pontos de vista, será necessário um maior gasto de energia durante o voo, para se conseguir capturar todos os detalhes e limitar os pontos com oclusões. Em contraste, um voo mais superficial, que cobre menos pontos de vista, demanda menos energia.

Assim, existem diferentes subgrupos de pontos de vista, que são mutuamente excludentes. Isso ocorre porque, ao planejar a rota, a escolha de um conjunto mais detalhado de pontos de vista para um determinado objeto elimina a necessidade de pontos de vista menos detalhados do mesmo objeto, e vice-versa, evitando redundâncias e otimizando a cobertura. Além disso, como cada subgrupo tem custos e benefícios distintos, a decisão sobre qual seguir depende da limitação energética do drone. Essa escolha é especialmente

relevante quando há múltiplos alvos de reconstrução em uma área, como ilustrado na Figura 1.2, especialmente quando alguns desses alvos tem maior prioridade em relação a outros. Portanto, a definição da rota de voo do drone, que deve cobrir os diferentes alvos com níveis variados de qualidade, torna-se uma tarefa essencial, dada a limitação de energia.

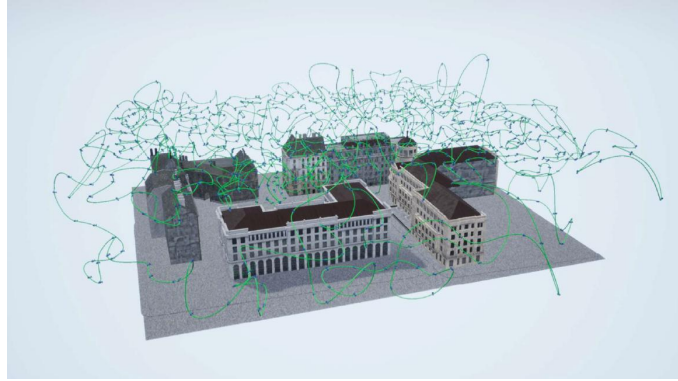


Figura 1.2: Resultado do planejamento de trajetória para a reconstrução 3D de diferentes prédios, conforme [N. Smith et al., 2018].

Essa mesma ideia pode ser aplicada à inspeção visual de construções a partir de imagens, e da mesma forma cada objeto pode ser inspecionado com maior ou menor qualidade e se deve escolher entre o melhor custo benefício entre visitar mais objetos ou visitar os subgrupos de maior qualidade/custo de cada alvo.

Outro contexto em que observamos conjuntos de pontos de interesse mutuamente exclusivos ocorre na interação humano-robô (HRI). Destacamos aqui o cenário no qual o robô deve visitar diferentes grupos de pessoas. Em cada grupo, estejam os indivíduos em pé ou sentados ao redor de uma mesa, existem diversas posições possíveis de serem alcançadas, enquanto outras seriam inadequadas ou constrangedoras, como, por exemplo, posicionar-se logo atrás de uma pessoa. Entre os pontos de visitação possíveis, alguns se mostram mais apropriados que outros, e, uma vez visitado um desses pontos, não há necessidade de visitar os demais do mesmo grupo (Figura 1.3). Assim, ao planejar a rota para múltiplos grupos de indivíduos, o trajeto ideal deve considerar tanto os aspectos de priorização das posições mais adequadas de abordagem quanto a maximização do número de grupos visitados, respeitando as limitações operacionais do robô.

Podemos ainda considerar o contexto de logística e entregas, em que os clientes podem ser agrupados de acordo com características específicas, seja pelo tipo de produto a ser entregue, perfil do cliente ou região de atendimento, e, adicionalmente, subdivididos em grupos prioritários. De modo análogo, no contexto de monitoramento marinho, é possível agrupar regiões potenciais de interesse no ambiente e subgrupá-las conforme as diferentes espécies que as habitam, cada uma com níveis distintos de importância e demandando abordagens específicas para a coleta de dados.

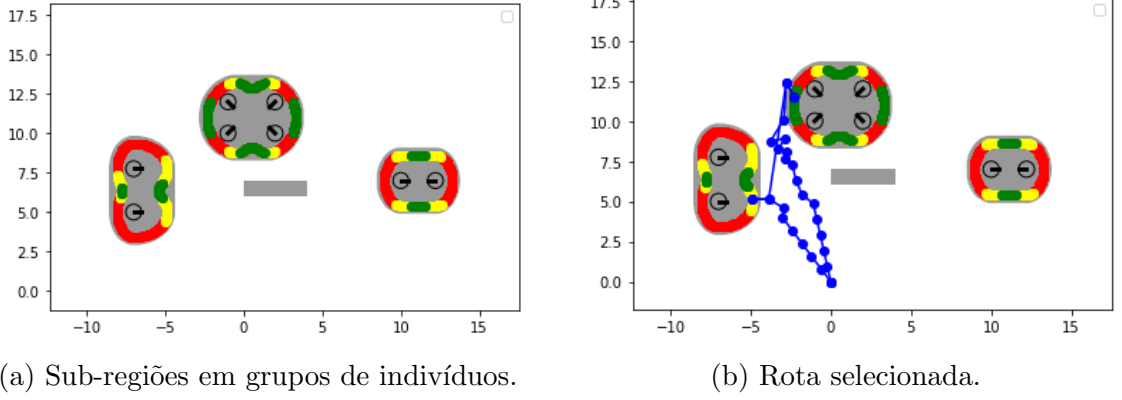


Figura 1.3: Exemplo de aplicação do COPS no contexto de HRI [Silva, 2025].

Tendo em vista esses desafios, desenvolvemos o presente trabalho modelando uma nova variação do OP que considera estas relações de unidade e exclusividade entre os pontos de interesse. Diversos exemplos de aplicações práticas são apresentados no Apêndice B.

1.3 Problema

O problema tratado neste trabalho é uma variação do OP, denominada *Clustered Orienteering Problem with Subgroups* (COPS). Em sua essência, a proposta é modelar e resolver cenários nos quais os locais-alvo ou pontos de interesse estão organizados em clusters e subdivididos em subgrupos. A principal característica desse modelo é que uma recompensa é associada a cada subgrupo, sendo obtida apenas quando todos os seus locais internos forem visitados. A decisão de qual subgrupo visitar, no entanto, é restrita pela limitação de orçamento, o que adiciona uma camada adicional de complexidade ao problema.

Mais especificamente, o objetivo do COPS é determinar uma rota que maximize a recompensa total coletada, respeitando as seguintes restrições principais:

- **Orçamento de viagem:** A soma das distâncias ou tempos de deslocamento na rota não pode exceder um orçamento pré-determinado $T_{\max}$.
- **Seleção de subgrupo por cluster:** Dentro de cada cluster, no máximo um subgrupo pode ser selecionado, e todos os PoIs desse subgrupo devem ser visitados para que a recompensa seja obtida.
- **Maximização da recompensa:** A rota deve maximizar a recompensa total, que é a soma das recompensas obtidas pela visita aos subgrupos selecionados.

Essa formulação expande as versões anteriores do problema de roteirização com lucros (OP) e dos problemas de agrupamento, como o COP e o SOP. Enquanto o COP considera a unidade entre os PoIs de um cluster (recompensa obtida apenas se todos os PoIs do cluster forem visitados), e o SOP impõe uma relação de exclusividade (onde no máximo um PoI de cada cluster pode ser visitado), o COPS introduz uma flexibilidade adicional ao permitir a existência de subgrupos dentro dos clusters. Isso possibilita diferentes formas de maximizar a recompensa, levando em consideração as particularidades de cada subgrupo dentro de um mesmo cluster.

Portanto, em contraste com COP e SOP, um cluster agora pode fornecer recompensas distintas de acordo com o subgrupo visitado. Essa formulação é útil em vários cenários, nos quais se pode usar os grupos para modelar tarefas de uma missão mais geral, enquanto os subgrupos definem maneiras específicas de executar cada uma das tarefas ou outras particularidades.

A Figura 1.4 ilustra o problema e mostra uma possível solução (caminho azul). Note que o cluster 6 não teve nenhum subgrupo visitado, devido ao limite de energia.

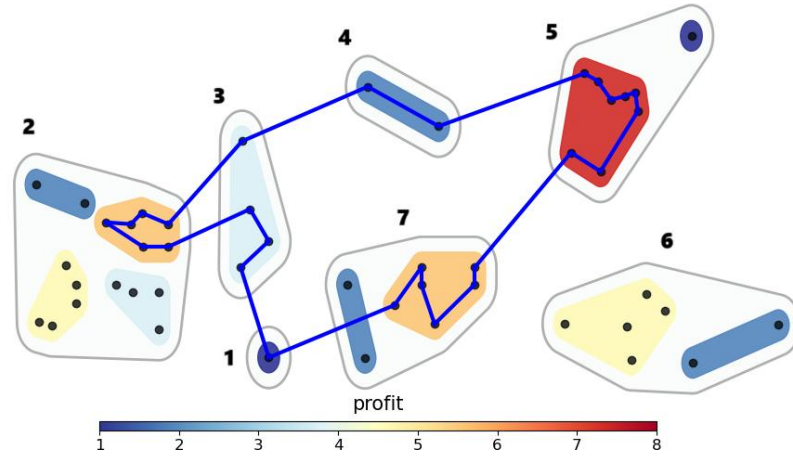


Figura 1.4: Exemplo de sete clusters com subgrupos distintos. A recompensa associada aos subgrupos é representada por suas cores. O cluster com um único vértice é o ponto inicial/final, e a linha azul é uma rota possível encontrada pelo algoritmo desenvolvido neste trabalho.

Uma definição, ainda que superficial, para o problema abordado neste trabalho, com o principal objetivo de introduzir a ideia geral, é apresentada a seguir. Os conceitos envolvidos e uma descrição formal para o problema serão apresentados no Capítulo 2.

Problema 1 (Clustered OP with Subgroups). *Considere uma coleção de clusters denotada por $\mathcal{C}$, onde cada cluster $c \in \mathcal{C}$ contém um ou mais subgrupos internos $\mathcal{S}' \subset \mathcal{S}$. Uma recompensa r_s é obtida para cada $s \in \mathcal{S}'$ quando todos os vértices de s são visitados. O objetivo é determinar uma rota que não ultrapasse o orçamento de viagem T_{max} , visite no máximo um subgrupo por cluster $c \in \mathcal{C}$ e maximize a recompensa total coletada.*

Essa formulação oferece uma representação mais flexível e realista para diversos cenários de planejamento de rotas, ao incorporar relações de complementaridade e exclusividade entre os pontos de interesse. O principal desafio consiste em equilibrar a maximização da recompensa com as restrições impostas pelo orçamento de viagem, decidindo estrategicamente quais subgrupos devem ser visitados em cada cluster. Vale destacar que o modelo considera apenas um único veículo, e também não inclui restrições relacionadas a janelas de tempo ou ordem de visitação. Essas e outras extensões serão discutidas ao longo deste trabalho, abrindo espaço para futuras investigações e aprimoramentos do modelo proposto.

1.4 Contribuições da Pesquisa

Esta dissertação propõe, modela e resolve o COPS, uma nova variante do OP que introduz maior flexibilidade na estrutura de agrupamento de PoIs, tornando possível representar e otimizar decisões em cenários mais realistas e complexos quando comparados com seus predecessores COP e SOP.

As principais contribuições deste trabalho são:

- **Formulação matemática unificada:** Desenvolveu-se um modelo matemático capaz de integrar o COPS e suas variantes anteriores (COP e SOP), por meio de uma única formulação parametrizável que generaliza os diferentes problemas.
- **Desenvolvimento de métodos de solução:** Foram implementadas duas abordagens principais para resolver o problema:
 - Um método exato baseado em *Integer Linear Programming* (Programação Linear Inteira, ILP), que garante a obtenção da solução ótima para instâncias de pequeno porte;
 - Uma heurística baseada na meta-heurística Busca Tabu, projetada para lidar com instâncias maiores, oferecendo soluções de alta qualidade em tempos computacionais reduzidos.
- **Avaliação experimental abrangente:** Realizou-se uma bateria de experimentos computacionais com instâncias *benchmark* além de cenários próprios simulados. As abordagens propostas foram comparadas entre si e também com métodos da literatura, demonstrando competitividade em relação à qualidade das soluções. Os resultados também validaram a flexibilidade e aplicabilidade do modelo COPS em problemas ainda não explorados na literatura.

Os principais resultados desta pesquisa foram publicados em um periódico:

- L. E. Almeida, C. A. Valle and D. G. Macharet, “Clustered Orienteering Problem With Subgroups”, in *IEEE Robotics and Automation Letters*, vol. 10, no. 12, pp. 12413-12420, 2025. doi: 10.1109/LRA.2025.3623047.

1.5 Estrutura da Dissertação

Além deste capítulo introdutório, esta dissertação está organizada da seguinte forma:

- O Capítulo 2 descreve formalmente o problema (COPS) proposto, destacando suas características, generalizações e uma formulação matemática unificada que abrange também os modelos COP e SOP.
- No Capítulo 3, se discute os conceitos básicos de otimização combinatória, incluindo grafos, complexidade computacional e problemas clássicos de roteamento como o TSP e o OP. Também são abordadas suas variantes relevantes, como o COP e o SOP, entre outras. Além de técnicas clássicas de resolução para os problemas relacionados ao OP e lacunas identificadas na literatura.
- No Capítulo 4, são apresentados os métodos desenvolvidos para a resolução do COPS, incluindo uma abordagem exata baseada em ILP e uma meta-heurística baseada na Busca Tabu.
- O Capítulo 5 traz os experimentos computacionais realizados para avaliar o desempenho dos métodos propostos, além da análise dos resultados em diferentes instâncias do problema.
- Por fim, o Capítulo 6 apresenta as conclusões do trabalho, discute suas limitações e propõe possíveis direções para trabalhos futuros relacionados ao tema.

Capítulo 2

Formalização do Problema

Seja $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ um grafo completo e não direcionado, onde $\mathcal{V}$ representa o conjunto de vértices e $\mathcal{E}$ o conjunto de arestas. Cada aresta $e \in \mathcal{E}$ possui um custo de travessia $t_e \in \mathbb{R}^+$, que pode representar distância, tempo ou qualquer outra métrica relevante. A rota total não pode exceder um orçamento máximo $T_{\max}$.

Os vértices do grafo são organizados hierarquicamente em dois níveis:

- **Subgrupos:** Um conjunto $\mathcal{S}$ de subconjuntos de vértices, onde cada subgrupo $s \in \mathcal{S}$ é composto por um subconjunto de vértices $s \subset \mathcal{V}$. Um mesmo vértice pode pertencer a múltiplos subgrupos. A cada subgrupo s está associado um lucro $r_s \in \mathbb{R}^+$, que é coletado somente se **todos os vértices** de s forem visitados;
- **Clusters:** Um conjunto $\mathcal{C}$ de subconjuntos de subgrupos $\mathcal{S}'$, onde cada cluster $c \in \mathcal{C}$ é formado por um subconjunto de subgrupos, ou seja, $c = \mathcal{S}' \subset \mathcal{S}$. Um mesmo subgrupo pode pertencer a múltiplos clusters.

A regra central do problema é que, para cada cluster $c \in \mathcal{C}$, **no máximo um subgrupo** pode ser selecionado para visita, e **todos os vértices** desse subgrupo devem ser incluídos na rota. Nenhum cluster pode ter mais de um subgrupo selecionado.

A rota $\mathcal{P} = \{p_1, p_2, \dots, p_w\} \subseteq \mathcal{V}$ é uma sequência ordenada de vértices que representa o caminho do agente, com as seguintes condições:

- A rota **inicia** no vértice 1, que pertence a um subgrupo especial $s_1 = \{1\}$, pertencente ao cluster inicial $c_1 = \{s_1\}$;
- A rota **termina** em um vértice pertencente a um dos subgrupos de $c_{end} \in \mathcal{C}$. c_{end} é o cluster final, contendo possíveis pontos finais para o percurso. Vale destacar que c_{end} pode possuir vários subgrupos, cada um contendo um único vértice, facilitando a modelagem de cenários como depósitos de chegada distintos ou estações de recarga. Se $c_{end} = \{1\}$, a solução será uma rota circular, começa e termina no mesmo ponto; caso contrário, a solução será uma rota não circular, tendo início e fim em pontos distintos;

- A soma dos custos das arestas percorridas na rota não pode exceder o orçamento $T_{\max}$;
- A visita aos vértices pode ser feita em qualquer ordem, os vértices de um subgrupo não precisam ser visitados em sequência;
- Um vértice pode pertencer a múltiplos subgrupos, assim como um subgrupo pode estar associado a múltiplos clusters.
- Um vértice é considerado *cliente* se não pertence ao subgrupo do cluster inicial nem a nenhum subgrupo do cluster final. Os conjuntos $\mathcal{V}_{client} \subset \mathcal{V}$ e $\mathcal{S}_{client} \subset \mathcal{S}$ representam, respectivamente, os vértices e subgrupos compostos exclusivamente por vértices clientes.

O **Clustered Orienteering Problem with Subgroups (COPS)** consiste em determinar uma rota que maximize o total de recompensas coletadas, respeitando um orçamento de viagem $T_{\max}$.

Seja $\mathcal{C}$ o conjunto de todos os clusters, onde cada cluster $c \in \mathcal{C}$ contém um ou mais subgrupos internos $\mathcal{S}' \subset \mathcal{S}$. Cada subgrupo $s \in \mathcal{S}$ está associado a uma recompensa r_s , a qual é obtida quando todos os vértices pertencentes a s são visitados.

O objetivo é encontrar uma rota que **visite no máximo um subgrupo por cluster** e **cubra todos os vértices do subgrupo selecionado**, sem exceder o orçamento $T_{\max}$, de forma a **maximizar a recompensa total obtida**. A Figura 2.1 ilustra o problema e mostra uma possível solução (caminho azul). Note que o cluster 5 não foi visitado, devido ao limite de energia estabelecido. O cluster 2 possui dois subgrupos os quais compartilham um vértice, e os clusters 3 e 6 também compartilham um vértice.

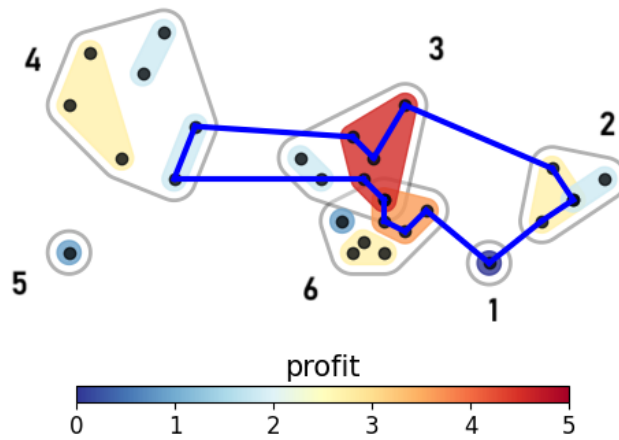


Figura 2.1: Exemplo de seis clusters com subgrupos distintos. A recompensa associada aos subgrupos é representada por suas cores. O cluster 1 é o ponto inicial/final, e a linha azul é uma rota possível encontrada pelo algoritmo desenvolvido neste trabalho.

Teorema 1. *O Clustered Orienteering Problem with Subgroups (COPS) é NP-difícil.*

Demonstração. A prova segue por redução a partir do OP, que é NP-difícil [Golden et al., 1987]. Considere uma instância do OP. Constrói-se uma instância equivalente do COPS em que cada cluster contém exatamente um subgrupo e cada subgrupo contém exatamente um vértice, com recompensa igual à do vértice correspondente no OP. Nessa configuração, as restrições adicionais do COPS tornam-se triviais, e o problema resultante é equivalente ao OP. Como essa transformação é polinomial, conclui-se que o COPS é NP-difícil. $\square$

2.1 Representação Matemática e Notação

Nesta seção, apresentamos a notação formal utilizada na modelagem do problema *Clustered Orienteering Problem with Subgroups* (COPS), incluindo os conjuntos, parâmetros e variáveis que compõem a instância e a solução do problema.

2.1.1 Conjuntos Base

- $\mathcal{V}$: Conjunto de vértices do grafo. Onde cada vértice pertence a um espaço euclidiano bidimensional ou tridimensional, ou seja, $v \in \mathcal{V} \subset \mathbb{R}^n$, $n \in \{2, 3\}$;
- $\mathcal{E}$: Conjunto de arestas do grafo, onde cada aresta $e \in \mathcal{E}$ conecta dois vértices;
- $\mathcal{S}$: Conjunto de subgrupos. Onde cada subgrupo é formado por um subconjunto de vértices $s \subset \mathcal{V}$;
- $\mathcal{C}$: Conjunto de clusters. Onde cada cluster é formado por um subconjunto de subgrupos $c \subset \mathcal{S}$.

2.1.2 Subconjuntos específicos

- c_1 : Cluster inicial (possui vértice de início da rota);

- c_{end} : Cluster final (possíveis vértices que finalizam a rota). No caso particular de rota circular $c_{end} = c_1$;
- $\mathcal{V}_{client} \subset \mathcal{V}$: Subconjunto de vértices clientes, ou seja, que não pertencem ao cluster inicial nem ao cluster final;
- $\mathcal{S}_{client} \subset \mathcal{S}$: Subconjunto de subgrupos compostos exclusivamente por vértices em $\mathcal{V}_{client}$;
- $\mathcal{S}' \subset \mathcal{S}$: Subconjunto de subgrupos que formam um cluster $c = \mathcal{S}'$.

2.1.3 Parâmetros

- t_e : Custo de travessia da aresta $e \in \mathcal{E}$;
- T_{max} : Orçamento máximo permitido para o custo total da rota;
- r_s : Lucro associado ao subgrupo $s \in \mathcal{S}$, coletado apenas se todos os vértices de s forem visitados.

2.1.4 Variáveis de Decisão (para formulações posteriores)

- $x_e \in \{0, 1\}$: Indica se a aresta $e \in \mathcal{E}$ é percorrida na rota;
- $y_j \in \{0, 1\}$: Indica se o vértice $j \in \mathcal{V}$ é visitado;
- $z_s \in \{0, 1\}$: Indica se o subgrupo $s \in \mathcal{S}$ foi ativado (todos os seus vértices foram visitados);

2.1.5 Funções e Regras

- Um subgrupo s é considerado **visitado** se $y_j = 1$ para todo $j \in s$;

- Um cluster c é considerado **atendido** se e somente se **exatamente um** dos sub-grupos $s \in c$ for visitado;

- O lucro total da solução é

$$\sum_{s \in \mathcal{S}} r_s z_s; \quad (1)$$

- O custo total da rota é

$$\sum_{e \in \mathcal{E}} t_e x_e \quad (2)$$

3.1.1 Definição Formal

Dado um grafo completo e ponderado $G = (V, E)$, em que $V = \{v_1, v_2, \dots, v_n\}$ representa o conjunto de cidades ou nós, E corresponde ao conjunto de arestas que conectam pares de cidades e cada aresta $(v_i, v_j) \in E$ possui um custo c_{ij} associado, que pode representar distância, tempo ou outro critério relevante. O objetivo do TSP é determinar um ciclo Hamiltoniano de custo mínimo. Tal ciclo pode ser descrito por uma permutação dos vértices $\pi = (\pi_1, \pi_2, \dots, \pi_n)$, de modo que o custo total do percurso seja dado por

$$C(\pi) = c_{\pi_n \pi_1} + \sum_{i=1}^{n-1} c_{\pi_i \pi_{i+1}},$$

e seja minimizado.

3.1.2 Classificação e Complexidade

O TSP pertence à classe de problemas NP-difíceis, o que significa que não se conhece um algoritmo que o resolva de forma eficiente (em tempo polinomial) para todas as instâncias. Além disso, verificar se uma solução é válida (isto é, se ela é um ciclo Hamiltoniano) pode ser feito em tempo polinomial, o que o coloca também na classe NP.

Existem diversas variantes do TSP, cada uma apresentada com características particulares. No caso do *TSP Simétrico*, os custos satisfazem a condição $c_{ij} = c_{ji}$, o que implica que o custo de deslocamento entre duas cidades é o mesmo em ambos os sentidos. Já no *TSP Assimétrico* (ATSP), essa simetria não é garantida, sendo comum em aplicações como redes de tráfego urbano, onde o custo para ir de um ponto a outro pode diferir do custo no sentido inverso. Outras variantes incluem o TSP com janelas de tempo, o TSP com múltiplos caixeiros e diversas extensões que incorporam restrições adicionais conforme a aplicação.

3.1.3 Aplicações Práticas

Apesar de sua formulação abstrata, o TSP está presente em uma ampla variedade de aplicações reais. Por exemplo, o roteamento de veículos de entrega [Gomes et al., 2021],

o planejamento de rotas para robôs móveis [Nedjatia e Vizvárib, 2020], a otimização de circuitos impressos (PCB) [Duman e Or, 2004] e o sequenciamento de operações industriais [Kumar e Luo, 2003]. O problema também é aplicado em tarefas de planejamento de inspeções e coletas [Dinkel et al., 1976], na otimização de roteamento em redes de computadores [Bharath-Kumar e Jaffe, 2003], no gerenciamento de satélites [Cho e Choi, 2019] e drones [Roberti e Ruthmair, 2021], além de desempenhar um papel relevante na compilação de circuitos quânticos em telecomunicações quânticas [Osaba et al., 2022]. Assim, sua relevância prática se estende por diferentes domínios da engenharia, logística, computação e tecnologias emergentes.

3.1.4 Métodos de Solução

Devido à sua complexidade, diferentes abordagens têm sido empregadas para resolver o TSP. Entre os métodos exatos, destacam-se aqueles baseados em programação inteira, algoritmos de *branch and bound* [Balas e Toth, 1983] e técnicas de *cutting planes* [Levine, 2000], os quais garantem a obtenção da solução ótima, embora seu custo computacional cresça rapidamente com o tamanho da instância.

Além desses métodos, utilizam-se também heurísticas e meta-heurísticas, que consistem em estratégias rápidas capazes de produzir soluções de boa qualidade, como o método do vizinho mais próximo [Hurkens e Woeginger, 2004] e a técnica de inserção mais barata [Hassin e Keinan, 2008]. Complementando essas abordagens, outras que oferecem mecanismos avançados com buscas iterativas no espaço de soluções, proporcionando um equilíbrio entre qualidade e tempo de execução. Entre elas destacam-se o *Simulated Annealing* [Skišcim e Golden, 1983], os Algoritmos Genéticos [Razali, Geraghty et al., 2011], o método GRASP [Marinakis et al., 2005], a Ant Colony Optimization [Yang et al., 2008], a Busca Tabu [Oliveira et al., 2015], além de heurísticas de melhoria como 2-opt [Uddin et al., 2023] e o procedimento Lin–Kernighan [Helsgaun, 2000].

Apesar das dificuldades teóricas associadas ao TSP, muitas de suas instâncias práticas podem ser resolvidas com eficiência, seja pela aplicação de algoritmos exatos em problemas de tamanho moderado, seja pelo uso de meta-heurísticas capazes de produzir soluções de alta qualidade em tempos computacionais reduzidos.

3.2 Problema da Mochila

O *Problema da Mochila* (em inglês, *Knapsack Problem*) é um dos problemas fundamentais da otimização combinatória. Amplamente estudado na pesquisa operacional e em aplicações práticas, sua formulação simples contrasta com a dificuldade de obter soluções exatas para instâncias grandes.

De forma geral, o problema consiste em escolher, a partir de um conjunto de itens com valores e pesos associados, uma seleção que maximize o valor total sem ultrapassar a capacidade de uma mochila.

3.2.1 Definição Formal

Dado um conjunto de n itens, em que cada item $i \in \{1, 2, \dots, n\}$ possui um valor $v_i > 0$ e um peso $w_i > 0$, e considerando uma mochila com capacidade máxima W , o objetivo consiste em determinar o subconjunto de itens que maximize o valor total transportado, respeitando a restrição de capacidade. Esse problema pode ser formulado como

$$\begin{aligned} &\text{Maximizar} && \sum_{i=1}^n v_i x_i \\ &\text{sujeito a} && \sum_{i=1}^n w_i x_i \leq W \quad x_i \in \{0, 1\}, \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

onde x_i indica se o item i foi incluído (1) ou não (0) na mochila.

3.2.2 Classificação e Complexidade

O Problema da Mochila 0-1 é classificado como NP-completo, o que implica que não se conhece um algoritmo em tempo polinomial capaz de resolvê-lo para todas as instâncias possíveis. Existem, entretanto, variantes do problema que apresentam diferentes níveis de complexidade. A *Mochila Fracionária*, por exemplo, permite que os itens sejam fracionados, isto é, $x_i \in [0, 1]$, podendo ser resolvida de forma eficiente por meio de algoritmos gulosos. Já na *Mochila 0-1*, cada item deve ser totalmente incluído ou excluído, o que mantém o problema dentro da classe NP-completa. Outras extensões,

como o problema de Múltiplas Mochilas ou a Mochila Multidimensional, ampliam tanto a complexidade quanto a aplicabilidade do modelo.

Apesar dessas dificuldades teóricas, diversas instâncias de tamanho moderado podem ser resolvidas de maneira eficiente, especialmente por meio de técnicas de programação dinâmica ou algoritmos de aproximação.

3.2.3 Aplicações Práticas

O problema da mochila encontra aplicações em diversos contextos do mundo real. Entre eles, destacam-se a alocação de recursos limitados [Vanderster et al., 2009], o planejamento de orçamentos e investimentos [Singh, 2020], a seleção de portfólios financeiros [Patalia e Kulkarni, 2011] e a otimização de carga em transporte e logística [Y. Zhang et al., 2020]. O problema também é relevante em tarefas de compressão de dados [Correia et al., 2018], na escolha de anúncios ou campanhas com restrições de espaço, ou tempo [Akgun, 2017], bem como no design de sistemas embarcados sujeitos a limitações de energia ou capacidade de processamento [Ray et al., 2004].

3.2.4 Métodos de Solução

Diversas abordagens são empregadas para resolver o Problema da Mochila, dependendo do tipo e tamanho da instância considerada. A programação dinâmica [Ahmed e Younas, 2011] constrói uma tabela que armazena as soluções ótimas para subproblemas menores, permitindo identificar a combinação de itens que maximiza o valor total sem ultrapassar a capacidade da mochila. Heurísticas e meta-heurísticas, incluindo Algoritmos Genéticos [Spillman, 1995], Simulated Annealing [Moradi et al., 2022], GRASP [Vianna e Arroyo, 2004], Ant Colony Optimization [Ke et al., 2010] e tabu search [Hanafi e Freville, 1998], são utilizadas para resolver instâncias grandes de forma eficiente, ainda que sem garantias de otimalidade. Por fim, métodos exatos baseados em programação inteira são aplicáveis a instâncias pequenas a médias, empregando técnicas como *branch and bound* [Shih, 1979] e *cutting planes* [Ferreira et al., 1996].

Assim como no TSP, apesar dos desafios teóricos associados ao problema da mochila, muitas instâncias práticas podem ser resolvidas com alto grau de eficiência, produzindo soluções de elevada qualidade.

3.3 *Orienteering Problem* (OP)

Orientação é um esporte ao ar livre que geralmente é praticado em áreas densamente florestadas (Figura 3.2). Localizados na floresta estão vários “pontos de controle”, cada um com uma pontuação associada. Os competidores, munidos de bússola e mapa, devem visitar um subconjunto desses pontos de controle a partir do ponto de partida (nó 1) de forma a maximizar sua pontuação total e retornar ao ponto final (nó n) dentro de um tempo determinado. Os competidores que chegam atrasados são desclassificados ou recebem uma penalidade severa.



Figura 3.2: Esportista em uma competição de orientação por [North East Orienteering Association, 2025].

O problema de otimização combinatória conhecido como *Orienteering Problem* (OP) é inspirado no cenário da competição de orientação [Golden et al., 1987], modelando a tarefa de determinar uma rota ótima que percorra um subconjunto de pontos de interesse (PoIs) em um ambiente geográfico, com o objetivo de maximizar a pontuação total obtida, respeitando um limite de tempo ou distância.

O termo *Orienteering Problem* foi introduzido formalmente na literatura científica por [Golden et al., 1987]. Nesse trabalho, os autores descreveram o problema como um desafio combinatório e propuseram uma formulação matemática que estabeleceu as bases para estudos subsequentes.

Desde então, o OP tem sido amplamente estudado nas áreas de ciência da computação, pesquisa operacional e otimização. Sua natureza desafiadora, combinando aspectos práticos e teóricos, tem motivado o desenvolvimento de diversos algoritmos e abordagens heurísticas e meta-heurísticas voltadas à sua resolução eficiente.

O OP possui uma forte relação com dois problemas clássicos da otimização combinatória: o *Traveling Salesman Problem* (Problema do Caixeiro Viajante, TSP) e o *Knapsack Problem* (Problema da Mochila, KP).

A relação entre o OP e o TSP surge do fato de ambos envolverem o planejamento

de uma rota que percorre um conjunto de pontos. No entanto, as principais diferenças entre esses problemas são notáveis. No OP, cada PoI possui um valor de recompensa associado, e o objetivo principal é maximizar a recompensa obtida ao visitar esses pontos. Em contraste, o TSP foca em minimizar a distância total percorrida, buscando a rota mais curta possível. Enquanto no TSP a solução ótima exige a visitação de todos os pontos e o retorno ao ponto inicial, no OP não é necessário visitar todos os pontos para que uma solução seja considerada válida. Além disso, no OP, o objetivo não é necessariamente encontrar a rota de menor distância, mas sim garantir que a distância total percorrida esteja dentro de um orçamento pré-estabelecido, enquanto a maximização da recompensa é o critério decisivo.

Já o KP, envolve selecionar um subconjunto de itens de uma lista, considerando seus valores e pesos, para que a soma dos valores seja maximizada dentro de uma restrição de capacidade. A relação com o OP se dá pela semelhança na ideia de selecionar elementos que tragam um benefício máximo. No caso do OP queremos obter o maior lucro possível dentro de um limite de tempo/distância, e no caso do Problema da Mochila queremos selecionar itens com pesos e valores de forma a maximizar o valor total transportado sem ultrapassar a capacidade da mochila.

Essas relações entre o OP, o TSP e o KP ajudam a entender sua complexidade e a desenvolver técnicas e algoritmos aplicáveis ao OP.

Em resumo, o OP consiste em determinar a rota mais eficiente para percorrer um conjunto de PoI, visando maximizar uma função de pontuação acumulada ao longo da rota, dentro de uma restrição de tempo ou distância total percorrida. Cada PoI possui um valor associado que representa sua importância ou atratividade, e a rota deve ser planejada de forma a otimizar essa pontuação acumulada.

3.3.1 Definição e formulação clássica do OP

O OP é um problema de otimização combinatória que modela cenários nos quais um viajante deve selecionar um subconjunto de locais a serem visitados dentro de um tempo de viagem fixo, com o objetivo de maximizar os pontos acumulados. Este problema foi adaptado para diversos contextos, como por exemplo, logísticos, roteirização de veículos com restrições temporais, gestão de inventários e agendamento de projetos.

Formalmente, o OP pode ser representado da seguinte maneira. Considere um conjunto de locais (nós) $N = \{1, 2, \dots, n\}$, com o nó inicial designado como 1 e o nó final como n . Cada nó i , exceto os nós inicial e final, possui um valor de recompensa associado $s(i) \geq 0$, e os tempos ou distâncias de viagem entre pares de nós $i, j \in N$ são dados por

$t(i, j) \geq 0$. A rota deve respeitar um tempo máximo de viagem $T_{\max}$ e começar no nó 1, terminando no nó n .

O objetivo é encontrar uma rota R que maximize a pontuação total das visitas aos nós intermediários, formalmente expressa por

$$S(R) = \sum_{i \in R \setminus \{1, n\}} s(i),$$

sujeita à restrição de orçamento de tempo,

$$T(R) = \sum_{(i, j) \in R} t(i, j) \leq T_{\max}.$$

Essa formulação encapsula a essência do problema: selecionar um caminho que comece no nó inicial e termine no destino, garantindo a maior pontuação possível sem ultrapassar o limite de tempo preestabelecido.

Modelo Matemático Clássico

O OP pode ser visto como uma variação do TSP, com um componente adicional de pontuação e uma restrição temporal. As formulações clássicas frequentemente adotam uma abordagem de programação inteira:

Variáveis de Decisão

- $x_{ij} \in \{0, 1\}$: igual a 1 se a rota incluir o trajeto de i a j , e 0 caso contrário.
- $y_i \in \{0, 1\}$: igual a 1 se o nó i for visitado, e 0 caso contrário.

Restrições

- **Conservação de fluxo:**

$$\sum_{j \in N} x_{ij} = y_i \quad \forall i \neq n, \quad \sum_{i \in N} x_{ij} = y_j \quad \forall j \neq 1$$

- **Nós de início e fim:**

$$\sum_{j \in N} x_{1j} = 1, \quad \sum_{i \in N} x_{in} = 1$$

- **Restrição de orçamento:**

$$\sum_{i, j \in N} t(i, j) \cdot x_{ij} \leq T_{\max}$$

- **Eliminação de sub-rotas e conectividade** para garantir que a solução seja uma única rota.

Função Objetivo

$$\text{Maximizar } \sum_{i \in N} s(i) \cdot y_i$$

Este modelo captura de maneira eficiente o núcleo do problema, mas é computacionalmente desafiador para grandes instâncias devido à sua complexidade NP-difícil.

Variações e Considerações do Mundo Real

Embora a formulação clássica do OP enfatize a maximização das pontuações dentro de restrições de tempo, cenários práticos frequentemente incorporam complexidades adicionais, como penalidades por atrasos, janelas temporais e prioridades de serviço. A inclusão desses elementos torna o problema mais realista e aplicável a diversos contextos industriais e logísticos. Penalidades por atrasos, por exemplo, surgem em situações em que o tempo excedido de viagem acarreta redução da pontuação ou aplicação de sanções, o que demanda ajustes na estratégia de roteirização. Já as janelas temporais impõem intervalos específicos durante os quais determinados pontos podem ser visitados, introduzindo uma camada adicional de coordenação temporal. Além disso, prioridades de serviço podem ser estabelecidas quando alguns locais apresentam maior urgência ou importância, exigindo que sejam atendidos de maneira preferencial.

Apesar dessas variações, o problema fundamental permanece o mesmo: selecionar os locais de maior valor possível, respeitando o orçamento total de tempo ou distância disponível.

3.3.2 Desafios computacionais

A formulação matemática do *Orienteering Problem* geralmente envolve a definição de um grafo, onde os PoIs são representados como vértices e as distâncias ou custos entre os PoIs são representados como arestas ponderadas. O objetivo é encontrar o caminho que maximize a pontuação acumulada, respeitando as restrições de orçamento.

Diversas abordagens têm sido propostas para resolver o OP. Algoritmos exatos, como *Mixed Integer Linear Programming* (Programação Linear Inteira Mista, MILP) podem ser aplicados para encontrar soluções ótimas em instâncias de pequeno porte. No entanto, devido à sua natureza NP-difícil, o OP é geralmente tratado com algoritmos heurísticos e meta-heurísticas.

Algoritmos baseados em busca local, como *Variable Neighborhood Search* (Busca em Vizinhança Variável, VNS) [Sevкли e Sevilgen, 2006; Pěnička et al., 2019], Busca Tabu [Liang et al., 2002] e algoritmo genético [Tasgetiren e A. E. Smith, 2000], são

amplamente utilizados para encontrar soluções aproximadas de boa qualidade. Além disso, algoritmos baseados em colônias de formigas [Liang e A. E. Smith, 2006], otimização por enxame de partículas [Dallard et al., 2007] e algoritmos evolutivos [Kobeaga et al., 2018] também têm sido aplicados com sucesso ao OP.

Como pode ser visto, o OP é uma variação interessante e desafiadora do problema de roteamento, com aplicações em diversas áreas. Ao otimizar o percurso para visitar pontos de interesse com restrições de tempo e maximizar a pontuação acumulada, o OP pode auxiliar na tomada de decisões eficientes e na otimização de recursos em diversas situações do mundo real.

3.4 Principais variantes do *Orienteering Problem*

Desde a sua formulação inicial, muitas variações do *Orienteering Problem* têm sido implementadas, neste capítulo exploraremos essas generalizações. Enquanto o OP tradicional se concentra na otimização do roteamento para visitar pontos de interesse com restrições de orçamento e maximizar a pontuação acumulada, suas variantes estendem e adaptam esse conceito fundamental para lidar com diferentes contextos e desafios específicos.

A seguir, detalharemos com mais profundidade algumas variações do *Orienteering Problem* que são relevantes para este trabalho, explorando suas nuances e características específicas, além de discutir os principais algoritmos de solução que têm sido propostos para lidar com essas variantes do problema.

3.4.1 *Clustered Orienteering Problem* (COP)

O *Clustered Orienteering Problem* (COP) [Angelelli et al., 2014] é uma generalização do OP, onde os PoIs são agrupados em clusters e um lucro é associado a cada cluster, sendo coletado somente se todos os PoIs do cluster forem visitados (Figura 3.3). O objetivo é encontrar a rota de veículo que maximize o lucro total coletado, respeitando um limite de tempo de percurso. O problema é resolvido, no artigo original, por meio dos algoritmos *Branch-and-Cut* [Padberg e Rinaldi, 1987] e Busca Tabu [Glover, 1986].

O *Branch-and-Cut* desenvolvido é uma abordagem exata que combina técnicas de *branch-and-bound* com a geração dinâmica de cortes para encontrar soluções ótimas. O

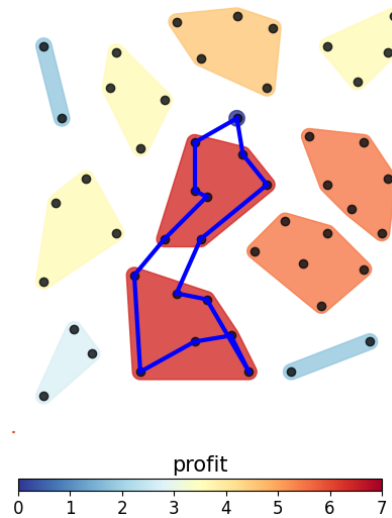


Figura 3.3: Um exemplo de como os clientes são organizados em grupos no *Clustered Orienteering Problem* (COP). O lucro é coletado somente se todos os PoIs forem visitados.

algoritmo inicia a partir de uma formulação linear relaxada do COP. Em cada nó da árvore de *branch-and-bound*, resolve-se a relaxação linear corrente e, sempre que soluções fracionárias são encontradas, novas restrições (cortes), como desigualdades válidas, podem ser identificadas por meio de algoritmos de separação e adicionadas ao modelo. Além disso, são implementadas regras de ramificação e algoritmos de separação para melhorar a eficiência do algoritmo. Esse processo iterativo de geração de cortes e ramificação prossegue até que uma solução inteira ótima seja encontrada.

Já a Busca Tabu desenvolvida é um algoritmo heurístico. Neste método, uma solução inicial é explorada iterativamente, permitindo movimentos que podem piorar temporariamente a solução atual, desde que não violem regras tabu. O algoritmo mantém uma lista de soluções tabu para evitar retornar a soluções anteriores e incentivar a exploração de novas regiões do espaço de soluções. Além disso, diferentes estruturas de vizinhança são definidas para explorar soluções alternativas. A Busca Tabu foi implementada com diferentes variantes, sendo que a abordagem baseada em um esquema de *multi-start* se mostrou a mais eficaz. Esse algoritmo conseguiu fornecer soluções de alta qualidade em um curto período, sendo uma alternativa eficiente para lidar com instâncias do COP que não puderam ser resolvidas pelo algoritmo exato.

O COP têm aplicações em áreas como logística, gerenciamento de frota e turismo, onde a agrupamento de pontos de interesse desempenha um papel importante na eficiência do roteamento. A inclusão do conceito de clusterização amplia o escopo do OP e proporciona uma abordagem mais realista e abrangente para o planejamento de rotas.

Algumas variações do COP já tem sido introduzidas como, por exemplo, o *Clustered Team Orienteering Problem* (CluTOP) [Yahiaoui et al., 2019] onde o COP é estendido

para múltiplos veículos. Já no *Clustered Coverage Orienteering Problem* (CCOP) [W. Zhang et al., 2020] os nós clientes são agrupados em clusters representando regiões, e um número mínimo de clientes devem ser visitados em cada cluster para que informações eficientes possam ser coletadas na região. O CCOP possui diversas aplicações práticas, como o monitoramento de áreas geográficas, onde é necessário garantir a cobertura eficiente de determinadas regiões. Por exemplo, em monitoramento ambiental, é importante visitar pontos estratégicos para coleta de dados ou amostras, garantindo uma cobertura adequada das áreas de interesse.

3.4.2 *Set Orienteering Problem* (SOP)

O SOP, introduzido por [Archetti et al., 2018], se distingue claramente do tradicional OP, trazendo elementos que modificam essencialmente a abordagem de roteamento. Nesta seção, examinaremos as características fundamentais do SOP em relação ao OP convencional, destacando suas diferenças e os desafios que elas implicam no desenvolvimento de soluções eficazes.

No SOP, os clientes são agrupados em conjuntos, e a cada conjunto é atribuído um valor de lucro. Esse lucro é obtido somente se no máximo um cliente do conjunto for visitado, como no exemplo da Figura 3.4. Esse método de agrupamento introduz uma nova dimensão ao problema de otimização em comparação com o OP, onde cada cliente tem seu próprio valor de lucro individual.

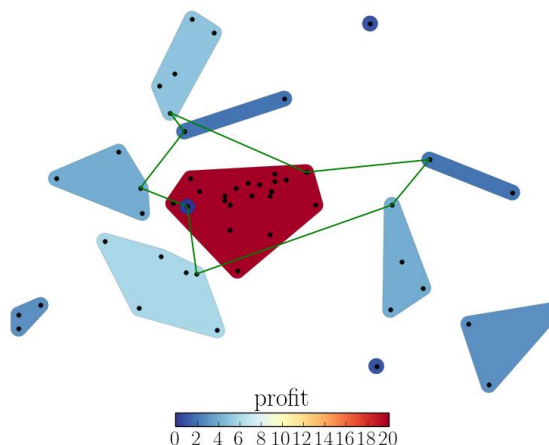


Figura 3.4: *Set Orienteering Problem* (SOP). No máximo um ponto de interesse de cada cluster deve ser visitado para se obter a recompensa associada àquele grupo, por [Pěnička et al., 2019].

Ao considerar essas diferenças, o SOP se destaca como uma variante desafiadora

do OP, proporcionando uma nova perspectiva para a otimização de rotas em cenários onde os clientes são agrupados em conjuntos com lucros associados. Compreender essas distinções é crucial para desenvolver abordagens eficazes na resolução do SOP e explorar seu potencial em aplicações práticas do mundo real. A solução introduzida em [Archetti et al., 2018] para resolver o SOP é denominada como *Matheuristic for the SOP* (MASOP). Uma *matheurística* é uma abordagem de otimização que combina elementos de métodos heurísticos e técnicas matemáticas para resolver problemas complexos eficientemente. O algoritmo proposto no estudo consiste em duas fases principais:

A primeira fase consiste na construção de uma solução inicial. O algoritmo inicia a construção da solução a partir do vértice de partida e então seleciona um cluster de clientes para visitar, em seguida, o algoritmo escolhe, de forma aleatória, um cliente dentro desse cluster e o coloca na solução, até que todos os clusters sejam visitados ou até que a capacidade de tempo do veículo seja atingida. Nesta fase o algoritmo visa criar uma solução viável que possa ser aprimorada na fase subsequente.

A segunda fase do algoritmo envolve a Busca Tabu, um método de otimização meta-heurística. A Busca Tabu é utilizada para melhorar iterativamente a solução inicial explorando o espaço de solução e realizando movimentos que levam a soluções melhores. Quando a Busca Tabu encontra dificuldades em encontrar uma nova solução viável, ela aproveita uma MILP para orientar sua busca. Essa combinação de busca heurística com MILP aprimora a capacidade do algoritmo de encontrar soluções de alta qualidade para o SOP.

Outro trabalho que também cria solução para o SOP é encontrado em [Pěnička et al., 2019], aqui os autores desenvolvem um algoritmo chamado *Variable Neighborhood Search for the Set Orienteering Problem* (VNS-SOP), fornecendo um método robusto para encontrar soluções de alta qualidade. Aqui estão alguns pontos-chave sobre como a VNS-SOP encontra soluções para o SOP:

O algoritmo VNS-SOP adota uma abordagem inicial gulosa, priorizando a minimização da distância em relação ao lucro adicional ao visitar novos conjuntos s_σ previamente não visitados. Ele seleciona conjuntos a serem adicionados à solução, formando uma sequência de conjuntos Σ_k . Em seguida, utilizando o algoritmo *Breadth-First Search* (busca em largura, BFS) em um grafo composto pelos conjuntos escolhidos e suas arestas conectando todos os vértices de um conjunto ao próximo na sequência, conforme mostrado na Figura 3.5, determinamos a sequência de vértices individuais Π_k que minimiza a distância percorrida nos conjuntos selecionados Σ_k . Essa solução inicial serve como ponto de partida para otimizações adicionais.

Após obter a solução inicial, o algoritmo VNS-SOP tenta melhorar a solução atual aplicando dois grupos de operadores de vizinhança predefinidos, os operadores *shake* e *local search*. Operações *shake* envolvem movimentos que alteram a ordem de visita dos conjuntos ou adicionam novos conjuntos, enquanto as operações *local search* examinam

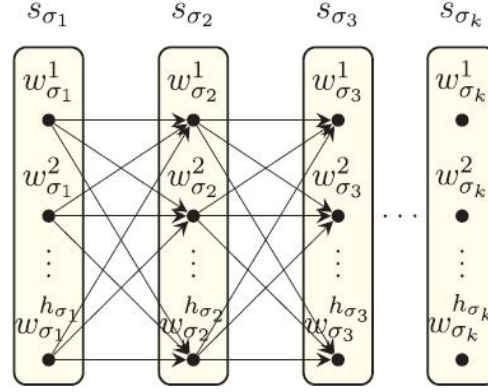


Figura 3.5: Grafo formado pela sequência de conjuntos no VNS-SOP [Pěnička et al., 2019].

modificações na sequência de conjuntos para melhorar a solução. Exemplos incluem mover conjuntos para novas posições ou trocar a ordem dos conjuntos para buscar soluções de melhor qualidade. Esses operadores ajudam a explorar eficientemente diferentes espaços de solução.

Ao aplicar iterativamente os operadores de vizinhança e diversificar a busca, a VNS-SOP pode refinar eficientemente as soluções e potencialmente escapar de ótimos locais para encontrar melhores soluções.

O algoritmo VNS-SOP proposto é ainda mais otimizado em termos de tempo ao empregar um teste rápido para limite de orçamento em cada nova solução avaliada. Em uma solução quase ótima típica do SOP, o comprimento total do caminho se aproxima do limite de orçamento $T_{\max}$, resultando em quase todas as modificações, como adições ou movimentos de conjuntos, ultrapassando o orçamento. Para identificar rapidamente tais cenários, calculamos e armazenamos previamente a distância mínima entre cada par de conjuntos, ou seja, o arco de comprimento mínimo entre eles, antes de criar a solução inicial do SOP. Por exemplo, ao adicionar o conjunto s_{σ_3} entre os conjuntos s_{σ_4} e s_{σ_5} , como mostrado na Figura 3.6, primeiro testamos se o limite inferior se enquadra no orçamento, utilizando os arcos de comprimento mínimo de s_{σ_5} para s_{σ_3} e de s_{σ_3} para s_{σ_4} . O caminho do limite inferior inclui ainda os caminhos mais curtos para o conjunto s_{σ_5} a partir do conjunto inicial e para o conjunto final a partir de s_{σ_4} , encontrados como a menor distância armazenada pela técnica de programação dinâmica para uma solução incumbente entre os vértices em s_{σ_5} e s_{σ_4} , respectivamente. Embora o limite inferior proposto possa ser inviável, devido à possível utilização de vértices diferentes nos conjuntos adicionados e nos conjuntos anteriores e subsequentes, uma solução viável não pode ter um comprimento menor. Assim, com operações simples, podemos descartar rapidamente soluções inválidas.

A VNS é conhecida por sua adaptabilidade a diferentes estruturas de problema e sua capacidade de lidar efetivamente com problemas de otimização combinatória complexos. Isso a torna uma escolha adequada para resolver o SOP, tornando o VNS-SOP uma

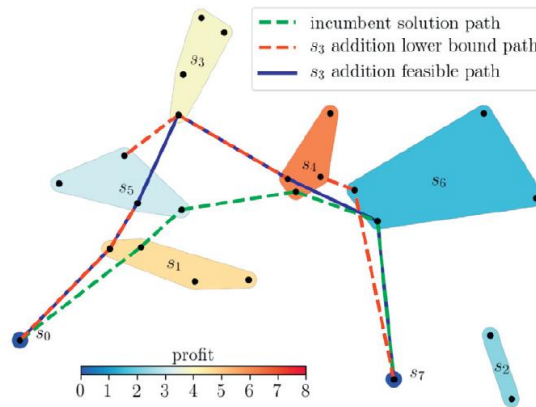


Figura 3.6: Exemplo de teste rápido para limite de orçamento por [Pěnička et al., 2019].

solução rápida e precisa para o problema.

Também existem variações do SOP, é o caso do *Set Team Orienteering Problem with Time Windows* (STOPTW) [Gunawan et al., 2021]. Nessa variação, o problema considera múltiplas equipes ou agentes que devem visitar um conjunto de pontos de interesse dentro de janelas de tempo específicas.

3.4.3 Outras variantes relevantes (breve panorama)

Uma extensão interessante do OP é conhecida como *Teams Orienteering Problem* (TOP) [Chao et al., 1996], que envolve equipes de veículos trabalhando juntos para maximizar a pontuação total alcançada. Nesse contexto, o objetivo é determinar as melhores rotas para cada equipe, considerando não apenas a pontuação individual de cada participante, mas também a colaboração e coordenação entre eles. O TOP apresenta desafios adicionais em relação ao OP, uma vez que é necessário balancear as competências individuais dos participantes, bem como a interação entre as equipes, a fim de alcançar o melhor desempenho coletivo possível. Essa generalização do problema do Orienteering reflete situações reais nas quais a cooperação entre equipes é fundamental para o sucesso de uma missão ou tarefa. [Bouchakhchoukha e Akeb, 2014] modela o problema TOP como uma heurística híbrida para solucionar o desafio do trabalho em equipe, na Figura 3.7 temos um exemplo de solução.

Em *Dubins Orienteering Problem* (Problema de Orientação Dubins, DOP) [Pěnička et al., 2017] e *Minimal Exposure Dubins Orienteering Problem* (MEDOP) [Macharet et al., 2021], temos restrições extras de movimento. No DOP, o objetivo é otimizar as trajetórias para visitar pontos de interesse, considerando as restrições de movimento de um veículo com restrição de curvatura, conhecido como *Dubins car*. Isso significa que as trajetórias

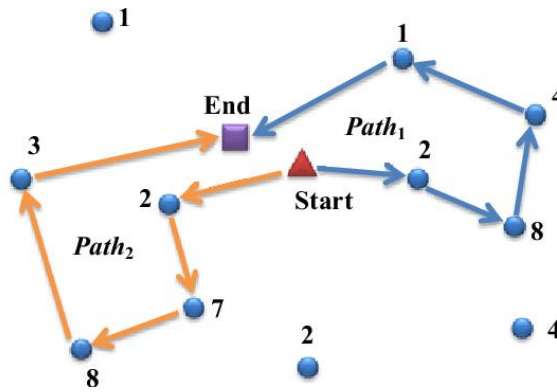


Figura 3.7: *Teams Orienteering Problem* (TOP) - Neste Exemplo dois veículos realizam duas rotas alcançando a recompensa associada a cada um dos vértices visitados por [Bouchakhchoukha e Akeb, 2014].

devem consistir apenas em curvas de curvatura limitada como na Figura 3.8. No caso do MEDOP considera-se não apenas o roteamento eficiente para visitar pontos de interesse, mas também busca minimizar a exposição a áreas perigosas ou indesejadas ao longo do caminho (Figura 3.9). Essas generalizações são aplicadas em cenários onde a restrição de curvatura e a minimização da exposição são fatores críticos, como planejamento de rotas para veículos autônomos, vigilância e monitoramento de áreas sensíveis.

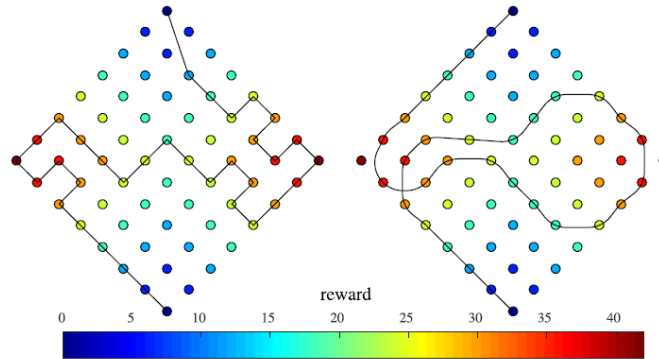


Figura 3.8: Comparação entre o OP e o DOP. À esquerda planejamento de caminho sem restrição de curvatura, à direita caminho com restrição de curvatura por [Pěnička et al., 2017].

No *Categorized Orienteering Problem with Count-Dependent Profits* (COPCDP) [Jandaghi et al., 2021], os clientes são categorizados, e a recompensa diminui à medida que se visita uma maior quantidade de nós da mesma categoria.

Em síntese, as variantes apresentadas, como o TOP, o DOP, o MEDOP e o COPCDP, ilustram a diversidade de extensões do OP, cada uma incorporando restrições ou objetivos adicionais que refletem cenários práticos mais complexos. Essas generalizações evidenciam a flexibilidade e relevância do modelo original, ao mesmo tempo em que reforçam os desafios computacionais associados à sua resolução. Assim, o estudo dessas variantes amplia não apenas o entendimento teórico sobre problemas de roteamento,

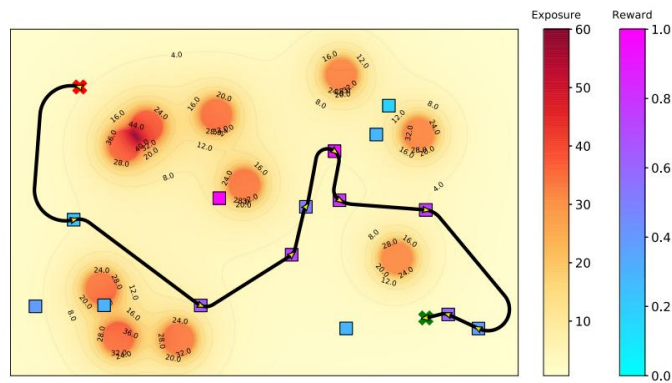


Figura 3.9: *Minimal Exposure Dubins Orienteering Problem* (MEDOP) - Regiões avermelhadas devem ser evitadas pelo veículo por [Macharet et al., 2021].

mas também seu potencial de aplicação em áreas estratégicas como transporte, vigilância, logística e planejamento de missões.

3.5 Aplicações práticas do OP e de suas variantes

Ao longo dos anos, o OP e suas variantes têm sido aplicados em uma ampla variedade de contextos, incluindo logística, transporte, turismo, coleta de dados e planejamento urbano. Uma revisão abrangente dessas aplicações é apresentada por [Gunawan et al., 2016], que classificam as principais extensões do problema e discutem suas adaptações em diferentes domínios. Com base nesse panorama, esta seção explora algumas dessas aplicações práticas, destacando os desafios enfrentados por pesquisadores e profissionais na busca por soluções eficientes.

Turismo e Planejamento de Roteiros

O OP encontra aplicações significativas no setor turístico, especialmente no planejamento de roteiros de visitação. Quando um turista dispõe de tempo limitado para explorar pontos de interesse em uma cidade ou região, o paradigma pode ser utilizado para determinar a sequência ideal de locais a visitar, maximizando o valor obtido, por exemplo, a atratividade, popularidade ou relevância cultural, sem ultrapassar restrições de tempo ou orçamento.

Diversas variantes do OP foram desenvolvidas para lidar com particularidades do turismo real, como múltiplos dias de viagem, janelas de tempo associadas ao horário de funcionamento dos locais, divisão geográfica em regiões compostas e transporte multimodal. Por exemplo, [Souffriau et al., 2008] aplicam o modelo clássico do OP para o

planejamento de visitas em um único dia e o *Teams Orienteering Problem* (TOP) para roteiros de múltiplos dias, permitindo ao turista maximizar a pontuação associada às atrações visitadas dentro das limitações temporais.

Em contextos urbanos, onde o transporte público desempenha papel central, o *Time Dependent Team Orienteering Problem with Time Windows* (TDTOPTW) proposto por [Gavalas et al., 2015] busca planejar roteiros turísticos que maximizem a satisfação do visitante enquanto minimizam o número de transferências e respeitam restrições de tempo e horários de funcionamento dos pontos de interesse. Já [Sylejmani et al., 2014] estendem o modelo *Multi-Constraint TOP with Multiple Time Windows* (MCTOPMTW) para lidar com múltiplas janelas de tempo, permitindo uma modelagem mais realista de cenários em que as atrações possuem horários variados de abertura e fechamento.

Outras abordagens tratam do turismo em larga escala e da personalização de experiências. [Herzog e Wörndl, 2014] consideram viagens compostas por múltiplas regiões, selecionando subconjuntos de áreas que maximizem o valor global da viagem dentro de restrições de tempo e orçamento. Em outro estudo, [Malucelli et al., 2015] propõem ajustes de roteiros que consideram preferências de diferentes perfis de usuários, buscando equilibrar atratividade e viabilidade logística. Complementarmente, [Yu et al., 2015] integram a maximização da coleta de informações com a minimização do tempo de deslocamento, aplicando um modelo com lucros variáveis em pontos turísticos de Istambul.

Essas aplicações evidenciam a versatilidade do OP como ferramenta de apoio ao turismo inteligente, permitindo a criação de roteiros personalizados, eficientes e compatíveis com as restrições práticas de tempo, transporte e preferências individuais dos viajantes.

Logística e Roteirização de Veículos

Uma das áreas mais recorrentes de aplicação do OP é a logística e a roteirização de veículos. No contexto de distribuição de mercadorias, por exemplo, empresas precisam otimizar as rotas de suas frotas de forma que a distância percorrida ou o tempo de viagem não ultrapassem determinados limites operacionais. O problema se torna particularmente relevante quando não é necessário visitar todos os pontos de entrega, mas é desejável maximizar o lucro ou o número de clientes atendidos dentro de restrições de tempo ou distância.

Diversas variações do OP têm sido propostas para capturar as complexidades desse tipo de cenário. O *Team Orienteering Arc Routing Problem* (TOARP), por exemplo, estende o problema para o contexto de roteamento em arcos, em que cada cliente está associado a um arco de um grafo direcionado e possui um lucro correspondente. O objetivo é selecionar quais arcos visitar, maximizando o lucro total coletado por uma frota de veículos, cada um com tempo de viagem limitado [Archetti et al., 2014].

Outras abordagens incorporam restrições práticas observadas em setores específicos. O *Multi-Period OP with Multiple Time Windows* (MuPOPTW), por exemplo, considera

múltiplos planejadores com janelas de tempo, aplicado ao planejamento de rotas de representantes de vendas do setor farmacêutico, onde há clientes obrigatórios e opcionais [Tricoire et al., 2010]. De modo semelhante, o *Stochastic OP with Time Windows* (SOPTW) é utilizado para definir quais médicos devem ser visitados e em qual ordem, equilibrando o tempo disponível e o potencial de retorno comercial [S. Zhang et al., 2014].

Além disso, há variações que integram o problema de roteamento com decisões de estoque, como o Problema de Roteamento de Inventário Cíclico com Veículo Único (SVCIRP), no qual o objetivo é simultaneamente minimizar os custos de distribuição e inventário, enquanto se maximiza a pontuação obtida pelos clientes visitados. Esse tipo de formulação busca determinar não apenas quais clientes atender, mas também a quantidade de produtos a ser entregue a cada um, respeitando as restrições de capacidade e evitando a falta de estoque [Vansteenwegen e Mateo, 2014].

Essas extensões ilustram a flexibilidade do OP e sua aplicabilidade em contextos reais de logística e transporte, nos quais o equilíbrio entre custos, tempo e benefícios coletados é essencial para a eficiência operacional.

Crowdsourcing Móvel

O *crowdsourcing móvel* representa uma aplicação emergente do OP, voltada à otimização da alocação de tarefas distribuídas entre usuários de dispositivos móveis. Nesse modelo, participantes voluntários podem realizar pequenos desvios em suas rotas cotidianas para executar tarefas em troca de recompensas, desde que respeitem restrições de tempo, localização e deslocamento. Essa abordagem tem se mostrado promissora em contextos urbanos dinâmicos, como coleta de conteúdo multimídia, inspeções, pequenas entregas e recomendações personalizadas de eventos.

Um dos primeiros trabalhos relevantes na área é o de [Ludwig et al., 2009], que propõem o sistema ROSE, um aplicativo móvel que combina recomendação de eventos, navegação a pé e integração com transporte público em tempo real. O problema de visitação de múltiplos destinos foi modelado como um OP, posteriormente estendido para o *Multi-Path OP with Time Windows* (MPOPTW), incorporando restrições de janelas de tempo e diferentes modos de transporte.

Avanços posteriores ampliaram o escopo do problema, considerando o uso do OP e suas variantes para modelar a coleta de informações em larga escala. [Liao e Hsu, 2013], por exemplo, desenvolveram um sistema de crowdsourcing voltado à coleta de conteúdo multimídia em locais e horários específicos, formulando o problema como uma versão generalizada do *Orienteering Problem with Time Windows* (OPTW), de modo a gerar rotas de desvio que maximizem o lucro obtido pelos usuários.

Em um contexto mais abrangente, [Chen et al., 2014] modelaram a execução de tarefas urbanas em grande escala como um TOP, onde os nós são classificados em obrigatórios (rotas rotineiras dos usuários) e opcionais (tarefas a serem realizadas). O objetivo

é maximizar a pontuação total obtida com as tarefas executadas, respeitando os limites de tempo disponíveis. Essa abordagem foi posteriormente aprimorada por [Cen et al., 2015], que introduziram um modelo probabilístico no qual cada usuário possui um conjunto finito de trajetos potenciais. Inspirado no *Multi-Period OP with Multiple Time Windows* (MuPOPTW), o método incorpora listas predefinidas de clientes obrigatórios e opcionais, considerando a sequência de visitação e as probabilidades associadas a cada trajeto.

Esses estudos demonstram como o OP pode ser adaptado para problemas modernos de mobilidade e colaboração distribuída, permitindo a coordenação eficiente de múltiplos participantes em ambientes urbanos dinâmicos e heterogêneos.

Outras Aplicações

O OP e suas variantes têm sido empregados em uma ampla gama de contextos práticos além destes citados, abrangendo desde o entretenimento e a gestão de visitantes até a coordenação de recursos em situações de emergência. Em geral, essas aplicações compartilham desafios relacionados a restrições de tempo, múltiplos agentes, disponibilidade limitada de recursos e incertezas associadas ao ambiente de operação.

No contexto de parques temáticos e grandes instalações de lazer, por exemplo, o problema tem sido utilizado para planejar trajetos otimizados de visitantes ou sistemas de navegação automatizados. [Lau et al., 2012] modelaram a navegação em parques temáticos como um *Set Orienteering Problem* (SOP), considerando fatores dinâmicos como tempos de fila e status de funcionamento das atrações. De forma complementar, [Gunawan et al., 2014] propuseram o uso do *Time Dependent Orienteering Problem* (TDOP) para o desenvolvimento de guias turísticos automáticos, em que o tempo de deslocamento entre dois pontos depende do momento em que a viagem é iniciada, tornando o modelo mais aderente às condições reais de fluxo e espera.

Outra linha de pesquisa explora o uso do OP em ambientes com múltiplos agentes e comportamento competitivo. [Varakantham et al., 2015] abordaram o problema de congestionamento em locais com grande circulação de pessoas, como museus, exposições e parques temáticos, por meio de uma formulação multiagente do OP. Nesse modelo, vários usuários egoístas percorrem simultaneamente o espaço, considerando as decisões e trajetórias dos demais, o que introduz elementos de interação e conflito típicos de cenários reais de mobilidade coletiva.

Em um contexto completamente distinto, [Van Der Merwe et al., 2014] aplicaram o *Cooperative OP with Time Windows* (COPTW) ao gerenciamento de recursos durante incêndios florestais. Nessa situação, é necessário alocar de forma eficiente caminhões e equipes de combate ao fogo para proteger ativos distribuídos em uma grande área, respeitando restrições severas de tempo e disponibilidade de recursos. O modelo cooperativo permite coordenar múltiplas unidades, maximizando o valor total das áreas protegidas.

Essas aplicações evidenciam a flexibilidade e o potencial do OP como ferramenta de apoio à decisão em ambientes complexos, dinâmicos e multiobjetivo, demonstrando sua utilidade em domínios que vão muito além da roteirização tradicional.

Conclusão

As aplicações do OP são vastas e diversificadas, refletindo a flexibilidade do modelo para se adaptar a diferentes tipos de restrições e objetivos. Em todas as áreas, os avanços nas técnicas de resolução, como algoritmos de otimização combinatória, aprendizado de máquina e inteligência artificial, têm permitido uma melhoria significativa na eficiência das soluções encontradas. À medida que novas aplicações surgem e mais dados se tornam disponíveis, o OP continuará a ser uma ferramenta valiosa para a otimização de processos em diversos setores.

3.6 Contextualização do trabalho

O OP é um problema clássico de otimização combinatória que combina elementos de roteamento e maximização de recompensas. Sua formulação básica consiste em selecionar um subconjunto de vértices a ser visitado, de modo a maximizar a soma das recompensas associadas, respeitando restrições de tempo ou distância. Graças a essa versatilidade, o OP tem sido amplamente estudado e aplicado em contextos como turismo, logística, coleta de dados, planejamento de missões robóticas e serviços móveis.

Entretanto, a formulação tradicional do OP assume que os vértices são independentes entre si, o que simplifica a modelagem, mas limita a capacidade de representar cenários mais realistas, nos quais tarefas, locais ou recursos apresentam algum tipo de relação estrutural. Para lidar com esse tipo de situação, foram propostas variantes que incorporam noções de agrupamento entre vértices, como o *Clustered Orienteering Problem* (COP) e o *Set Orienteering Problem* (SOP).

O COP introduz a ideia de clusters, exigindo que, para obter a recompensa, seja necessário visitar todos os vértices pertencentes a um cluster. Já o SOP define clusters em que apenas um vértice de cada conjunto pode ser escolhido. Embora representem avanços em relação ao OP clássico, essas variantes ainda possuem limitações importantes: o COP tende a ser excessivamente rígido, pois força a visita integral de clusters, enquanto o SOP restringe em excesso as escolhas, ao permitir apenas um vértice por conjunto.

Essas restrições tornam difícil modelar cenários mais ricos e realistas, como aqueles em que opções alternativas existem dentro de uma mesma região, ou em que diferentes

níveis de esforço e recompensa estão disponíveis em subconjuntos específicos de tarefas. Para superar essas limitações, propomos o *Clustered Orienteering Problem with Subgroups* (COPS), uma nova formulação que introduz uma estrutura hierárquica de agrupamento. Nela, vértices formam subgrupos e os subgrupos, por sua vez, compõem clusters. A regra central é que apenas um subgrupo pode ser selecionado por cluster, e a recompensa está associada ao subgrupo como um todo.

3.6.1 Relação do COPS com o COP

No problema COP, os vértices são agrupados em **clusters** e, para que um cluster seja considerado visitado, **todos os seus vértices** devem ser visitados. O objetivo é selecionar um subconjunto de clusters a serem completamente visitados, respeitando o orçamento de viagem, de modo a maximizar o lucro total.

O COPS generaliza o COP ao introduzir um nível intermediário de agrupamento. A seguir, são apresentados os principais pontos dessa generalização:

- Em vez de agrupar diretamente vértices em clusters, o COPS agrupa vértices em *subgrupos*, e os subgrupos em clusters.
- No COPS, **exatamente um subgrupo** deve ser escolhido por cluster, e todos os vértices desse subgrupo devem ser visitados, para se obter a recompensa associada ao subgrupo.
- O COP pode ser visto como um caso particular do COPS em que **cada cluster contém exatamente um subgrupo**, e esse subgrupo contém **todos os vértices** do cluster original, como ilustrado na Figura 3.10.

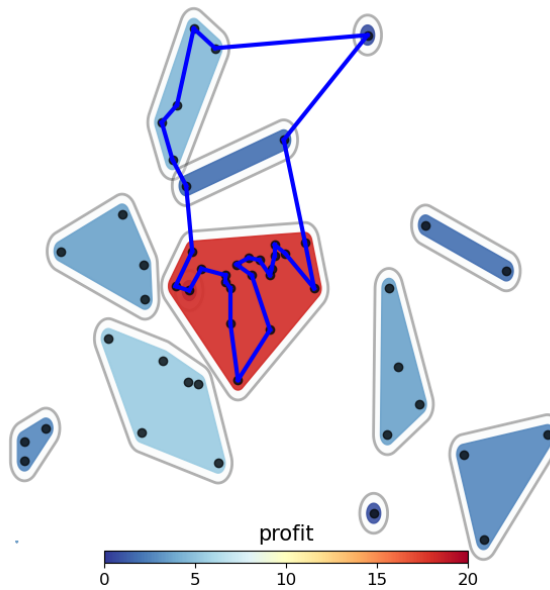


Figura 3.10: Exemplo do COP como caso particular do COPS. Neste exemplo, cada cluster é formado por somente um subgrupo contendo todos os vértices do cluster.

3.6.2 Relação do COPS com o SOP

No problema SOP, os vértices também estão agrupados em clusters, mas a regra é distinta: **exatamente um vértice** de um cluster precisa ser visitado para que o cluster seja considerado atendido.

O COPS também generaliza o SOP, como ilustrado na Figura 3.11. A seguir, são apresentados os principais pontos dessa generalização:

- Se cada subgrupo contém **apenas um vértice**, e os clusters são formados por conjuntos de tais subgrupos, então o COPS se reduz ao SOP.
- No COPS, ao selecionar um subgrupo (que, neste caso, equivale a um único vértice), satisfaz-se a condição de visitar exatamente um vértice por cluster, equivalente ao comportamento do SOP.

Note que, no caso do SOP, todos os vértices de um cluster possuem a mesma prioridade de atendimento, pois a visita a qualquer vértice gera a mesma recompensa. Por outro lado, no COPS, é possível estabelecer prioridades de escolha dentro do cluster, uma vez que a recompensa está associada a cada subgrupo de maneira individual.

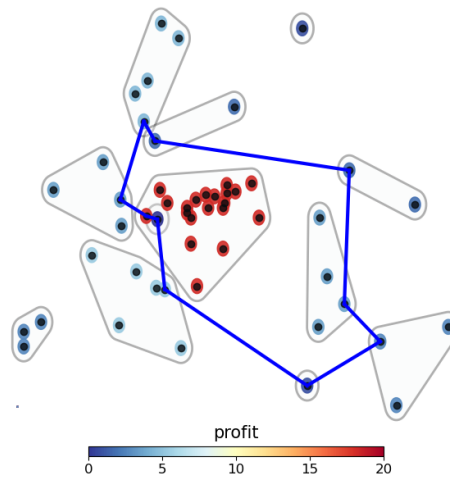


Figura 3.11: Exemplo do SOP como caso particular do COPS. Neste exemplo, todos os subgrupos são formados por um único vértice, e a rota da solução seleciona no máximo um subgrupo por cluster.

3.6.3 Resumo das Relações

A formulação proposta generaliza diferentes variantes do OP ao introduzir uma estrutura hierárquica em dois níveis: clusters e subgrupos. Essa generalização oferece maior flexibilidade, permitindo capturar heterogeneidade interna, modelar alternativas mutuamente exclusivas e equilibrar entre rigidez e liberdade na seleção de vértices.

A Tabela 3.1 resume as relações entre as variantes discutidas. É importante destacar que, na tabela, as variantes COP e SOP são apresentadas não em suas formulações originais, mas como casos particulares do modelo COPS obtidos por meio de restrições na estrutura hierárquica.

Assim, o COPS não apenas unifica características dessas variantes, como também amplia o espaço de modelagem e aplicação do *Orienteering Problem* em cenários práticos complexos.

Tabela 3.1: Relações estruturais entre o COPS e suas variantes particulares.

Variante (no contexto do COPS)	clusters contêm	Subgrupos contêm
COPS (geral)	múltiplos subgrupos	múltiplos vértices
COP (como caso particular)	1 subgrupo	múltiplos vértices
SOP (como caso particular)	múltiplos subgrupos	1 vértice cada

Capítulo 4

Propostas de Solução

4.1 Formulação em Programação Linear Inteira

Nesta seção, apresentamos a formulação de um modelo de ILP para resolver o COPS. O objetivo da ILP é otimizar a rota do veículo, maximizando a recompensa total coletada ao visitar subgrupos específicos em cada cluster, enquanto respeita as restrições de limite de orçamento $T_{\max}$. Este modelo oferece uma abordagem para lidar com a complexidade do COPS, permitindo o cálculo de soluções exatas. Ao construir este modelo, consideramos as relações entre clusters, subgrupos e vértices, além das limitações impostas pelo problema, para oferecer uma solução robusta.

Seja $\delta(U)$ o grupo de arestas com uma extremidade em U e a outra em $\mathcal{V} \setminus \{U\}$ (Para simplificar, usamos $\delta(i)$ em vez de $\delta(\{i\})$). Seja também $\mathcal{S}_j$ o conjunto de subgrupos que contêm o vértice j . Utilizaremos as seguintes variáveis de decisão:

- x_e variável binária, possui valor 1 se a aresta $e \in E$ for visitada; caso contrário, seu valor será igual a 0.
- y_j variável binária, possui valor 1 se o vértice $v_j \in \mathcal{V}$ for visitado; caso contrário, seu valor será igual a 0.
- z_s variável binária, possui valor 1 se todos os vértices do subgrupo $s \in \mathcal{S}$ forem visitados; caso contrário, seu valor será igual a 0.

A ILP que resolve o COPS com uma rota circular é a seguinte:

$$\max \sum_{s \in \mathcal{S}} r_s z_s, \quad (1)$$

sujeito às seguintes restrições:

$$y_1 = 1 \quad (2)$$

$$\sum_{e \in \delta(1)} x_e = 2 \quad (3)$$

$$\sum_{e \in \delta(j)} x_e = 2y_j \quad \forall j \in \mathcal{V}_{client} \quad (4)$$

$$\sum_{e \in E} t_e x_e \leq T_{\max} \quad (5)$$

$$\sum_{e \in \delta(U)} x_e \geq y_j \quad \forall j \in U, \forall U \subseteq \mathcal{V}_{client} \quad (6)$$

$$z_s \leq y_j \quad \forall j \in s, \forall s \in \mathcal{S} \quad (7)$$

$$\sum_{s \in c} z_s \leq 1 \quad \forall c \in \mathcal{C} \quad (8)$$

$$z_s \in \{0, 1\} \quad \forall s \in \mathcal{S} \quad (9)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (10)$$

$$y_j \in \{0, 1\} \quad \forall j \in \mathcal{V} \quad (11)$$

$$y_j \leq \sum_{s \in \mathcal{S}_j} z_s \quad \forall j \in \mathcal{V}_{client} \quad (12)$$

No caso de rota não circular, adicionar:

$$\sum_{j \in c_{end}} y_j = 1 \quad (13)$$

$$\sum_{e \in \delta(c_{end})} x_e = 1 \quad (14)$$

e substituir a Equação (3) por:

$$\sum_{e \in \delta(1)} x_e = 1 \quad (15)$$

Atente ao fato de que, a formulação matemática descrita aqui é muito similar à formulação do COP em [Angelelli et al., 2014], com as seguintes diferenças: neste trabalho foram acrescentados; os agrupamentos $S_j \in \mathcal{S}$ e as restrições 8 e 12.

Se os pontos de início e fim forem os mesmos (rota circular), as Restrições (2) e (3) garantem que a rota sai e retorna ao depósito. No caso de rotas não circulares, as Restrições (2) juntamente com (13)-(15) garantem que a rota comece no vértice inicial e termine em um dos vértices finais possíveis. Em [Angelelli et al., 2014], apenas rotas circulares são consideradas. As Restrições (4) garantem que, se um cliente for visitado, então duas arestas o alcancem (uma de chegada e uma de saída). As Restrições (5) garantem que o custo do trajeto não exceda o orçamento, e as Restrições (6) eliminam subtours.

As Restrições (7) garantem que a recompensa só seja coletada se todos os vértices do subgrupo correspondente forem visitados. As Restrições (8) garantem que, no máximo, um subgrupo por cluster seja visitado, e as Restrições (9)-(11) definem os limites das variáveis.

As Restrições (12) evitam rotas que contenham vértices adicionais que não agregam recompensa à solução. Elas permitem que um vértice seja visitado apenas se houver alguma recompensa associada a um dos subgrupos que o contêm. Essas restrições não são estritamente necessárias para a definição do problema, uma vez que sempre existirá uma solução ótima que visita apenas os vértices para os quais há recompensa. No entanto, como a função objetivo não penaliza rotas mais longas, a adição dessas restrições ajuda a eliminar soluções redundantes, tornando inviáveis as rotas que incluem vértices desnecessários. Vale notar que essa restrição é particularmente adequada em situações nas quais o menor caminho entre dois vértices é, de fato, o caminho direto entre eles, algo que, de forma geral, está associado a propriedades como a desigualdade triangular. Nessas condições, não há vantagem em passar por vértices intermediários, reforçando a utilidade das restrições ao evitar visitas desnecessárias.

Finalmente, a Função Objetivo (1) maximiza a recompensa obtida ao visitar os subgrupos.

Devido ao número exponencial de restrições, para resolver o ILP, adotamos um procedimento de *Branch-and-Cut* [Padberg e Rinaldi, 1987] para separar as restrições (6). Para soluções integrais candidatas, sejam $\bar{y}_j$ e $\bar{x}_e$ os valores assumidos pelas variáveis correspondentes y_j e x_e em uma relaxação de programação linear da ILP proposta. Seja $\bar{G} = (\bar{V}, \bar{E})$ um grafo induzido, onde $j \in \bar{V}$ se $\bar{y}_j = 1$ e $e \in \bar{E}$ se $\bar{x}_e = 1$. O algoritmo de separação consiste em verificar, por meio de uma busca em profundidade, se $\bar{G}$ é conectado; caso não seja, adicionamos uma restrição do tipo (6), onde U contém todos os vértices de $\bar{V}$ que estão desconectados do vértice 1.

Observe que também é possível verificar as restrições violadas para soluções fracionárias. Nesse caso, $j \in \bar{V}$ se $\bar{y}_j > \varepsilon$ e $e \in \bar{E}$ se $\bar{x}_e = \varepsilon$ (para um pequeno ε). Ao tornar $\bar{G}$ um grafo direcionado com arestas duplicadas $e \in \bar{E}$ com pesos $\bar{x}_e$ em ambas as direções, pode-se encontrar de forma polinomial o fluxo máximo entre 1 e todos os outros vértices em $\bar{V}$. Se a capacidade do corte mínimo entre 1 e o vértice atual for menor que 2, então o conjunto de vértices do outro lado do corte (em relação a 1) constitui uma restrição violada da equação (6). Esse algoritmo de separação expandido tem o potencial de melhorar a eficiência da formulação, mas deixamos isso como trabalho futuro.

4.2 Heurística baseada em Busca Tabu

A Busca Tabu (*Tabu Search*) [Glover, 1986] é uma técnica de otimização meta-heurística desenvolvida para resolver problemas complexos, principalmente os de natureza combinatória, em que métodos exatos tornam-se inviáveis devido ao alto custo computacional. Ela se diferencia de outras meta-heurísticas tradicionais por incorporar um mecanismo de memória que permite uma exploração mais eficaz do espaço de soluções, evitando que o processo de busca fique preso em ótimos locais, soluções aparentemente boas, mas que não são as melhores possíveis.

O funcionamento da Busca Tabu baseia-se na realização de movimentos no espaço de soluções, inclusive permitindo, em certos momentos, aceitar soluções piores do que a atual. Essa estratégia é importante para escapar de situações de estagnação e continuar avançando na busca por soluções mais promissoras. Para controlar essa liberdade, a técnica utiliza uma lista tabu, uma forma de memória que registra movimentos ou soluções recentemente visitadas e os proíbe temporariamente, evitando assim que o algoritmo volte a estados anteriores e entre em ciclos repetitivos.

Além disso, a Busca Tabu pode empregar critérios de aspiração, que permitem ignorar restrições da lista tabu quando a solução encontrada apresenta qualidade significativamente superior às anteriores. Essa flexibilidade contribui para a eficácia da técnica em diversos contextos. Por sua capacidade de avançar além dos ótimos locais e adaptar-se a diferentes tipos de problemas, a Busca Tabu é amplamente aplicada em áreas como o escalonamento de tarefas, a roteirização de veículos e o clássico problema do caixeiro-viajante, destacando-se como uma das meta-heurísticas mais robustas no campo da otimização.

Neste contexto, propomos o desenvolvimento de uma heurística baseada na meta-heurística Busca Tabu para resolver o problema de coleta de recompensas em subgrupos de vértices COPS. O objetivo é combinar as vantagens da Busca Tabu, que permite transcender de máximos locais, com a estrutura do problema, que envolve a escolha de clusters e subgrupos.

A heurística funciona iterativamente, explorando diferentes soluções vizinhas a partir de uma solução inicial. Durante o processo de busca, o mecanismo de lista tabu é aplicado para evitar revisitar soluções já exploradas e evitar ciclos de busca. Isso permite uma diversificação eficiente do espaço de busca, evitando que a heurística fique presa em soluções subótimas.

Além disso, a estratégia de intensificação é incorporada à heurística com o objetivo de aprofundar a exploração em torno de soluções promissoras. Essa intensificação se dá por meio da otimização de um critério adicional, neste caso, o custo total do caminho percorrido. Embora a minimização desse custo não seja uma exigência direta do *Orienteering*

Problem, ao incluí-lo na busca local, a heurística tende a favorecer regiões do espaço de soluções compostas por trajetos mais eficientes. Esses trajetos, por demandarem menos recursos, viabilizam a visita a um maior número de vértices, o que potencializa a coleta de recompensas. Dessa forma, mesmo sendo um critério secundário, a consideração do custo contribui indiretamente para uma aproximação mais eficaz do ótimo global.

Ao longo das iterações, a heurística avalia a qualidade das soluções vizinhas com base não apenas na função objetivo original, mas também nas restrições específicas do problema. A abordagem proposta, baseada em Busca Tabu, demonstra capacidade de encontrar soluções de boa qualidade, equilibrando a maximização das recompensas obtidas nos subgrupos visitados com a observância de limites de tempo ou distância percorrida. Por meio de uma exploração eficiente do espaço de busca e da diversificação promovida pela lista tabu, o algoritmo busca adaptar-se a diferentes contextos e exigências do problema, ampliando seu potencial de aplicação e robustez.

Algorithm 1 Busca Tabu para o COPS

```

1: Computar uma solução inicial  $\mathcal{P}0$ 
2:  $\mathcal{P} \leftarrow \mathcal{P}0$ 
3:  $\mathcal{P}^* \leftarrow \mathcal{P}0$ 
4: Inicializar a memória de longo prazo
5: Criar a memória de soluções inviáveis
6: while iterações sem melhoria  $\leq \tau$  do
7:   Atualizar a lista tabu
8:   Gerar um vizinho válido COPS  $\mathcal{P}'$  a partir de  $\mathcal{P}$ 
9:    $\mathcal{P} \leftarrow \mathcal{P}'$ 
10:  Pré-verificação de  $\mathcal{P}'$ 
11:  if  $\mathcal{P}'$  não está na memória de soluções inviáveis then
12:    Cálculo da TSP e verificação do custo de  $\mathcal{P}'$ 
13:  end if
14:  if  $\mathcal{P}'$  é melhor que  $\mathcal{P}^*$  then
15:     $\mathcal{P}^* \leftarrow \mathcal{P}'$ 
16:  end if
17:  Atualizar a memória de longo prazo
18:  Atualizar a memória de soluções inviáveis
19:  if caminho não circular não melhorado por  $T$  passos then
20:    Alterar o subgrupo final
21:  end if
22: end while
23: Return  $\mathcal{P}^*$ 

```

A abordagem proposta neste trabalho para resolver o COPS, conforme descrito no Algoritmo 1, inicia com uma solução inicial e explora iterativamente o espaço de soluções. A cada iteração, a busca avança para alguma solução na vizinhança atual. Para evitar ciclos e extrapolar de ótimos locais, o método utiliza uma lista tabu (*tabu list*), que armazena atributos de movimentos recentes (por exemplo, o subgrupo recém-adicionado

ou removido) e proíbe que esses movimentos sejam desfeitos por um número determinado de iterações. A Busca Tabu foi utilizada para resolver o problema COP em [Angelelli et al., 2014], e nossa abordagem é inspirada nesse trabalho. No entanto, no contexto do COPS, é fundamental considerar que, uma vez que um subgrupo é incorporado à solução, todos os demais subgrupos pertencentes ao mesmo cluster tornam-se inelegíveis para inserção. Assim, para gerar um vizinho plausível, nossa abordagem consiste em buscar um subgrupo que não pertença a nenhum dos clusters já presentes na solução atual.

Solução inicial $\mathcal{P}_0$

Para a solução inicial $\mathcal{P}_0$, escolhemos aleatoriamente um cluster por vez e, para cada cluster, selecionamos o subgrupo com o maior nível de rentabilidade. O nível de rentabilidade é definido aqui como o lucro obtido ao visitar o subgrupo dividido pelo número de vértices desse subgrupo. Em seguida, esse subgrupo é inserido em $\mathcal{P}_0$, e um novo percurso é gerado utilizando a estratégia 2-opt [Croes, 1958] com os vértices pertencentes a todos os subgrupos da solução atual. O procedimento é encerrado quando todos os clusters tiverem sido considerados ou quando o percurso ultrapassar o orçamento por 10 iterações consecutivas.

Memória de longo prazo

Esta é uma memória, nomeada aqui por η_s , que armazena o número de iterações que um subgrupo s permaneceu dentro ou fora da solução atual $\mathcal{P}$. Ela será definida como um quando s for inserido em $\mathcal{P}$, e será definida como zero toda vez que for removido de $\mathcal{P}$; a partir daí, ela aumentará ou diminuirá respectivamente a cada iteração.

Memória de soluções inviáveis

Semelhante a [Angelelli et al., 2014], a memória de soluções inviáveis armazena soluções que violam o orçamento. Cada vez que um caminho $\mathcal{P}'$ é calculado sobre um conjunto $\tilde{\mathcal{S}}$ de subgrupos, e este caminho $\mathcal{P}'$ for considerado inviável, essa informação é armazenada em memória. Assim, se o conjunto $\tilde{\mathcal{S}}$ for inviável, ou seja, se $\mathcal{P}' = TSP(\tilde{\mathcal{S}}) \geq T_{\max}$, o COPS-TABU não calculará novamente o TSP para $\tilde{\mathcal{S}}$ nem para qualquer superconjunto $\tilde{\mathcal{S}}' \supseteq \tilde{\mathcal{S}}$ nas próximas iterações.

Note que, no COPS-TABU, o valor de $TSP(\tilde{\mathcal{S}})$ é calculado por meio do algoritmo de 2-opt [Croes, 1958] e, portanto, trata-se de um valor heurístico. Isso significa que o COPS-TABU pode descartar soluções viáveis.

Condição de parada

O algoritmo para após τ iterações sem aumentar a recompensa da melhor solução, ou sem diminuir a distância percorrida, mantendo a mesma recompensa. O valor de τ obtido

após testes preliminares foi $\tau = 3k$, sendo k a quantidade de subgrupos.

Lista tabu

Um subgrupo será considerado *tabu* (proibido) para inserção ou remoção se, respectivamente, $\eta_s > -\alpha$ e $\eta_s < \alpha$.

O valor de α é sempre mantido dentro do intervalo $1 \leq \alpha \leq \lfloor \frac{1}{3} \times k \rfloor$, onde k é o número de subgrupos. Cada vez que encontramos uma nova melhor solução, o valor de α é reduzido em 25% do seu valor atual. Por outro lado, se 10 iterações se passarem sem melhorar a solução, o valor de α é aumentado em 1. Este parâmetro segue a mesma definição e estratégia de ajuste apresentada por Angelelli et al. [2014].

A razão por trás dessa regra de atualização do valor de α é que, quando uma nova melhor solução é encontrada, queremos explorar profundamente o espaço de soluções ao redor dessa solução, portanto, precisamos reduzir drasticamente o valor de α para evitar perder boas soluções por serem classificadas como tabu.

Por outro lado, quando a solução não é melhorada, queremos evitar retornar à solução atual e, por isso, aumentamos o valor de α . No entanto, para evitar restringir demais o conjunto de soluções não-tabus, α é aumentado gradualmente por um valor constante igual a 1

Gerar um vizinho válido do COPS $\mathcal{P}'$

A cada iteração, será gerado um grupo de vizinhos da solução atual $\mathcal{P}$ em uma sequência predefinida. Para evitar resolver diversas instâncias do TSP e, só então, escolher o melhor movimento, a primeira solução plausível encontrada deve ser considerada como o vizinho $\mathcal{P}'$. Em seguida, esse vizinho é comparado com a melhor solução $\mathcal{P}^*$. Ele substituirá a melhor solução caso seja mais lucrativo, ou apresente a mesma recompensa com um custo de percurso menor. Note que, nesse caso, a escolha privilegia caminhos mais eficientes quando a recompensa é a mesma, mas o custo é menor (critério secundário de intensificação).

As regras para geração de vizinhos seguem duas estratégias:

- Inserção de um subgrupo cujo cluster associado atualmente não possui outro subgrupo em $\mathcal{P}$. Essa inserção é realizada em duas etapas:
 - Inicialmente uma estratégia gulosa simples. Para cada vértice a ser inserido, o algoritmo identifica o vértice mais próximo já presente na rota atual, com base na matriz de distâncias pré-computada, e insere o novo vértice imediatamente após ele. Essa abordagem preserva, tanto quanto possível, a estrutura existente da rota, permitindo soluções compactas e eficientes;

- Após todas as inserções, é aplicada uma busca local 2-opt [Croes, 1958] para refinar a rota.
- Remoção de um subgrupo da solução atual. A operação de remoção é ainda otimizada pela eliminação direta dos vértices do subgrupo da solução, conectando seu predecessor ao sucessor. No entanto, deve-se tomar cuidado para não remover qualquer vértice que seja compartilhado com outro subgrupo ainda presente na solução atual $\mathcal{P}$.

Os vizinhos são gerados na seguinte sequência:

Inserção Não-Tabu Escolhe-se aleatoriamente, a partir de $\mathcal{S}_{client}$, um subgrupo que não esteja na lista tabu e que não pertença a um agrupamento já presente na solução atual $\mathcal{P}$, e insere-se esse subgrupo no vizinho $\mathcal{P}'$.

Remoção Antiga Remove-se de $\mathcal{P}$ um subgrupo escolhido aleatoriamente de $\mathcal{S}_{client}$ tal que $\eta_s > \beta$. Onde, $\beta = \frac{\tau}{10}$, tal qual a definição e estratégia de ajuste apresentada por Angelelli et al. [2014].

Inserção Tabu Insere-se um subgrupo da lista tabu que não pertença a nenhum agrupamento atendido na solução atual. O subgrupo escolhido é aquele com o maior nível de aspiração. De forma semelhante a [Angelelli et al., 2014], o nível de aspiração é o maior valor de recompensa obtido em qualquer solução que tenha contido esse subgrupo.

Inserção Aleatória Escolhe-se e insere-se aleatoriamente qualquer subgrupo de $\mathcal{S}_{client}$ que não pertença a algum agrupamento atendido na solução atual.

Remoção Não-Tabu Remove-se de $\mathcal{P}$ um subgrupo escolhido aleatoriamente de $\mathcal{S}_{client}$ que não esteja em estado tabu para remoção.

Remoção Aleatória Escolhe-se aleatoriamente um subgrupo de clientes da solução atual e remove-se.

Pré-verificação e verificação do vizinho $\mathcal{P}'$

A incorporação de um procedimento de pré-verificação aumenta significativamente a eficiência do algoritmo. Se algum elemento da memória de soluções inviáveis for um subconjunto de $\mathcal{P}'$, ele pode ser rapidamente classificado como inviável sem a necessidade de realizar cálculos da rota.

Se $\mathcal{P}'$ não corresponder a nenhuma solução pré-verificada, isso indica uma nova configuração, que requer a verificação do comprimento da rota. Os vértices de $\mathcal{P}'$ são então processados pelo algoritmo 2-opt para avaliar a viabilidade da rota.

Essa pré-verificação foi projetada para ser mais eficaz em cenários onde a re-otimização do percurso com 2-opt é o principal custo computacional e muitos vizinhos gerados compartilham subcaminhos inviáveis previamente identificados, evitando assim cálculos custosos.

Caminhos não circulares

Se o ponto inicial for diferente do ponto final do caminho, então haverá um cluster final, e todos os subgrupos finais devem conter apenas um vértice, e o solucionador escolherá qual subgrupo será o depósito final.

Nesse caso, o ponto final será alterado após T iterações sem melhoria. Para isso, escolhemos o subgrupo com a memória de longo prazo mínima, ou seja, o subgrupo que ficou mais tempo fora da solução. T é definido como β dividido pelo número de subgrupos finais. A ideia é que a vizinhanças de todos os pontos finais sejam testadas antes do término da execução.

Capítulo 5

Resultados

Este capítulo apresenta os resultados computacionais obtidos para a avaliação da variante proposta, o COPS. O objetivo é analisar o comportamento do modelo em diferentes cenários, desde exemplos simples e ilustrativos até instâncias de maior complexidade, permitindo verificar sua capacidade de capturar as particularidades da formulação e de gerar soluções de qualidade.

Os experimentos foram organizados em diferentes etapas, de modo a oferecer uma análise clara e objetiva:

- Seção 5.1 – Exemplo ilustrativo simples: um caso reduzido é apresentado para demonstrar, de forma clara e visual, a formação de subgrupos e clusters, bem como as restrições específicas do COPS.
- Seção 5.2 – Conjunto de benchmarks: nesta etapa são utilizados dois tipos de instâncias como casos particulares do COPS:
 - instâncias do COP disponíveis em <https://or-brescia.unibs.it/>;
 - instâncias do SOP disponibilizadas em <https://github.com/ctu-mrs/vns-sop>;
- Seção 5.3 - Instâncias próprias, desenvolvidas com o objetivo de explorar características específicas do COPS, tais como a hierarquia dos subgrupos, casos 3D e rotas não circulares com múltiplos vértices de saída.

Todas as instâncias e algoritmos utilizadas nos experimentos encontram-se reunidas no repositório público <https://github.com/verlab/COPS>, de forma a garantir a reprodutibilidade e facilitar a comparação com trabalhos futuros. Os algoritmos foram implementados em Python 3.10, e a ILP utiliza o Gurobi¹ 10.0.0 com licença acadêmica como solucionador. Todos os testes foram realizados em uma máquina com processador Intel Xeon E5-2630 v3 de 2,4 GHz e 128 GB de RAM.

A análise dos resultados considera tanto a qualidade das soluções quanto o tempo computacional requerido, destacando aspectos em que a nova variante se aproxima ou se diferencia dos problemas já estabelecidos na literatura.

¹<https://www.gurobi.com/>

5.1 Exemplo Ilustrativo do COPS

Nesta seção, apresentamos um exemplo ilustrativo para facilitar a compreensão do funcionamento do COPS, destacando suas regras e particularidades. A Figura 5.1 exibe uma instância do problema, incluindo a estrutura hierárquica de clusters e subgrupos, bem como de uma rota válida como solução do problema.

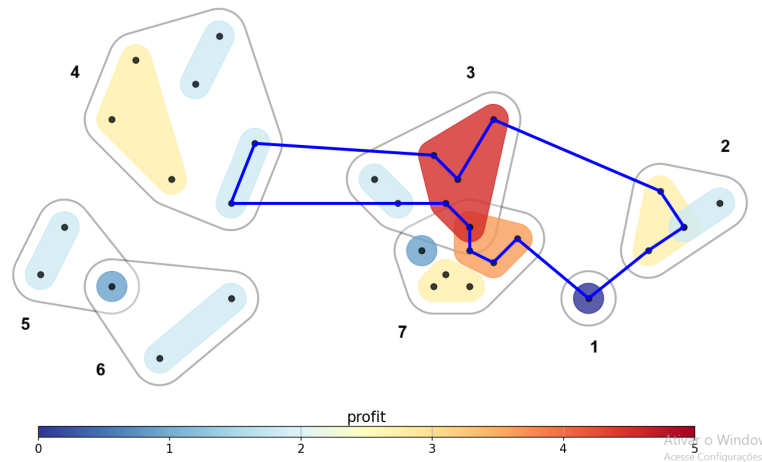


Figura 5.1: Exemplo ilustrativo de uma instância do problema COPS.

Na figura, é possível observar:

- O cluster 1 representa o ponto de partida da rota. Ele é composto por apenas um subgrupo que possui um único vértice.
- Há múltiplos **clusters** (indicados por contornos), cada um contendo um ou mais **subgrupos** (indicados pelos agrupamentos coloridos). Cada subgrupo contém um ou mais vértices.
- A cor de cada subgrupo representa a recompensa associada a ele, que neste exemplo corresponde ao número de vértices do subgrupo. Essa associação entre cor e valor é indicada pela barra de cores na parte inferior da Figura 5.1.
- Observe que o cluster 2 possui 2 subgrupos que compartilham 1 vértice em comum.
- Os clusters 3 e 7 também compartilham um vértice.
- Os clusters 5 e 6 compartilham um subgrupo com um vértice.
- O objetivo é escolher no máximo **um subgrupo por cluster**, e visitar todos os seus vértices. Apenas os subgrupos completamente visitados geram recompensa.
- No exemplo mostrado, a rota destacada em azul percorre vértices de diferentes subgrupos, respeitando a regra de selecionar no máximo um subgrupo por cluster.

- Neste exemplo a rota é circular, ou seja, o ponto final é igual ao ponto de partida.
- Observa-se que os vértices de um mesmo subgrupo não precisam ser visitados sequencialmente na rota (ver, por exemplo, o cluster 3), o que confere flexibilidade à construção do caminho.

Esse exemplo destaca a natureza hierárquica do problema, onde:

- A seleção de subgrupos está subordinada à restrição de um por cluster;
- A coleta de lucro depende da visita completa ao subgrupo;
- A ordem de visita dos vértices pode ser arbitrária, desde que o custo total da rota não ultrapasse o orçamento $T_{\max}$.

5.2 Comparação com Cenários Clássicos do COP e SOP

O COPS é uma formulação com alta capacidade de adaptação, capaz de abordar uma variedade de cenários ao se ajustar aos requisitos específicos de cada problema. Dessa forma, o COPS demonstra a capacidade de resolver diversos problemas clássicos de orientação de maneira unificada. Nesta seção, propomos utilizar a modelagem do COPS para resolver dois problemas clássicos: o COP e o SOP.

5.2.1 COPS resolvendo cenários SOP

Primeiramente adaptamos as instâncias SOP introduzidas por [Archetti et al., 2018] para instâncias COPS.

Essas instâncias originais foram formadas a partir de instâncias do problema do Caixeiro Viajante Generalizado (GTSP), propostas por [Fischetti et al., 1997]. Essas instâncias possuem entre 52 e 1084 vértices e uma quantidade de clusters aproximadamente 20% do número de vértices. O limite orçamentário ($T_{\max}$) de cada instância foi definido como uma fração ω do custo ótimo conhecido do GTSP, sendo usados $\omega = 0.4$, 0.6 e 0.8 para instâncias pequenas de até 76 vértices e $\omega = 1.0$ para as grandes. Os custos das arestas foram arredondados para cima (CEIL_2D), e os lucros dos clusters foram

definidos de duas formas: 1° proporcional ao número de nós do cluster, e, 2° com lucros pseudoaleatórios. Na Tabela 5.1 diferenciamos o segundo tipo de lucro através do sufixo (p2)

Tabela 5.1: Comparação dos resultados do SOP original MASOP [Archetti et al., 2018] e VNS-SOP [Pěnička et al., 2019], com as nossa formulações em ILP e COPS-TABU. A análise considera a Recompensa (R) e Tempo de execução (T) em segundos.

Instance	Size	$T_{\max}$	MASOP		VNS-SOP		ILP		COPS-TABU	
			R	T	R	T	R	T	R	T
11berlin52	52	1616	37	1.75	37	0.11	37	0.53	37	0.21
11berlin52	52	2424	43	2.4	43	0.16	43	2.1	43	1.2
11berlin52	52	3232	47	4.63	47	0.19	47	1.3	47	0.91
11eil51	51	69	24	1.85	24	0.09	24	0.8	28	1.2
11eil51	51	104	39	5.13	39	0.14	39	1.2	39	1.8
11eil51	51	139	43	2.30	43	0.18	43	1.43	43	1.1
14st70	70	126	33	4.4	33	0.14	33	10.4	33	5.9
14st70	70	252	65	8.8	5	0.31	65	535.1	65	3.3
16pr76	76	64925	74	8.6	74	0.6	—	—	74	1.6
16pr76p2	76	64925	3765	10.6	3765	0.6	—	—	3765	2.1
20kroA100	100	9711	96	10.5	96	0.8	—	—	96	36.8
20kroA100p2	100	9711	4868	11.5	4868	0.8	—	—	4868	33.9
20kroB100	100	10328	98	13.4	98	0.8	—	—	98	34.7
20kroB100p2	100	10328	4916	10.7	4916	0.9	—	—	4916	42.1
20kroD100	100	9450	96	10.3	96	1	—	—	96	41.7
20kroE100	100	9523	96	9.1	96	0.9	—	—	96	20.1
20rd100	100	3650	97	10.4	97	1.3	—	—	97	42.6
22pr107	107	27898	101	8.6	101	0.6	—	—	101	114
26bier127	127	72418	125	16	125	1.8	—	—	125	96.0
26bier127p2	127	72418	6314	16.2	6314	1.8	—	—	6314	100
29pr144	144	45886	141	17.4	141	1.7	—	—	141	185
30kroB150	150	12196	148	15.2	148	2.5	—	—	148	251
46pr226	226	64007	224	26.9	224	3.6	—	—	224	207
53pr264	264	29549	262	34	262	7.1	—	—	262	420
84fl417	417	9651	415	103.5	415	18	—	—	415	369
131p654	654	27428	652	356	652	45.9	—	—	652	611
131p654p2	654	27428	32991	360.1	32950	44.2	—	—	32981	713
201pr1002	1002	114311	994	304.9	994	480.4	—	—	984	2544
217vm1084	1084	130704	1078	498.5	1070	832.0	—	—	1072	2833

Para modelar o SOP dentro da formulação COPS, cada vértice da instância original é inserido em um subgrupo, mantendo a recompensa original associada ao subgrupo. Esses subgrupos, por sua vez, são organizados em clusters, conforme a estrutura original. A Figura 5.2 ilustra uma instância típica de SOP, conforme proposta no trabalho de [Archetti et al., 2018], e sua adaptação à modelagem COPS. Na mesma figura, mostramos a rota gerada pelo nosso algoritmo de busca Tabu (COPS-TABU).

Uma das principais vantagens da abordagem COPS em relação à formulação original do SOP é a flexibilidade na definição dos lucros. Enquanto no SOP clássico os

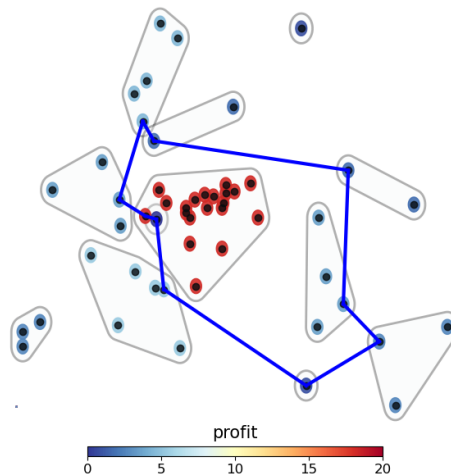


Figura 5.2: SOP formato - pontos pretos representam os vértices, áreas coloridas representam os subgrupos (a cor depende da recompensa correspondente) e linha cinza representa o cluster. Neste caso, cada subgrupo contém apenas um vértice. Como o COPS escolhe apenas um subgrupo em cada cluster, alcançamos um resultado idêntico ao SOP. A linha em azul é o caminho encontrado pelo COPS-TABU.

vértices de um mesmo cluster possuem lucros idênticos, a formulação COPS permite a atribuição de lucros variados aos vértices, criando instâncias SOP mais realistas onde se deve visitar apenas um vértice dentro de um grupo, porém alguns destes vértices podem ser prioritários.

Na Tabela 5.1, comparamos a recompensa total (R) e runtime (T) para o Tabu Search MASOP proposto por [Archetti et al., 2018], o VNS-SOP proposto por [Pěnička et al., 2019] e pela nossa abordagem COPS-TABU. Cabe ressaltar que os resultados dos métodos MASOP e VNS-SOP foram extraídos diretamente das publicações originais, e, portanto, as condições experimentais não são idênticas às do presente estudo.

Um teste t pareado, com 95% de confiança, revela que o algoritmo COPS-TABU é competitivo, alcançando soluções, em termo de recompensa (R), de qualidade semelhante às dos métodos clássicos, embora com um tempo de execução superior ao do MASOP e do VNS-SOP. A ILP é capaz de resolver as instâncias pequenas, mas, para instâncias maiores (acima de 76 vértices), o tempo de execução elevado torna sua utilização inviável. Nos nossos testes, interrompemos a execução da ILP após 30 minutos.

5.2.2 COPS resolvendo cenários COP

Neste experimento adaptamos o dataset com instâncias COP utilizado por [Angeli et al., 2014] para executarmos como COPS.

Quando adaptamos o COPS para o COP, a modificação é simples: definimos um subgrupo que agrupa todos os vértices de um cluster. Nesse cenário, a modelagem COPS mantém a estrutura do COP clássico, onde todos os vértices de um cluster precisa ser visitado. A Figura 5.3 mostra a mesma instância utilizada na Figura 5.2, mas com a organização de um subgrupo por cluster, representando uma adaptação fiel ao COP clássico. O caminho gerado pelo algoritmo COPS-TABU visita todos os vértices do subgrupo.

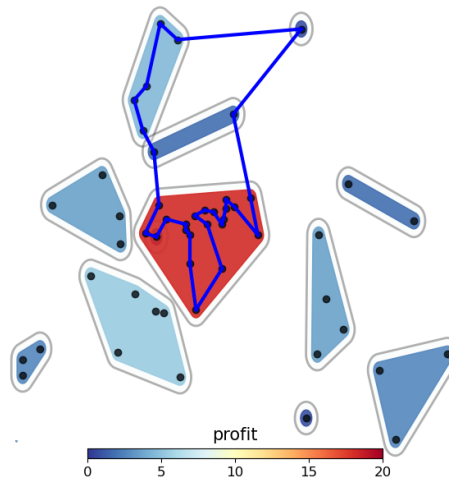


Figura 5.3: COP formato - pontos pretos representam os vértices, áreas coloridas representam os subgrupos (a cor depende da recompensa correspondente) e linha cinza representa o cluster. Neste caso, cada cluster contém apenas um subgrupo com vários vértices. Já que o COPS visita todos os vértices dentro de um subgrupo, conseguimos assim um resultado idêntico ao COP. A linha em azul é o caminho encontrado pelo COPS-TABU.

O dataset original foi gerado a partir de instâncias de benchmark do TSP provenientes da biblioteca TSPLIB95, disponível na seguinte URL: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95>. Estas instâncias possuem tamanhos variando de 42 a 532 vértices. Para formar os clusters de clientes, foram utilizados diferentes números de clusters (10, 15, 20 e 25), com sobreposições parciais entre eles, usando uma abordagem sequencial para balancear tamanhos. Os lucros dos clientes foram atribuídos de duas formas: uma constante igual a 1 ou um valor pseudo-aleatório calculado por uma fórmula específica. Além disso, o limite de tempo máximo ($T_{\max}$) foi definido como frações de $1/2$ e $3/4$ do valor ótimo do TSP correspondente a cada instância, permitindo uma análise diversificada da dificuldade e do desempenho dos algoritmos em diferentes configurações.

Na Tabela 5.2 temos uma comparação entre os resultados encontrados no trabalho [Angelesli et al., 2014] executadas com o algoritmo COPS-TABU. Cabe ressaltar que os resultados do método COP-TABU foram extraídos diretamente da publicação original, e, portanto, as condições experimentais não são idênticas às do presente estudo. Os nomes das instâncias na tabela seguem uma convenção que codifica essas características. Por exemplo, na instância berlin52s10g1q2, o prefixo berlin52 indica o nome original da

instância na TSPLIB95, s10 representa a quantidade de clusters (neste caso, 10), g1 indica que o lucro de cada vértice é igual a 1 (enquanto g2 indicaria lucros aleatórios), e q2 refere-se a um $T_{\max}$ correspondente a 1/2 do valor ótimo do TSP (sendo q3 utilizado para 3/4 do valor ótimo).

Tabela 5.2: Comparação dos resultados do COP original COP-TABU [Angelelli et al., 2014], com as nossa formulações em ILP e COPS-TABU. A análise considera a Recompensa (R) e Tempo de execução (T) em segundos.

Instance	Size	$T_{\max}$	COP-TABU		ILP		COPS-TABU	
			R	T	R	T	R	T
eil51s25g1q2	51	229	32	94.64	32	1.2	32	50.3
berlin52s10g1q2	52	3785	21	2.69	21	2.1	21	5.4
st70s10g2q2	70	355	999	2.39	999	6.7	999	5.6
eil76s10g2q2	76	292	1490	1	1726	11.5	1726	13.1
eil76s15g2q2	76	292	1511	5	1994	8.5	1670	77
pr76s10g2q2	76	54096	2362	3	2362	168.4	2362	18.5
kroA100s25g1q2	100	10672	30	43	—	—	30	27.6
kroB100s25g1q3	100	16638	95	175	—	—	95	146.7
kroC100s25g1q2	100	10401	24	7	—	—	24	32.7
kroE100s20g2q3	100	16591	4373	103.1	—	—	4298	516
kroE100s25g1q2	100	11061	36	118.07	—	—	30	60.4
rd100s20g1q2	100	3979	35	12	—	—	35	108
rd100s25g1q2	100	3979	42	193.65	—	—	42	149.7
bier127s15g2q2	127	59176	4245	28	—	—	4315	160
ch130s15g1q3	130	4632	88	13.3	—	—	87	165
ch150s15g1q2	150	3304	36	7	—	—	36	39.3
kroA150s10g2q3	150	19956	4360	7	—	—	4379	99.0
kroB150s15g2q3	150	19655	5421	29	—	—	5421	167
u159s10g1q2	159	21053	72	35.91	—	—	71	21.6
d198s10g1q2	198	7950	110	17	—	—	110	20.6
kroA200s10g1q2	200	14739	44	2	—	—	44	9.5
kroB200s25g1q3	200	22161	139	287	—	—	139	279
pr226s10g1q2	226	40195	81	33.62	—	—	81	21
pr226s10g2q3	226	60292	9632	21.04	—	—	9633	104.1
gil262s10g1q2	262	1261	61	8.61	—	—	61	23
a280s10g1q2	280	1306	128	32.26	—	—	128	61
rd400s10g1q2	400	7738	92	12.51	—	—	92	33
fl417s10g2q3	417	9089	17247	41	—	—	17401	76.2
fl417s15g1q3	417	9089	380	255.6	—	—	349	320.4
d493s10g1q2	493	17615	285	73.33	—	—	285	45
d493s10g1q3	493	26423	400	69	—	—	401	160

Com base nos resultados do teste t pareado com 95% de confiança, observamos que o algoritmo COPS-TABU alcança recompensas (R) semelhantes às do algoritmo original COP-TABU. No entanto, o tempo de execução do COPS-TABU é superior ao do MASOP. A ILP é capaz de resolver de forma ótima as instâncias pequenas, mas para instâncias

grandes o tempo de solução excedeu o limite imposto nos experimentos de 30 minutos de execução para a ILP.

5.2.3 Análise dos Experimentos

Os resultados indicam que o desempenho do COPS-TABU em termos de recompensa (R) alcançada é semelhante ao dos métodos já estabelecidos para resolver tanto o COP quanto o SOP. Em relação ao tempo computacional, pode-se observar que a nossa abordagem obtém resultados piores quando comparada aos benchmarks, especialmente para instâncias de maior porte.

Esses resultados são altamente relevantes, pois sugerem que o COPS oferece uma modelagem eficiente para ambos os problemas, sem a necessidade de algoritmos diferentes para cada um. Além disso, esses resultados incentivam futuras investigações sobre outras generalizações que podem ser modeladas como o COPS como, por exemplo, o *Dubins Orienteering Problem* (Problema de Orientação Dubins, DOP) e o *Orienteering Problem with Neighborhoods* (OPN), já que foi demonstrado em [Pěnička et al., 2019] que estes problemas podem ser modelados como o SOP.

5.3 Casos de Estudo em Cenários Próprios

O COPS se apresenta como uma variação interessante que adiciona complexidade aos problemas de clusterização do *Orienteering Problem*. Ao incluir os subgrupos e clusters, o COPS cria uma estrutura hierárquica que pode ser utilizada para criar diversos problemas que até então não foram investigados na literatura científica. Essa flexibilidade permite aplicar o COPS em uma ampla gama de contextos reais. Nesta seção investigaremos algumas destas novas aplicações.

5.3.1 Estudo de Caso — Caso simples para demonstrar a escolha dos subgrupos mais valiosos com orçamento $T_{\max}$ variável.

Este estudo de caso apresenta um exemplo simplificado com o objetivo de demonstrar a funcionalidade fundamental do COPS, que consiste na seleção dos subgrupos mais vantajosos de cada cluster, de acordo com as restrições impostas pelo orçamento disponível. Para tal, o orçamento $T_{\max}$ destinado ao deslocamento será variado, de modo a ilustrar as diferentes configurações e comportamentos do modelo.

O cenário considerado envolve dois clusters clientes, cada um composto por dois subgrupos (Figura 5.4). A visita aos subgrupos de maior relevância implica percursos mais extensos, porém com maior retorno associado. O propósito é evidenciar que a escolha estratégica dos subgrupos dentro de cada cluster pode maximizar o ganho total, desde que respeitadas as limitações orçamentárias.

Em determinadas situações, optar por subgrupos de menor rentabilidade pode representar uma estratégia mais eficiente, pois possibilita a visita a um número maior de clusters, resultando em um benefício agregado superior, isto pode ser visto, por exemplo, na Figura 5.4b, com $T_{\max} = 30$ o algoritmo decide visitar os dois subgrupos menores pois visitar um subgrupo maior implicaria em não ter mais orçamento para visitar o outro cluster, neste caso visitar ambos subgrupos menores alcança recompensa $R = 6$ e visitar apenas um subgrupo maior geraria $R = 4$ apenas. Já com $T_{\max} = 35$ é possível visitar um subgrupo maior e sobrar orçamento para visitar o outro cluster, veja na Figura 5.4c. A Tabela 5.3 mostra os comprimentos (L) de cada rota.

A Tabela 5.3 também mostra os resultados discriminados para o algoritmo ILP e para a heurística COPS-TABU. Observe que para este exemplo a recompensa (R) alcançada pela heurística é sempre igual a ótima encontrada pela ILP.

Uma característica relevante do *Orienteering Problem* pode ser observada neste exemplo: ao analisar o comprimento do caminho (L) obtido pelo ILP e pela heurística, nota-se que estes valores podem divergir mesmo sob o mesmo orçamento disponível ($T_{\max}$). Tal comportamento ocorre porque o objetivo principal do OP é a maximização da recompensa total, não havendo, necessariamente, uma priorização explícita pela minimização do comprimento do trajeto para alcançá-la.

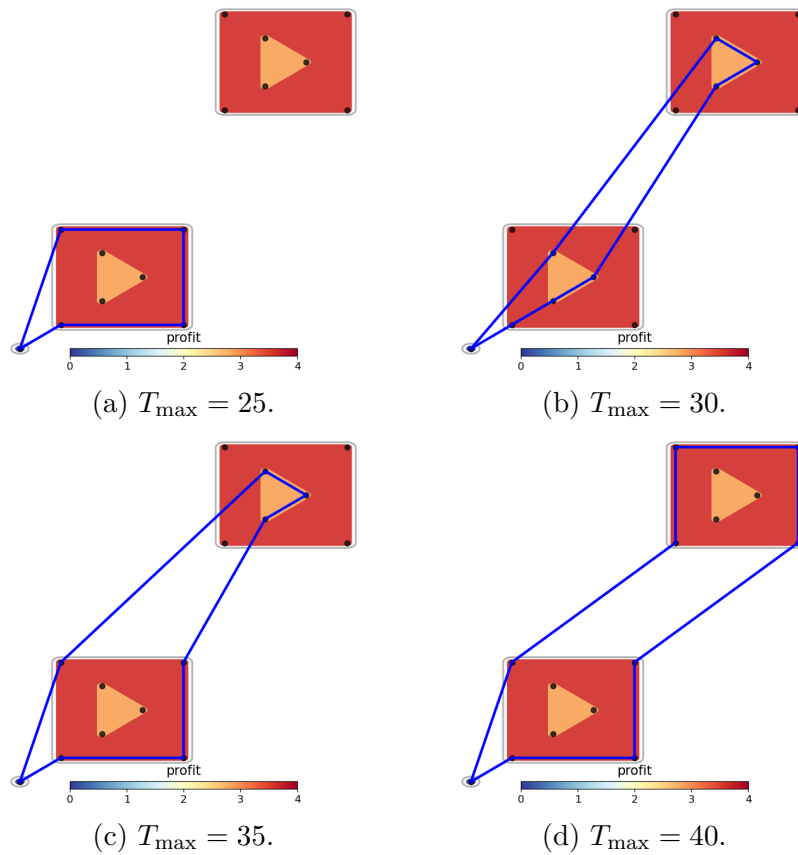


Figura 5.4: Exemplos de aplicação do COPS com dois clusters clientes e 4 subgrupos clientes, dois subgrupos para cada cluster. O algoritmo COPS-TABU escolhe diferentes caminhos para diferentes valores de $T_{\max}$, sempre escolhendo caminhos que alcançam maior recompensa possível. O subgrupo mais externo possui recompensa de 4, enquanto o subgrupo mais interno possui recompensa igual a 3.

Tabela 5.3: Avaliação do ILP e do COPS-TABU variando o orçamento do robô, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.

$T_{\max}$	ILP			COPS-TABU		
	R	L	T	R	L	T
40	8	38.4	0.01	8	37.3	0.04
35	7	33.8	0.02	7	32.8	0.04
30	6	29.9	0.05	6	16.5	0.02
25	4	20.5	0.05	4	16.5	0.02
20	4	18.8	0.04	4	16.5	0.01
15	3	12.12	0.04	3	10.1	0.02
10	0	0	0.03	0	0	0.01

5.3.2 Estudo de Caso — Avaliação do desempenho do COPS aumentando o número de clusters e mantendo o orçamento $T_{\max}$ constante.

Neste segundo estudo de caso, busca-se avaliar o comportamento do modelo mantendo um orçamento fixo de deslocamento em $T_{\max} = 150$ e aumentando progressivamente o número de clusters disponíveis. O objetivo é observar a seleção de clusters e subgrupos visitados, bem como o impacto dessa variação na recompensa total acumulada. Cada cluster é composto por três subgrupos distintos, com recompensas associadas diferenciadas (alguns cenários ilustrados na Figura 5.5).

As instâncias usadas foram criadas manualmente seguindo as seguintes regras:

- Dentro de cada cluster cliente, os vértices foram agrupados em subgrupos de forma que aqueles com recompensas mais altas tivessem rotas internas mais caras, ou seja, exigiam um tempo de deslocamento maior entre seus vértices quando considerados isoladamente.
- Cada cluster cliente é composto por três subgrupos:
 - Um subgrupo menor com 3 vértices, com recompensa $R = 60$.
 - Um subgrupo médio com 6 vértices, com recompensa $R = 70$
 - Um subgrupo maior com 8 vértices, com recompensa $R = 100$.
- Os clusters foram posicionados progressivamente mais distantes da origem, simulando o aumento do esforço de deslocamento à medida que a rota avança

A análise dos resultados apresentados na Tabela 5.4 confirma o comportamento esperado: à medida que o número de clusters aumenta, há um crescimento na recompensa (R) coletada, até que as restrições de orçamento impõem um limite natural à expansão da rota, o que acontece a partir de 7 clusters clientes.

Adicionalmente, as Figuras 5.5b e 5.5c demonstram a capacidade do método em selecionar subgrupos de diferentes níveis de recompensa de forma estratégica. Essa característica evidencia que a abordagem implementada é capaz de:

- Identificar subgrupos de alto valor, priorizando trajetos que maximizem o ganho sem exceder o limite orçamentário;
- Explorar subgrupos de menor recompensa quando necessário, visando garantir cobertura mais ampla e ganhos acumulativos significativos;

- Gerenciar a complexidade crescente do problema, visto que a adição de novos clusters amplia exponencialmente o espaço de busca.

Na Tabela 5.4 vemos como o COPS-TABU deixa de encontrar a rota ótima quando a complexidade aumenta para os problemas com 8,9 e 10 clusters clientes. E a execução da ILP deixa de ser viável, devido ao tempo de execução, para 9 e 10 clusters.

Assim, este estudo evidencia a escalabilidade da abordagem proposta e reforça sua aplicabilidade em cenários com número crescente de clusters, mantendo o equilíbrio entre maximização da recompensa total e respeito às restrições de orçamento.

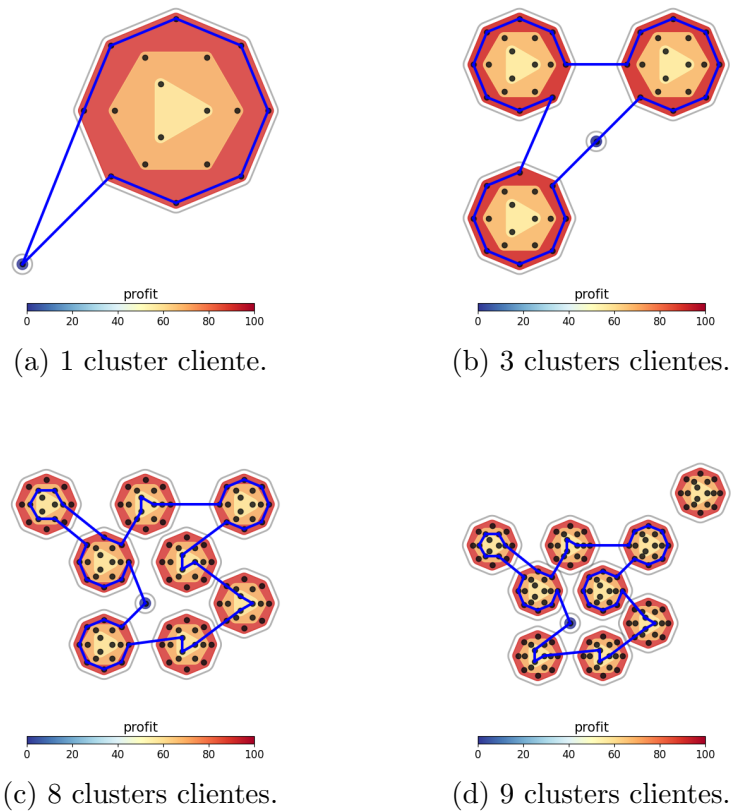


Figura 5.5: Exemplos de aplicação do COPS aumentando progressivamente o número de clusters mantendo o mesmo orçamento $T_{\max} = 150$. Cada cluster cliente possui 3 subgrupos, com recompensas de 60, 70 e 100.

Tabela 5.4: Avaliação do ILP e do COPS-TABU variando a quantidade de clusters e mantendo o orçamento $T_{\max}$ constante, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.

clusters	$T_{\max}$	ILP			COPS-TABU		
		R	L	T	R	L	T
1	150	100	36.07	0.02	100	25.5	0.3
2	150	200	63.18	0.03	200	50.2	0.5
3	150	300	132.5	0.06	300	65.8	0.7
4	150	400	149.0	0.2	400	90.3	1.0
5	150	500	149.7	0.25	500	119.1	1.4
6	150	600	149.8	0.26	600	137.1	1.9
7	150	700	149.4	1.1	700	149.5	3.0
8	150	700	147.9	109.9	670	148.1	40.4
9	150	—	—	—	670	148.2	341.9
10	150	—	—	—	690	149.1	918.5

5.3.3 Estudo de Caso — Cluster final com vários subgrupos (Rota não circular).

Neste estudo de caso, apresentamos uma análise experimental do COPS aplicado a um cenário com rota não circular, em que o ponto inicial e o ponto final da rota são distintos. O objetivo permanece o mesmo: maximizar a recompensa coletada ao longo do trajeto, respeitando as restrições estruturais do problema.

O estudo de caso aqui desenvolvido contempla um conjunto de clusters, onde:

- O cluster final é composto por seis subgrupos, cada um contendo um único vértice, representando possíveis pontos de chegada (endpoints);
- Existem quatro clusters clientes, cada um contendo dois subgrupos:
 - Um subgrupo menor, com recompensa $R = 6$.
 - Um subgrupo maior, com recompensa $R = 10$.

A cada execução, o algoritmo deve decidir não apenas quais subgrupos visitar, mas também qual endpoint do cluster final utilizar, de forma a fechar a rota dentro do orçamento $T_{\max}$ disponível.

Parâmetros do Experimento:

- Número de clusters clientes: 4
- Subgrupos por cluster cliente: 2 (recompensas: 6 e 10)
- Cluster final: 6 subgrupos (1 vértice cada)

- Recompensa total possível: $R = 4 \times 10 = 40$
- Orçamento total disponível $T_{\max}$ variado progressivamente de um valor mínimo (inviável) até um valor suficiente para a rota máxima.

Na Tabela 5.5 temos os dados do experimento. Observe que a partir do $T_{\max} = 65$ a recompensa obtida alcança o nível máximo $R = 40$. E, para $T_{\max} = 15$ não existe caminho viável para a rota percorrer nenhum subgrupo e alcançar algum endpoint.

A Figura 5.6 ilustra exemplos de rotas geradas para diferentes valores de orçamento $T_{\max}$. Neste caso, o cluster inicial encontra-se à direita do mapa, enquanto o cluster final, que contém os possíveis endpoints, está posicionado à esquerda. Conforme o valor do orçamento disponível, o algoritmo seleciona estrategicamente quais subgrupos visitar em cada cluster cliente e qual vértice utilizar como ponto final da rota.

Na Figura 5.6c, com $T_{\max} = 55$, o COPS opta por visitar dois subgrupos menores, alocando o orçamento de forma a maximizar a recompensa total (R) ao cobrir os quatro clusters. Por outro lado, com $T_{\max} = 50$ (Figura 5.6b), o algoritmo já não tem capacidade de visitar todos os clusters e, portanto, decide priorizar os três subgrupos maiores. Já na Figura 5.6a, com um orçamento bastante restrito, a rota consegue visitar apenas um subgrupo de menor custo e, conseqüentemente, menor recompensa.

Tabela 5.5: Avaliação do ILP e do COPS-TABU variando o orçamento do robô em um cenário onde o cluster final contém diferentes endpoints, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.

$T_{\max}$	ILP			COPS-TABU		
	R	L	T	R	L	T
70	40	69.8	0.1	40	67.2	1.5
65	40	64.4	0.1	40	64.02	1.7
60	36	59.9	0.17	36	58.7	1.3
55	32	53.4	0.26	32	53.9	1.8
50	30	47.4	0.24	30	47.4	1.4
45	26	42.4	0.18	26	41.8	1.2
40	22	39.7	0.24	22	36.6	1.1
35	20	32.4	0.21	20	34.4	1.1
30	16	27.9	0.19	16	29.5	1.1
25	12	22.2	0.12	12	22.2	1.1
20	6	19.1	0.06	6	19.1	1.1
15	0	0	0.04	0	0	0.8

Esses exemplos evidenciam a capacidade do modelo em adaptar suas decisões de roteamento de forma eficiente, levando em conta tanto a distribuição espacial dos subgrupos quanto o trade-off entre custo de deslocamento e recompensa.

Essa dinâmica demonstra a efetividade do modelo COPS na alocação inteligente de recursos em ambientes restritos, oferecendo soluções que equilibram ganho e custo de

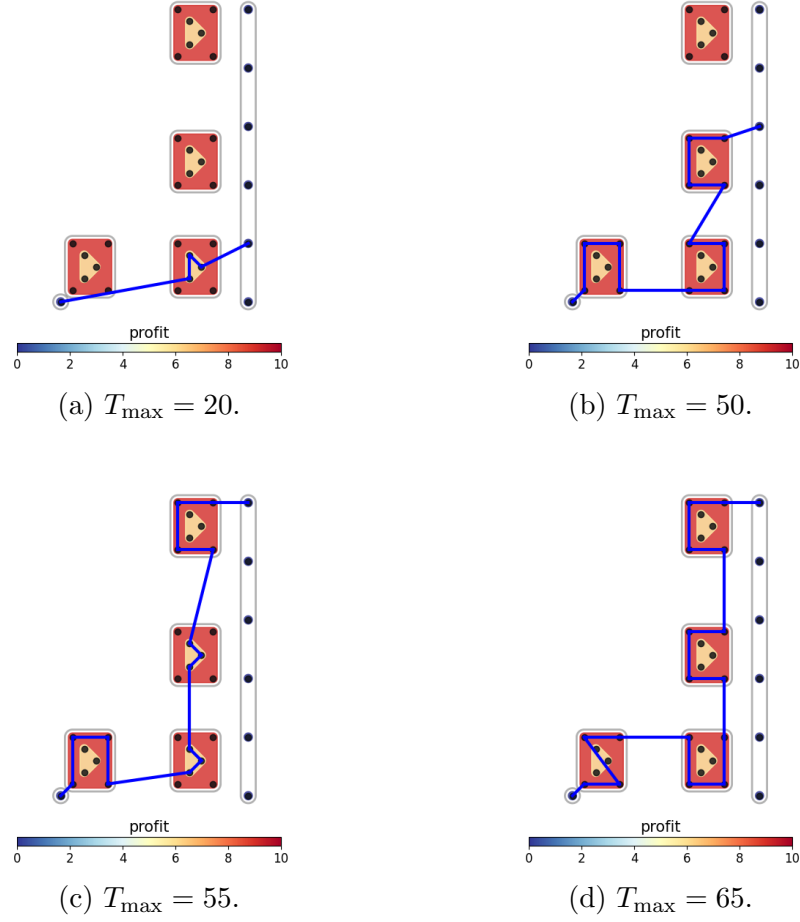


Figura 5.6: Exemplos de aplicação do COPS Não Circular. Cada cluster cliente possui 2 subgrupos, possuindo as recompensas de 6 e 10. E, o cluster final possui 4 subgrupos, o algoritmo escolhe o endpoint de acordo com o orçamento da viagem.

forma adaptativa conforme as condições de operação. A abordagem adotada permite não apenas maximizar a recompensa dentro das limitações impostas, mas também evidencia a capacidade do algoritmo de adaptar-se dinamicamente ao orçamento disponível, escolhendo rotas mais eficientes e estratégias de encerramento mais vantajosas.

5.3.4 Estudo de Caso — Cenário de aplicação em 3 dimensões.

Para ilustrar e validar a proposta do COPS para cenários tridimensionais, adaptamos a instância clássica berlin52 do problema do Caixeiro Viajante Generalizado (GTSP), propostas por [Fischetti et al., 1997], adicionando uma terceira dimensão (eixo z) aos dados originais da instância berlin52. Os valores do eixo z foram gerados de forma pseudo-

aleatória, variando entre 0 e 50, adicionando complexidade espacial à instância. Apenas um subgrupo foi adicionado a cada cluster, de maneira a permitir a visitação de todos os vértices. A recompensa associada a cada subgrupo foi mantida no mesmo valor da instância original.

As Figuras 5.7a e 5.7b ilustram os resultados obtidos para o $R=8000$ onde todos os clusters são visitados e para o $R=2000$ onde apenas um subgrupo mais valioso é visitado. Nas figuras a rota em azul evidencia o caminho tridimensional gerado como resposta.

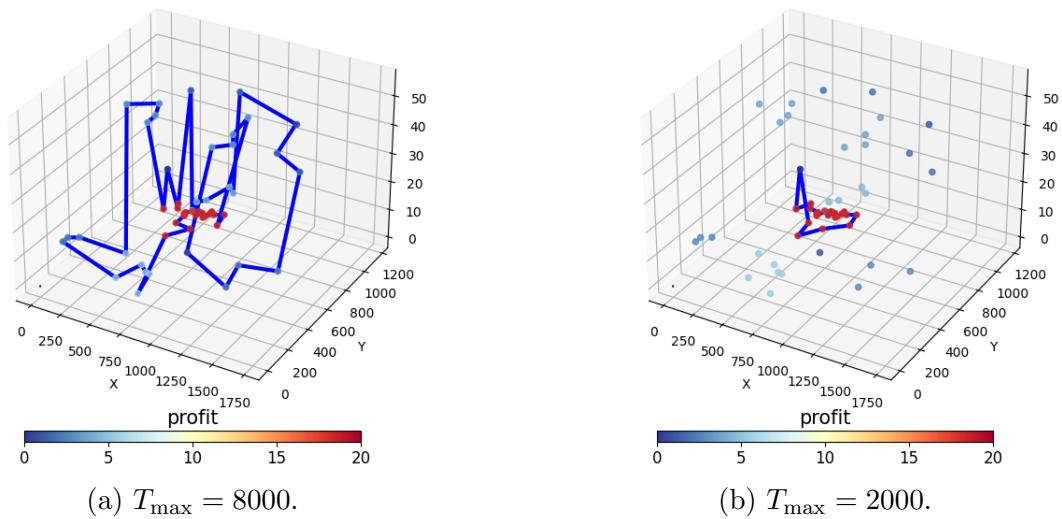


Figura 5.7: Exemplos de aplicação do COPS em 3 dimensões.

A Tabela 5.6 apresenta uma comparação entre os resultados obtidos pela formulação exata via Programação Linear Inteira (ILP) e o algoritmo heurístico COPS-TABU. Avaliamos múltiplos valores de orçamento para verificar a capacidade de ambos os métodos em encontrar soluções de alta qualidade dentro das restrições impostas.

Os resultados indicam que o algoritmo COPS-TABU é capaz de encontrar soluções próximas do ótimo na maioria dos casos. Entretanto, em situações onde o comprimento do caminho ótimo está muito próximo do valor limite de orçamento ($T_{\max}$), o COPS-TABU apresenta uma leve degradação na qualidade da solução. Isso se deve à dificuldade do algoritmo em explorar eficientemente soluções que exigem precisão próxima ao limite orçamentário.

Tabela 5.6: Avaliação do ILP e do COPS-TABU em um cenário com 3 dimensões, considerando Recompensa (R), Comprimento (L) e Tempo de execução (T) em segundos.

$T_{\max}$	ILP			COPS-TABU		
	R	L	T	R	L	T
8000	51	7588.4	0.39	51	7583.7	29.4
7000	49	6995.7	1.2	48	6977.5	33.0
6000	45	5934.0	1.3	45	5920.2	32.9
5000	40	4680.9	0.33	38	4860.7	25.2
4000	34	3961.0	1.26	33	3755.7	11.2
3000	27	2994.1	0.58	27	2653.0	3.45
2000	20	1456.3	0.24	20	1486.5	0.45
1000	4	957.5	0.12	4	997.8	0.07
500	0	0	0.02	0	0	0.06

5.4 Análise de performance ILP

Para compreender de forma abrangente o comportamento dos métodos em cenários de escalabilidade, realizamos uma análise estatística para avaliar como o número total de vértices, subgrupos e clusters impactam o desempenho.

Inicialmente, sistematicamente aumentamos o número de vértices em diferentes configurações de clusters e subgrupos. Esses resultados demonstraram que diversos fatores, além do número de vértices, podem influenciar o tempo computacional, dentre estes fatores destacamos o número de subgrupos, o número de clusters e o valor do orçamento ($T_{\max}$).

O tempo médio de computação é apresentado pela linha em vermelho na Figura 5.8. O experimento atingiu um limite de 65 vértices devido ao aumento no tempo de execução de alguns casos de ILP, particularmente aqueles com muitos clusters e poucos subgrupos, os quais resultam em soluções com rotas mais longas.

Embora a avaliação tenha sido limitada, essa restrição não comprometeu a integridade das nossas descobertas, destacando a utilização limitada de modelos baseados em ILP e sua característica exponencial de crescimento do tempo computacional. Mesmo com a adoção de estratégias avançadas como o *Branch-and-cut*, o problema mantém sua complexidade intrínseca. À medida que o grafo aumenta, o número de possibilidades viáveis cresce exponencialmente, tornando a resolução inviável para instâncias maiores. Nossos resultados evidenciam esse comportamento: o tempo de execução cresce rapidamente, especialmente em configurações com muitos clusters e poucos subgrupos, que tendem a gerar rotas mais extensas e complexas. Assim, reforça-se a necessidade de métodos aproximativos ou heurísticos para garantir escalabilidade em aplicações práticas.

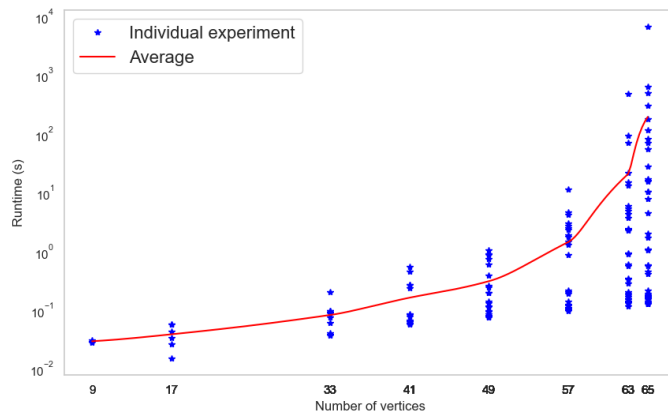


Figura 5.8: Análise de desempenho da ILP com o aumento do número de vértices em diferentes configurações de clusters e subgrupos.

5.4.1 Planejamento Experimental: Experimento 2^k

Para reforçar nossas descobertas com uma abordagem mais rigorosa, empregamos um planejamento experimental fatorial 2^k com 30 replicações.

O planejamento fatorial 2^k , onde k representa o número de fatores analisados, é composto por fatores testados em dois níveis distintos: alto (+1) e baixo (-1). Esse tipo de experimento permite avaliar simultaneamente os efeitos principais de cada fator, bem como as interações entre eles, sobre a variável de resposta.

O experimento 2^k é amplamente utilizado por sua eficiência na investigação de múltiplos fatores com um número reduzido de ensaios, sendo especialmente útil em fases iniciais de análise. Ao considerar todas as combinações possíveis dos níveis dos fatores, obtém-se um conjunto completo de 2^k experimentos.

No presente estudo, adotamos $k = 3$ fatores, testados em dois níveis cada, abrangendo todas as combinações possíveis. Especificamente, os fatores avaliados foram: número de clusters (8 e 16), número de subgrupos (16 e 32) e número de vértices (32 e 64). O principal objetivo desse planejamento fatorial foi identificar quais fatores e suas interações impactam significativamente o tempo de execução do algoritmo.

Após a realização dos experimentos, os dados foram analisados estatisticamente para determinar os efeitos significativos dos fatores e suas interações. Essa abordagem proporcionou uma melhor compreensão do comportamento do sistema sob diferentes condições de entrada.

A análise mostrou que, entre os fatores explorados, o número de clusters contribui com 11,8% da variação no tempo computacional, enquanto o número de subgrupos representa 14,9% e o número de vértices contribui com 15,4%. Além disso, a presença de erros responde por 7,7% da variação. O restante é devido às interações entre os diferentes fatores; a interação entre número de clusters e subgrupos corresponde à 11,8%, entre clusters

e vértices 11,6%, entre subgrupos e vértices 15,1% e, finalmente, a interação entre os 3 fatores, clusters, subgrupos e vértices é responsável por 11,7% do custo computacional. Esse erro refere-se à variabilidade residual não explicada pelos fatores testados, geralmente decorrente de flutuações naturais incontroláveis, como precisão do equipamento ou condições ambientais. Notavelmente, as interações entre essas três variáveis explicam a proporção restante da variação no tempo computacional.

Esses resultados iluminam a complexa interação das variáveis que impactam a eficiência computacional no projeto, mostrando que tanto o número de vértices, quanto o número de subgrupos e de clusters impactam no tempo de execução da ILP.

5.5 Análise de performance da Heurística

Para analisar a heurística, inicialmente realizamos um experimento utilizando as mesmas instâncias empregadas na avaliação da ILP, descrita na seção anterior.

Ao compararmos os resultados obtidos pela ILP (Figura 5.8) com os da heurística COPS-TABU (Figura 5.9), observamos que ambos apresentam desempenhos semelhantes, em termos de tempo de execução, para instâncias com menor número de vértices. No entanto, o COPS-TABU não atingiu seu limite de experimentação, permitindo a realização de testes mais elaborados e detalhados. Além disso, verificamos que o aumento médio da complexidade computacional da heurística não segue um crescimento exponencial, ao contrário do observado no caso da ILP.

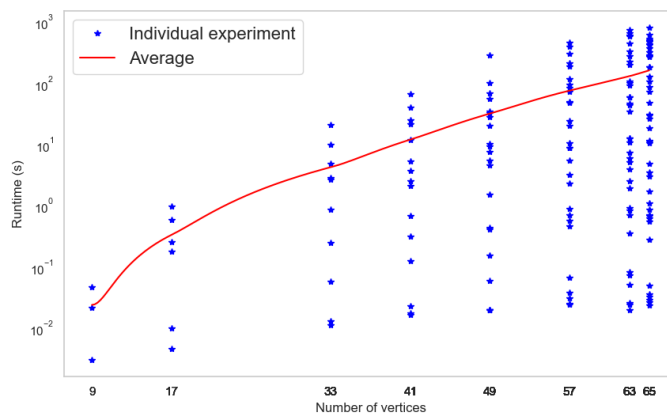


Figura 5.9: Análise de desempenho do COPS-TABU com o aumento do número de vértices em diferentes configurações de clusters e subgrupos.

5.5.1 Formalização com o experimento 2^k

Expandindo nossa investigação sobre a influência do número de vértices, subgrupos e clusters no tempo de execução, conduzimos um experimento fatorial 2^k para aprofundar no entendimento do comportamento do algoritmo COPS-TABU. Neste experimento, os níveis de clusters foram definidos em 10 e 15, os níveis de subgrupos em 30 e 60, e os níveis de vértices em 100 e 200. O objetivo foi avaliar como esses fatores, individualmente e em interação, influenciam o tempo de execução.

O número de vértices foi identificado como o principal fator responsável pela variância, contribuindo com 33,5%, seguido pelos subgrupos, com 19,7%, e pelos clusters, com 8,2%. Adicionalmente, as interações combinadas entre os fatores apresentaram impacto significativo no tempo de execução: a interação entre clusters e subgrupos contribuiu com 2,0%, entre clusters e vértices com 5,1%, entre subgrupos e vértices com 18,6%, e a interação conjunta dos três fatores explicou 10,3% da variância total.

Esses resultados do experimento fatorial 2^k fornecem insights valiosos sobre como as variações no número de clusters, subgrupos e vértices, bem como suas interações, influenciam significativamente o desempenho do tempo de execução do COPS-TABU. A baixa porcentagem de erro experimental, de 2,6%, ressalta a robustez e a confiabilidade dos efeitos observados neste estudo.

5.5.2 Avaliação do Tempo de Execução do COPS-TABU com Variação em Vértices, Subgrupos e clusters

Por fim, investigamos o impacto do número total de vértices, clusters e subgrupos no algoritmo COPS-TABU. Este experimento foi realizado em quatro etapas: nas etapas 1 e 2, mantivemos o número de clusters em 10 enquanto aumentamos progressivamente os subgrupos de 30 para 60; nas etapas 3 e 4, elevamos o número de clusters para 15, reavaliando as quantidades de subgrupos previamente testadas. Nossa avaliação envolve a análise do tempo de execução do COPS-TABU por meio de uma sequência de execuções, começando com 60 vértices e incrementando em 10 até alcançar um tempo de execução próximo a 1500 segundos (com $T_{max} = 1000$, abrangemos efetivamente todos os cenários possíveis de caminhos).

A Figura 5.10 mostra que o tempo de execução aumenta com o número de clusters, pois a Busca Tabu deve explorar uma vizinhança maior para alcançar um máximo local.

A comparação entre cenários com subgrupos iguais confirma que mais clusters resultam em tempos de execução mais longos. Por outro lado, o tempo de execução se correlaciona inversamente com o número de subgrupos, como evidenciado pelas diferenças entre as linhas azul e verde, e as linhas vermelha e laranja. Essa tendência ocorre porque menos subgrupos significam mais vértices por subgrupo, exigindo que a estratégia 2-opt avalie soluções maiores, o que aumenta o tempo de execução.

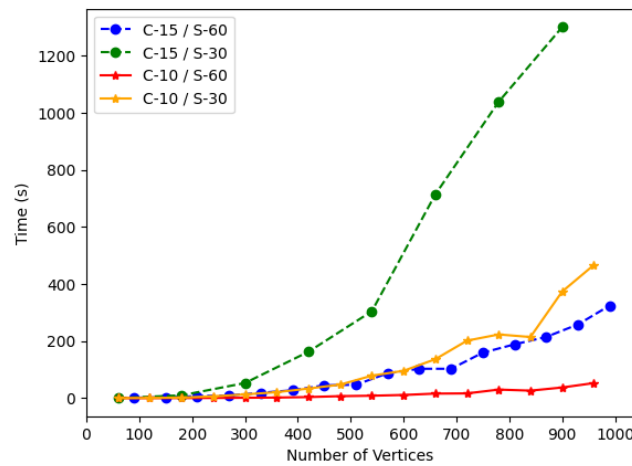


Figura 5.10: Análise do comportamento do tempo de execução do algoritmo COPS-TABU ao aumentar o número de vértices para diferentes tamanhos de subgrupos e clusters.

Capítulo 6

Conclusões e Trabalhos Futuros

A principal contribuição desta dissertação é a proposição do *Clustered Orienteering Problem with Subgroups* (COPS), uma extensão do *Orienteering Problem* (OP), capaz de representar novos cenários de roteirização, nos quais os vértices são organizados em regiões (clusters) e, dentro dessas regiões, podem ser aplicados critérios de reagrupamento ou priorização, permitindo diferenciar vértices mais relevantes do que outros.

Os resultados evidenciaram a versatilidade da modelagem COPS, tanto na generalização dos problemas clássicos de clusterização, como o *Clustered Orienteering Problem* (COP) e o *Set Orienteering Problem* (SOP), quanto na sua capacidade de abordar outros problemas de orientação que envolvem não apenas a clusterização de vértices, mas também a priorização de determinados grupos, considerando hierarquias, níveis de interesse ou critérios de exclusividade. Além disso, os resultados demonstram que nossa formulação é adequada para cenários não circulares, nos quais o ponto de destino difere do ponto de partida, permitindo inclusive a definição de múltiplos destinos dentro do mesmo cluster final, o algoritmo é capaz de selecionar automaticamente o ponto de destino mais apropriado para maximizar a recompensa da rota. Também foi validada sua aplicabilidade em cenários tridimensionais, sendo ideal, por exemplo, para planejamento de rotas com drones.

Foram desenvolvidas uma formulação em ILP e uma heurística baseada em Busca Tabu como alternativas de solução do problema. Em termos computacionais, a ILP mostrou-se eficiente para instâncias pequenas e médias, especialmente quando o número de clusters é moderado e há muitos subgrupos. Entretanto, à medida que a dimensão do problema aumenta, a ILP se torna impraticável em termos de tempo de execução. A heurística desenvolvida consegue resolver cenários maiores, nos quais a ILP é inviável, mantendo soluções de qualidade comparável em termos de recompensa. A comparação com benchmarks dos problemas COP e SOP indica que há margem para otimização no tempo de execução da heurística, o que será explorado em trabalhos futuros.

A análise do desempenho computacional revelou insights importantes sobre a escalabilidade e a eficiência das abordagens propostas. Em termos de tempo de execução, verificou-se que, à medida que o número de vértices e clusters aumenta, o tempo computacional tende a crescer, o que é esperado devido à complexidade do problema. No

entanto, um comportamento interessante foi observado em relação ao número de subgrupos: quando o número de subgrupos aumentou, o tempo computacional diminuiu, o que sugere que a estrutura hierárquica pode, de certa forma, facilitar a busca por soluções em alguns cenários. Esses resultados indicam que o COPS pode ser mais eficiente quando há uma maior granularidade na divisão dos clusters, o que pode ser útil em determinadas aplicações.

Estudos comparativos demonstram que, além de reproduzir adequadamente os cenários do COP e do SOP, o COPS vai além, permitindo a seleção estratégica de subgrupos dentro dos clusters, oferecendo um nível de controle e flexibilidade ausente nas versões clássicas. Dessa forma, o COPS amplia a literatura existente, introduzindo a capacidade de modelar e resolver problemas que não haviam sido contemplados anteriormente.

Por fim, os resultados demonstram não apenas o poder de abstração e generalização do COPS, mas também seu potencial de aplicação em cenários reais.

6.1 Análise Crítica das Limitações do COPS

Embora o COPS represente um avanço significativo em relação às variações anteriores do OP, oferecendo uma estrutura mais flexível ele também apresenta limitações inerentes que devem ser reconhecidas. Esta seção discute os principais desafios e restrições da formulação do COPS, tanto do ponto de vista teórico quanto prático, além de indicar possíveis direções para melhorias futuras.

1. Dificuldade na modelagem de subgrupos reais

Apesar de o COPS permitir uma modelagem mais rica, a definição prática dos subgrupos e clusters pode ser desafiadora. Em muitas aplicações do mundo real, pode não ser trivial identificar quais vértices devem compor um subgrupo, qual critério deve ser adotado para a formação dos clusters e como balancear, de maneira justa ou proporcional, a recompensa associada a cada subgrupo.

A dependência de uma boa definição da estrutura hierárquica implica que o COPS é tão eficaz quanto a qualidade da modelagem do problema. Isso pode exigir conhecimento específico do domínio ou métodos auxiliares de agrupamento, como clustering automático, o que adiciona uma camada de complexidade ao pré-processamento.

Considerando os desafios inerentes à definição manual de subgrupos e clusters no COPS, uma abordagem promissora é a utilização de Modelos de Linguagem de Grande Escala (LLMs) para auxiliar na clusterização semântica dos vértices e na definição das

recompensas associadas. A partir de descrições textuais dos vértices, os LLMs podem sugerir agrupamentos temáticos coerentes, além de atribuir valores de recompensa com base na relevância ou atratividade percebida.

Essa integração entre linguagem natural e otimização combinatória pode tornar o modelo mais adaptável a aplicações reais e menos dependente de parametrizações manuais ou conhecimento de domínio especializado. Um exemplo de aplicação de LLM para criar as instâncias COPS foi adicionado ao Apêndice A.

2. Limitação Temporal do COPS

Embora o COPS apresente uma estrutura expressiva para lidar com agrupamentos hierárquicos de vértices, ele não incorpora restrições explícitas de ordem temporal. Em outras palavras, o modelo não leva em consideração janelas de tempo ou sequências de visitação que respeitem restrições cronológicas específicas.

Essa ausência de ordenação pode gerar soluções inviáveis ou pouco realistas em cenários práticos. Por exemplo, em uma aplicação de roteirização turística, o COPS não possui mecanismos para garantir que a visita a um restaurante ocorra dentro de um intervalo de tempo coerente com o horário de almoço. Assim, o modelo pode propor rotas que, embora otimizadas em termos de pontuação ou distância, não atendam às necessidades reais de planejamento temporal.

Essa limitação destaca a importância de estender o COPS com restrições de janelas temporais ou modelagens híbridas que integrem aspectos espaciais e temporais. Dessa forma, seria possível aproximar melhor o modelo das necessidades operacionais de problemas reais, tornando as soluções obtidas mais aplicáveis em contextos práticos.

3. Foco em Recompensa em Detrimento da Eficiência Operacional

O OP e suas variações, como o COPS, têm como objetivo central maximizar a recompensa acumulada ao visitar um subconjunto de vértices, respeitando um orçamento de custo (tempo, distância ou outro recurso limitado). Diferentemente de problemas clássicos de roteirização, como o *Traveling Salesman Problem* ou o *Vehicle Routing Problem*, o OP não busca minimizar o custo total da rota, mas sim encontrar o trajeto que ofereça a maior recompensa possível dentro das restrições impostas.

Como consequência, o COPS não prioriza a eficiência em termos de custo operacional das rotas geradas. Em cenários práticos, essa característica pode resultar em soluções com rotas significativamente mais longas ou dispendiosas, ainda que altamente recompensadoras. Por exemplo, em um contexto logístico ou turístico, optar por trajetos que maximizam a pontuação sem considerar a eficiência operacional pode levar a gastos excessivos de combustível, tempo e recursos humanos, tornando a solução menos viável economicamente.

Essa limitação evidencia a necessidade de explorar abordagens híbridas ou extensões do COPS que considerem múltiplos objetivos, equilibrando a maximização da recompensa com a minimização de custos operacionais. Tais adaptações poderiam aproximar o modelo das demandas reais, onde a relação custo-benefício é um fator crítico para a aplicabilidade das soluções propostas.

4. Dificuldade de comparação com benchmarks existentes

Por ser uma variação nova e mais complexa, o COPS não possui ainda instâncias-padrão ou benchmarks consolidados na literatura. Isso dificulta a comparação direta de desempenho com algoritmos de outras variações do OP. Além disso, a adaptação de instâncias existentes do OP, COP ou SOP para o COPS tende a contemplar apenas cenários específicos de aplicação, não sendo suficiente para representar toda a variabilidade de situações que o COPS pode abranger.

5. Aumento da complexidade computacional

Uma das consequências diretas da maior expressividade do COPS é o aumento significativo da complexidade computacional do problema. Ao introduzir dois níveis de agrupamento (clusters e subgrupos), a dimensão do espaço de busca cresce substancialmente em comparação com o OP, o COP e o SOP.

Esse crescimento de complexidade não apenas amplia o espaço de busca, mas também dificulta o desenvolvimento de algoritmos eficientes, tornando desafiadora a construção de métodos capazes de lidar com instâncias de grande porte. Assim, soluções exatas tornam-se rapidamente inviáveis, e o projeto de heurísticas ou heurísticas sofisticadas passa a ser indispensável para obter resultados práticos.

6. Limitação do COPS em Cenários com Múltiplos Veículos

O COPS é formulado como um problema de roteirização para um único agente ou veículo, o que restringe sua aplicabilidade em cenários que envolvem frotas ou múltiplos recursos operacionais. Em muitas aplicações reais, como logística urbana, transporte de passageiros, inspeção de infraestruturas ou roteirização turística em grupo, o uso de múltiplos veículos é essencial para otimizar a cobertura geográfica, reduzir custos e aumentar a eficiência do sistema.

A ausência de suporte nativo a múltiplos veículos faz com que o COPS, em sua forma tradicional, não capture a complexidade adicional relacionada ao balanceamento de carga entre veículos, sincronização de rotas e coordenação de recursos. Assim, mesmo que o modelo seja eficiente em selecionar subgrupos de vértices e maximizar a recompensa total em um único trajeto, ele não se mostra adequado para operações em larga escala que exigem o planejamento simultâneo de múltiplas rotas.

Para superar essa limitação, seria necessário desenvolver extensões do COPS que incorporem a capacidade de planejar múltiplas rotas de forma coordenada. Isso poderia incluir o uso de técnicas de decomposição do problema, métodos de otimização multiagente ou heurísticas avançadas que considerem simultaneamente diversos veículos, balanceando a carga de trabalho entre eles e garantindo maior eficiência operacional. Dessa forma, o modelo se tornaria mais aderente às demandas de cenários reais, onde a operação com vários recursos é fundamental.

6.2 Trabalhos Futuros

O COPS representa um avanço relevante no estudo de problemas de roteamento com restrições de agrupamento, pois amplia a capacidade de modelagem e possibilita representar cenários mais complexos e realistas. Entretanto, sua aplicação prática demanda atenção especial a fatores como a elevada complexidade computacional, a definição criteriosa da estrutura hierárquica de clusters e subgrupos, além da adequação do modelo de recompensa aos objetivos reais do sistema.

Reconhecer essas limitações é fundamental para orientar pesquisas futuras que visem consolidar e expandir o potencial do COPS. Entre os principais caminhos de investigação, destacam-se:

- Desenvolvimento de métodos automáticos para definição de clusters e subgrupos;
- Extensão do modelo para suportar múltiplos agentes ou veículos;
- Integração de restrições temporais que reflitam cenários reais de planejamento;
- Criação de instâncias *benchmark* específicas para o COPS, permitindo avaliações comparativas sob condições padronizadas;
- Proposição de heurísticas mais eficientes, com foco na redução do tempo de execução sem comprometer a qualidade das soluções.

Essas direções de pesquisa podem contribuir para tornar o COPS uma ferramenta ainda mais robusta e aplicável em diferentes contextos, ampliando sua relevância acadêmica e potencial de uso em problemas reais.

Bibliografia

- Ahmed, Zaheed e Irfan Younas [2011]. “A dynamic programming based GA for 0-1 modified knapsack problem”. Em: *International Journal of Computer Applications* 16.7, pp. 1–6.
- Akgun, Gokcen [2017]. “Optimization of Resource Allocation and Routing in Political Campaign Problems”. Tese de dout. Wichita State University.
- Angelelli, Enrico, Claudia Archetti e Michele Vindigni [2014]. “The clustered orienteering problem”. Em: *European Journal of Operational Research* 238.2, pp. 404–414.
- Archetti, Claudia, Francesco Carrabs e Raffaele Cerulli [2018]. “The set orienteering problem”. Em: *European Journal of Operational Research* 267.1, pp. 264–272.
- Archetti, Claudia et al. [2014]. “The team orienteering arc routing problem”. Em: *Transportation Science* 48.3, pp. 442–457.
- Balas, Egon e Paolo Toth [1983]. “Branch and bound methods for the traveling salesman problem”. Em.
- Bharath-Kumar, Kadaba e J Jaffe [2003]. “Routing to multiple destinations in computer networks”. Em: *IEEE Transactions on communications* 31.3, pp. 343–351.
- Bouchakhchoukha, Adel e Hakim Akeb [set. de 2014]. “A Hybrid Heuristic for the Team Orienteering Problem”. Em: *International Journal of Mathematical, Computational, Physical and Quantum Engineering* 8, pp. 1135–1140.
- Cen, Chen et al. [2015]. *Towards city-scale mobile crowdsourcing: Task recommendations under trajectory uncertainties*.
- Chao, I-Ming, Bruce L Golden e Edward A Wasil [1996]. “The team orienteering problem”. Em: *European journal of operational research* 88.3, pp. 464–474.
- Chen, Cen et al. [2014]. “Traccs: a framework for trajectory-aware coordinated urban crowd-sourcing”. Em: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 2, pp. 30–40.
- Cho, Doo-Hyun e Han-Lim Choi [2019]. “A traveling salesman problem-based approach to observation scheduling for satellite constellation”. Em: *International Journal of Aeronautical and Space Sciences* 20.2, pp. 553–560.
- Correia, Pedro, Luís Paquete e José Rui Figueira [2018]. “Compressed data structures for bi-objective $\{0, 1\}$ -knapsack problems”. Em: *Computers & Operations Research* 89, pp. 82–93.
- Croes, Georges A [1958]. “A method for solving traveling-salesman problems”. Em: *Operations research* 6.6, pp. 791–812.

- Dallard, Herby, Sarah S Lam e Sadan Kulturel-Konak [2007]. “Solving the orienteering problem using attractive and repulsive particle swarm optimization”. Em: *2007 IEEE International Conference on Information Reuse and Integration*. IEEE, pp. 12–17.
- Dinkel, John J et al. [1976]. “Environmental inspection routes and the constrained traveling system salesman problem”. Em: *Computers & Operations Research* 3.4, pp. 269–282.
- Duman, Ekrem e I Or [2004]. “Precedence constrained TSP arising in printed circuit board assembly”. Em: *International Journal of Production Research* 42.1, pp. 67–78.
- Ferreira, Carlos E, Alexander Martin e Robert Weismantel [1996]. “Solving multiple knapsack problems by cutting planes”. Em: *SIAM Journal on Optimization* 6.3, pp. 858–877.
- Fischetti, Matteo, Juan José Salazar González e Paolo Toth [1997]. “A branch-and-cut algorithm for the symmetric generalized traveling salesman problem”. Em: *Operations Research* 45.3, pp. 378–394.
- Gavalas, Damianos et al. [2015]. “Heuristics for the time dependent team orienteering problem: Application to tourist route planning”. Em: *Computers & Operations Research* 62, pp. 36–50.
- Glover, Fred [1986]. “Future paths for integer programming and links to artificial intelligence”. Em: *Computers & operations research* 13.5, pp. 533–549.
- Golden, Bruce L, Larry Levy e Rakesh Vohra [1987]. “The orienteering problem”. Em: *Naval Research Logistics (NRL)* 34.3, pp. 307–318.
- Gomes, David E et al. [2021]. “Applying a genetic algorithm to a m-TSP: case study of a decision support system for optimizing a beverage logistics vehicles routing problem”. Em: *Electronics* 10.18, p. 2298.
- Gunawan, Aldy, Hoong Chuin Lau e Pieter Vansteenwegen [2016]. “Orienteering problem: A survey of recent variants, solution approaches and applications”. Em: *European Journal of Operational Research* 255.2, pp. 315–332.
- Gunawan, Aldy, Zhi Yuan e Hoong Chuin Lau [2014]. *A mathematical model and metaheuristics for time dependent orienteering problem*. PATAT.
- Gunawan, Aldy et al. [2021]. “Set Team Orienteering Problem with Time Windows”. Em: *Learning and Intelligent Optimization: 15th International Conference, LION 15, Athens, Greece, June 20–25, 2021, Revised Selected Papers 15*. Springer, pp. 142–149.
- Hanafi, Said e Arnaud Freville [1998]. “An efficient tabu search approach for the 0–1 multidimensional knapsack problem”. Em: *European Journal of Operational Research* 106.2-3, pp. 659–675.
- Hassin, Refael e Ariel Keinan [2008]. “Greedy heuristics with regret, with application to the cheapest insertion algorithm for the TSP”. Em: *Operations Research Letters* 36.2, pp. 243–246.

- Helsgaun, Keld [2000]. “An effective implementation of the Lin–Kernighan traveling salesman heuristic”. Em: *European journal of operational research* 126.1, pp. 106–130.
- Herzog, Daniel e Wolfgang Wörndl [2014]. “A Travel Recommender System for Combining Multiple Travel Regions to a Composite Trip.” Em: *Cbreccsys@ recsys* 1245, pp. 42–48.
- Hurkens, Cor AJ e Gerhard J Woeginger [2004]. “On the nearest neighbor rule for the traveling salesman problem”. Em: *Operations Research Letters* 32.1, pp. 1–4.
- Jandaghi, Hossein, Ali Divsalar e Saeed Emami [2021]. “The categorized orienteering problem with count-dependent profits”. Em: *Applied Soft Computing* 113, p. 107962.
- Ke, Liangjun et al. [2010]. “An ant colony optimization approach for the multidimensional knapsack problem”. Em: *Journal of Heuristics* 16.1, pp. 65–83.
- Kobeaga, Gorka, María Merino e Jose A Lozano [2018]. “An efficient evolutionary algorithm for the orienteering problem”. Em: *Computers & Operations Research* 90, pp. 42–59.
- Kolev, K. et al. [ago. de 2009]. “Continuous Global Optimization in Multiview 3D Reconstruction”. Em: *International Journal of Computer Vision* 84.1, pp. 80–96.
- Kumar, Ratnesh e Zhonghui Luo [2003]. “Optimizing the operation sequence of a chip placement machine using TSP model”. Em: *IEEE Transactions on Electronics Packaging Manufacturing* 26.1, pp. 14–21.
- Lau, Hoong Chuin et al. [2012]. “Dynamic stochastic orienteering problems for risk-aware applications”. Em: *arXiv preprint arXiv:1210.4874*.
- Levine, Matthew S [2000]. “Finding the right cutting planes for the TSP”. Em: *Journal of Experimental Algorithmics (JEA)* 5, 6–es.
- Liang, Yun-Chia, Sadan Kulturel-Konak e Alice E Smith [2002]. “Meta heuristics for the orienteering problem”. Em: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02 (Cat. No. 02TH8600)*. Vol. 1. IEEE, pp. 384–389.
- Liang, Yun-Chia e Alice E Smith [2006]. “An ant colony approach to the orienteering problem”. Em: *Journal of the Chinese Institute of Industrial Engineers* 23.5, pp. 403–414.
- Liao, Chen-Chih e Cheng-Hsin Hsu [2013]. “A detour planning algorithm in crowdsourcing systems for multimedia content gathering”. Em: *Proceedings of the 5th Workshop on Mobile Video*, pp. 55–60.
- Ludwig, Bernd, Bjørn Zenker e Jan Schrader [2009]. “Recommendation of personalized routes with public transport connections”. Em: *International Conference on Intelligent Interactive Assistance and Mobile Multimedia Computing*. Springer, pp. 97–107.
- Macharet, Douglas G, Armando Alves Neto e Daigo Shishika [2021]. “Minimal exposure dubins orienteering problem”. Em: *IEEE Robotics and Automation Letters* 6.2, pp. 2280–2287.

- Malucelli, Federico, Alessandro Giovannini e Maddalena Nonato [2015]. “Designing single origin-destination itineraries for several classes of cycle-tourists”. Em: *Transportation Research Procedia* 10, pp. 413–422.
- Marinakis, Yannis, Athanasios Migdalas e Panos M Pardalos [2005]. “Expanding neighborhood GRASP for the traveling salesman problem”. Em: *Computational Optimization and Applications* 32.3, pp. 231–257.
- Moradi, Nima, Vahid Kayvanfar e Majid Rafiee [2022]. “An efficient population-based simulated annealing algorithm for 0–1 knapsack problem”. Em: *Engineering with Computers* 38.3, pp. 2771–2790.
- Nedjatia, Arman e Béla Vizvárib [2020]. “Robot path planning by traveling salesman problem with circle neighborhood: Modeling, algorithm, and applications”. Em: *arXiv preprint arXiv:2003.06712*.
- North East Orienteering Association [2025]. *North East Orienteering Association – Orienteering across Northumberland, Tyne & Wear, County Durham and Cleveland*. <https://neorienteering.org.uk/>. Website accessed on 28 August 2025.
- Oliveira, Matheus RR, André Gustavo dos Santos e Matheus de Freitas Araujo [2015]. “Uma heurística busca tabu para o problema do mochileiro viajante”. Em: *Brazilian SBPO*.
- Osaba, Eneko, Esther Villar-Rodriguez e Izaskun Oregi [2022]. “A systematic literature review of quantum computing for routing problems”. Em: *IEEE Access* 10, pp. 55805–55817.
- Padberg, M. e G. Rinaldi [1987]. “Optimization of a 532-city symmetric traveling salesman problem by branch-and-cut”. Em: *Operations Research Letters* 6.1, 1–7. DOI: [10.1137/1033004](https://doi.org/10.1137/1033004).
- Patalia, Tejas P e GR Kulkarni [2011]. “Design of genetic algorithm for knapsack problem to perform stock portfolio selection using financial indicators”. Em: *2011 International Conference on Computational Intelligence and Communication Networks*. IEEE, pp. 289–292.
- Pěnička, Robert, Jan Faigl e Martin Saska [2019]. “Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants”. Em: *European Journal of Operational Research* 276.3, pp. 816–825.
- Pěnička, Robert et al. [2017]. “Dubins orienteering problem”. Em: *IEEE Robotics and Automation Letters* 2.2, pp. 1210–1217.
- Ray, Abhijit, Wu Jigang e Thambipillai Srikanthan [2004]. “Knapsack model and algorithm for hardware/software partitioning problem”. Em: *Computing and Informatics* 23.5-6, pp. 557–569.
- Razali, Noraini Mohd, John Geraghty et al. [2011]. “Genetic algorithm performance with different selection strategies in solving TSP”. Em: *Proceedings of the world congress*

- on engineering*. Vol. 2. 1. International Association of Engineers Hong Kong, China, pp. 1–6.
- Roberti, Roberto e Mario Ruthmair [2021]. “Exact methods for the traveling salesman problem with drone”. Em: *Transportation Science* 55.2, pp. 315–335.
- Schönberger, Johannes Lutz e Jan-Michael Frahm [2016]. “Structure-from-Motion Revisited”. Em: *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sevkli, Zülal e F Erdoğan Sevilgen [2006]. “Variable neighborhood search for the orienteering problem”. Em: *International Symposium on Computer and Information Sciences*. Springer, pp. 134–143.
- Shih, Wei [1979]. “A branch and bound method for the multiconstraint zero-one knapsack problem”. Em: *Journal of the Operational Research Society* 30.4, pp. 369–378.
- Silva, Aline Fernanda Furtado [Maio de 2025]. “Roteamento Eficiente para Abordagem de Indivíduos em Ambientes com Obstáculos Baseado em Normas Sociais”. Tese de Doutorado. Universidade Federal de Minas Gerais.
- Singh, Vishnu Pratap [2020]. “An approach to solve fuzzy knapsack problem in investment and business model”. Em: *Networked Business Models in the Circular Economy*. IGI Global, pp. 154–173.
- Skiścim, Christopher C e Bruce L Golden [1983]. “Optimization by simulated annealing: A preliminary computational study for the tsp”. Em: *Proceedings of the 15th conference on Winter Simulation-Volume 2*, pp. 523–535.
- Smith, Neil et al. [2018]. “Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark”. Em.
- Souffriau, Wouter et al. [2008]. “A personalized tourist trip design algorithm for mobile tourist guides”. Em: *Applied Artificial Intelligence* 22.10, pp. 964–985.
- Spillman, Richard [1995]. “Solving large knapsack problems with a genetic algorithm”. Em: *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*. Vol. 1. IEEE, pp. 632–637.
- Sylejmani, Kadri et al. [2014]. “Solving tourist trip planning problem via a simulated annealing algorithm”. Em: *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, pp. 1124–1129.
- Tasgetiren, Mehmet Fatih e Alice E Smith [2000]. “A genetic algorithm for the orienteering problem”. Em: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*. Vol. 2. IEEE, pp. 910–915.
- Tricoire, Fabien et al. [2010]. “Heuristics for the multi-period orienteering problem with multiple time windows”. Em: *Computers & Operations Research* 37.2, pp. 351–367.
- Uddin, Fakhar et al. [2023]. “An improvement to the 2-opt heuristic algorithm for approximation of optimal TSP tour”. Em: *Applied Sciences* 13.12, p. 7339.

- Van Der Merwe, Martijn et al. [2014]. “The cooperative orienteering problem with time windows”. Em: *Optimization Online* 7.11, pp. 1–11.
- Vanderster, Daniel C et al. [2009]. “Resource allocation on computational grids using a utility model and the knapsack problem”. Em: *Future Generation computer systems* 25.1, pp. 35–50.
- Vansteenwegen, Pieter e Manuel Mateo [2014]. “An iterated local search algorithm for the single-vehicle cyclic inventory routing problem”. Em: *European Journal of Operational Research* 237.3, pp. 802–813.
- Varakantham, Pradeep et al. [2015]. *DIRECT: A scalable approach for route guidance in selfish orienteering problems*. AAMAS.
- Vianna, Dalessandro Soares e José Elias Claudio Arroyo [2004]. “A GRASP algorithm for the multi-objective knapsack problem”. Em: *XXIV International Conference of the Chilean Computer Science Society*. IEEE, pp. 69–75.
- Yahiaoui, Ala-Eddine, Aziz Moukrim e Mehdi Serairi [2019]. “The clustered team orienteering problem”. Em: *Computers & Operations Research* 111, pp. 386–399.
- Yang, Jinhui et al. [2008]. “An ant colony optimization method for generalized TSP problem”. Em: *Progress in natural science* 18.11, pp. 1417–1422.
- Yu, Jingjin et al. [2015]. “Anytime planning of optimal schedules for a mobile sensing robot”. Em: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 5279–5286.
- Zhang, Shu, Jeffrey W Ohlmann e Barrett W Thomas [2014]. “A priori orienteering with time windows and stochastic wait times at customers”. Em: *European Journal of Operational Research* 239.1, pp. 70–79.
- Zhang, Wei et al. [2020]. “Clustered coverage orienteering problem of unmanned surface vehicles for water sampling”. Em: *Naval Research Logistics (NRL)* 67.5, pp. 353–367.
- Zhang, Yan, Yanmei Zhou e Hao Hu [2020]. “Research on optimization of knapsack problem in logistics distribution”. Em: *Advanced Multimedia and Ubiquitous Engineering: MUE-FutureTech 2020*. Springer, pp. 1–9.

Apêndice A

Métodos automáticos para definição de clusters e subgrupos utilizando Linguagem Natural e LLM

Neste apêndice, descrevemos uma ideia inicial para um modelo de linguagem natural (LLM) para auxiliar na definição da estrutura hierárquica do problema COPS, que consiste em organizar vértices em subgrupos e clusters, respeitando restrições de visita.

Etapas do processo:

- Preparação dos dados: Foram coletadas informações detalhadas sobre os vértices (atrações turísticas), incluindo nome, descrição, tipo, localização e nota de atratividade, como no exemplo abaixo:
 - dados = [{
 - "id": 1,
 - "nome": "Museu de Arte Moderna",
 - "descricao": "Museu de arte moderna com exposições interativas.",
 - "tipo_atracao": "Cultural", "tempo_medio_visita": "1h30",
 - "bairro": "Centro", "rua": "Av. das Artes, 100",
 - "nota_atratividade": 4.6 }, ...
- Geração de embeddings semânticos: Utilizamos o modelo de linguagem para converter textos descritivos em vetores numéricos (embeddings), que capturam a similaridade semântica entre os vértices.
- Clusterização e formação dos subgrupos: Com base nos embeddings, aplicamos técnicas de agrupamento (clustering) para identificar subgrupos de vértices semanticamente relacionados. Posteriormente, os subgrupos foram agrupados em clusters maiores, respeitando a hierarquia do problema.

- Cálculo da recompensa dos subgrupos: A recompensa de cada subgrupo foi definida combinando a nota de atratividade (atributo quantitativo) com a coerência semântica dos vértices agrupados, refletindo seu valor agregado.
- Validação e apresentação dos resultados: Os clusters e subgrupos foram descritos em formato estruturado (JSON) e interpretados para facilitar o uso no modelo matemático do problema COPS.

A estrutura hierárquica gerada organiza as atrações turísticas em 3 clusters, cada um contendo subgrupos de pontos com características semelhantes. A recompensa de cada subgrupo foi calculada com base na nota de atratividade combinada com a coerência semântica entre os vértices.

A.0.1 Cluster 2 – Atrações culturais e artísticas

- **Subgrupo sub_2_1**
 - Vértices: 1 (Museu de Arte Moderna), 8 (Galeria Contemporânea 22)
 - Recompensa: 8.27
 - Tema: Arte moderna e contemporânea
- **Subgrupo sub_2_0**
 - Vértice: 5 (Teatro Municipal Antigo)
 - Recompensa: 9.6
 - Tema: Patrimônio histórico / apresentações clássicas

A.0.2 Cluster 0 – Natureza, família e educação

- **Subgrupo sub_0_0**
 - Vértices: 2 (Parque Ecológico), 4 (Zoológico Municipal)
 - Recompensa: 7.69
 - Tema: Contato com natureza e lazer familiar

- **Subgrupo sub_0_1**

- Vértice: 7 (Aquário Marinho)
- Recompensa: 9.2
- Tema: Educação ambiental / vida marinha

A.0.3 Cluster 1 – Entretenimento, lazer e aventura

- **Subgrupo sub_1_1**

- Vértices: 3 (Shopping Lux), 9 (Praia Central), 10 (Feira Cultural)
- Recompensa: 7.31
- Tema: Lazer urbano e experiências comerciais/culturais

- **Subgrupo sub_1_0**

- Vértice: 6 (Trilha da Serra Azul)
- Recompensa: 9.8
- Tema: Aventura e ecoturismo

A.0.4 Conclusão

Cada cluster representa um tema geral de interesse turístico (cultura, natureza, lazer urbano).

Os subgrupos com um único vértice tendem a ter recompensa mais alta, pois concentram alto valor individual e alta consistência.

Os subgrupos maiores têm recompensa moderada, refletindo leve dispersão semântica entre os pontos.

Essa estrutura já está pronta para ser usada como entrada no algoritmo COPS, garantindo que no máximo um subgrupo por cluster seja visitado, maximizando a recompensa da rota.

Benefícios do uso do LLM:

- Automatização da criação da estrutura hierárquica complexa a partir de dados textuais ricos;

- Integração de múltiplas fontes de informação (descrição, tipo, localização);
- Suporte à tomada de decisão para definir agrupamentos e recompensas que não seriam triviais manualmente.
- Este processo demonstra como técnicas modernas de inteligência artificial podem enriquecer e flexibilizar a modelagem de problemas combinatórios aplicados ao planejamento e roteirização.

Apêndice B

Exemplos de Aplicação do Modelo COPS

Este anexo apresenta uma coletânea de cenários que ilustram possíveis aplicações práticas do modelo proposto. O objetivo é fornecer ao leitor uma compreensão mais clara de como a estrutura hierárquica de clusters e subgrupos pode ser utilizada para representar diferentes contextos de tomada de decisão em ambientes com restrições orçamentárias e recompensas associadas a visitas parciais de regiões.

O modelo COPS se destaca por permitir um grau elevado de flexibilidade e personalização na seleção de vértices, é especialmente útil em problemas de roteirização onde há:

- Agrupamentos geográficos ou temáticos de pontos de interesse;
- Diferentes níveis de prioridade ou importância entre os vértices;
- Necessidade de decisões estratégicas sobre o quanto investir em cada região;
- Demandas que se beneficiam da visita parcial, sem obrigatoriedade de cobertura total do espaço.

Os cenários apresentados a seguir foram concebidos com o intuito de explorar essas possibilidades, adaptando o modelo COPS a diferentes domínios como logística urbana, turismo cultural e planejamento de serviços. Em cada cenário, são destacadas variações na estrutura dos clusters, nas regras de recompensa, e nas estratégias de roteamento, evidenciando a versatilidade do modelo para representar diferentes realidades operacionais.

Cenário 1: Aplicação em Roteirização Logística com Agrupamento Hierárquico de Demandas

Considere o contexto de uma empresa de logística que realiza coletas ou entregas em uma cidade de médio ou grande porte. Com o objetivo de otimizar o planejamento de suas rotas diárias, a empresa agrupa a malha urbana em clusters geográficos, que podem representar bairros, zonas administrativas ou regiões separadas por barreiras logísticas

(como pedágios, pontes ou longas vias expressas). Cada um desses clusters representa uma área onde o deslocamento entre vértices é mais eficiente do que entre clusters diferentes, respeitando o princípio da regionalização.

Dentro de cada cluster, os pontos de interesse (ou vértices) são categorizados em três níveis de prioridade, definidos de acordo com a urgência da visita e o valor agregado ao serviço. Cada categoria forma um subgrupo no contexto do problema:

- **Subgrupo 1 – Vértices Prioritários:** São os pontos de maior prioridade, com alta demanda ou urgência. A visita a este subgrupo concede uma recompensa R por vértice. A política operacional da empresa determina que, caso um cluster seja atendido apenas neste nível, todos os vértices prioritários devem ser visitados, a fim de evitar uma nova visita ao cluster no dia seguinte.
- **Subgrupo 2 – Vértices Prioritários + Secundários:** Além dos prioritários, este subgrupo inclui vértices de importância média, com recompensa reduzida. A recompensa por visitar este subgrupo é de R por cada vértice primário mais $R/2$ para cada vértice secundário, por exemplo. A lógica da operação determina que, ao optar por visitar também os vértices secundários de um cluster, todos os vértices prioritários e secundários devem ser atendidos, evitando a necessidade de retorno àquela região ao longo da semana.
- **Subgrupo 3 – Vértices Prioritários + Secundários + Terciários:** Neste subgrupo, incluem-se todos os vértices do cluster, abrangendo também os pontos de baixa prioridade, com recompensa $R/10$, por exemplo. Da mesma forma, caso este subgrupo seja selecionado, todos os vértices do cluster devem ser visitados para evitar o retorno à região no futuro próximo. A recompensa por visitar este subgrupo é de R por cada vértice primário mais $R/2$ para cada vértice secundário mais $R/10$ para cada vértice terciário.

Na Figura B.1, é apresentado um exemplo de clusterização dos vértices em diferentes regiões. Cada cluster pode ser subdividido em até três subgrupos, conforme as especificações previamente descritas. A formação desses subgrupos segue o procedimento detalhado anteriormente

Neste cenário, cada cluster pode conter até três subgrupos hierárquicos, que representam diferentes níveis de cobertura da região. A escolha de qual subgrupo atender em cada cluster envolve um trade-off entre o custo de deslocamento, o tempo disponível na rota e a recompensa obtida. A principal característica desse problema é que apenas um subgrupo pode ser selecionado por cluster, e sua escolha implica a visita obrigatória de todos os vértices que o compõem.

As Figuras B.2 e B.3 ilustram exemplos de rotas geradas para diferentes valores de $T_{\max}$. Observa-se que o algoritmo tende a maximizar a recompensa sempre que o

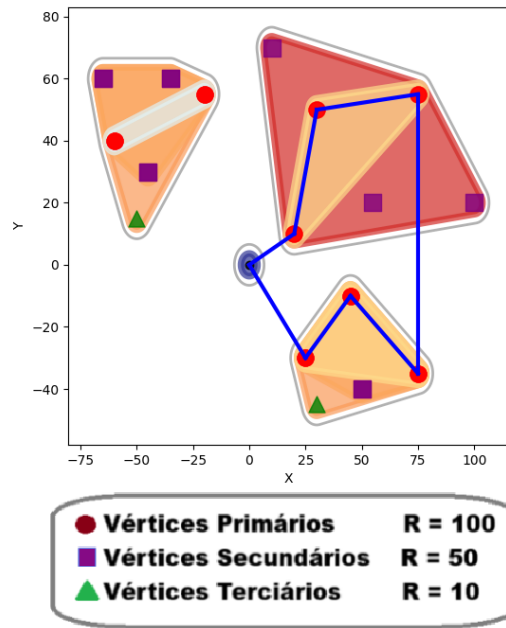


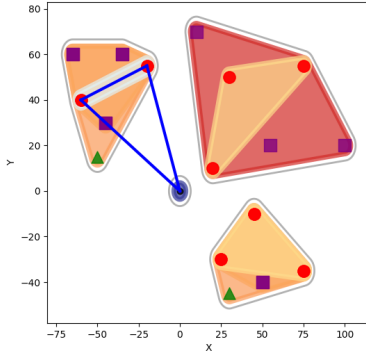
Figura B.1: Divisão de regiões (clusters) a serem atendidas no cenário 1. Em cada região podemos ter vértices prioritários, secundários e terciários.

orçamento permite. Em algumas situações, ele substitui um subgrupo prioritário por um secundário, enquanto em outras, com orçamento suficiente, opta por atender a mais de um cluster. A Tabela B.1 apresenta informações detalhadas sobre as distâncias percorridas em cada rota.

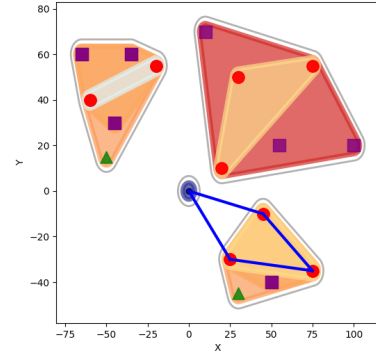
Este cenário representa uma aplicação prática do COPS, onde a decisão estratégica de quais subgrupos visitar em cada cluster é essencial para maximizar a recompensa total sob restrições de tempo ou distância da rota.

Tabela B.1: Resultados obtidos para diferentes valores de orçamento ($T_{\max}$) no cenário proposto. A tabela apresenta a recompensa total obtida (R), o comprimento da rota (L) e observações sobre quais subgrupos foram atendidos em cada caso.

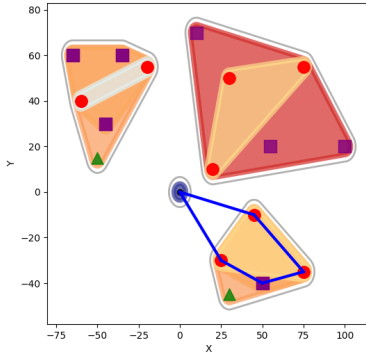
$T_{\max}$	Resultado		
	R	L	Obs:
174	200	173.35	Atende um grupo prioritário.
174.5	300	174.4	Atende um grupo prioritário.
177	350	176.6	Atende um subgrupo secundário.
190	360	186.1	Atende um subgrupo terciário.
300	450	291	Atende um subgrupo secundário.
304	500	303.8	Atende dois grupos prioritários.
310	600	305.2	Atende dois grupos prioritários.
315	650	310.9	Atende um subgrupo prioritário e um secundário.



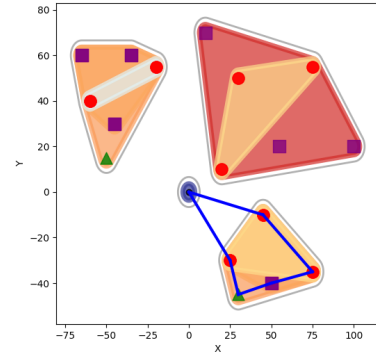
(a) Rota com $T_{\max} = 174$, o caminho encontrado atende um subgrupo prioritário com $R = 200$.



(b) Rota com $T_{\max} = 174.5$, o caminho encontrado atende um subgrupo prioritário com 3 vértices com $R = 300$.



(c) Rota com $T_{\max} = 177$, o caminho encontrado atende um subgrupo secundário com $R = 350$.



(d) Rota com $T_{\max} = 190$, o caminho encontrado atende um subgrupo terciário, atendendo todos os vértices do cluster com $R = 360$.

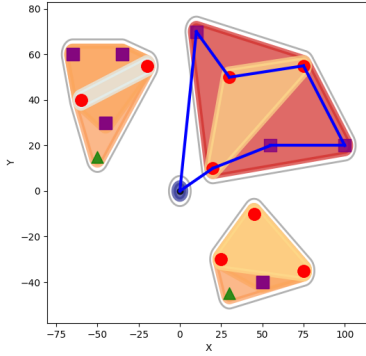
Figura B.2: Rotas para diversos orçamentos no cenário 1.

Cenário 2: Tratamento Isolado de Vértices Terciários como Subgrupos Independentes

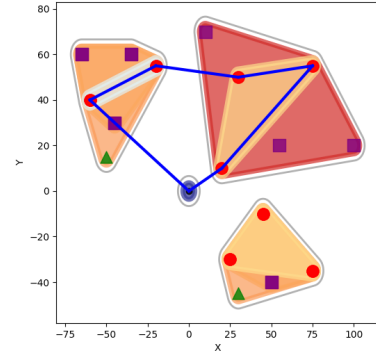
Neste segundo cenário, mantêm-se as mesmas premissas do Cenário 1 em relação aos clusters e aos vértices prioritários e secundários. A principal diferença aqui está no tratamento dos vértices terciários.

Enquanto no primeiro cenário os vértices terciários eram agrupados com os prioritários e secundários dentro de um subgrupo completo por cluster, aqui cada vértice terciário é tratado de forma independente, formando seu próprio subgrupo e também seu próprio cluster.

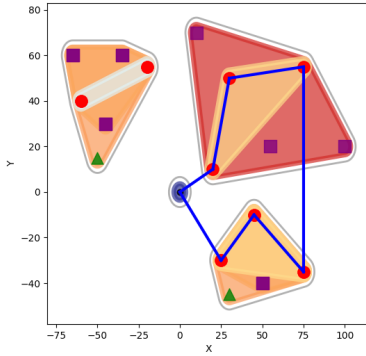
Na Figura B.4, adicionamos mais vértices terciários para demonstrar que, com um orçamento maior, o algoritmo pode optar por visitar apenas um dos vértices desse grupo. Nesta configuração, ele se torna mais eficiente ao alocar os recursos disponíveis para alcançar vértices isolados, maximizando a recompensa obtida com o uso do orçamento.



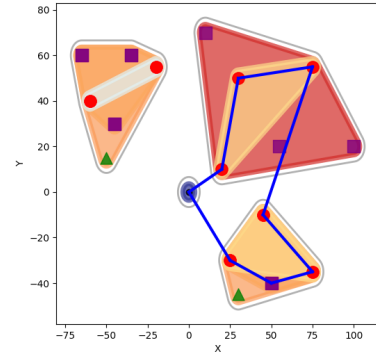
(a) Rota com $T_{\max} = 300$, o caminho encontrado atende um subgrupo secundário com $R = 450$.



(b) Rota com $T_{\max} = 304$, o caminho encontrado atende dois subgrupos prioritários com $R = 500$.



(c) Rota com $T_{\max} = 310$, o caminho encontrado atende dois subgrupos prioritários com $R = 600$.



(d) Rota com $T_{\max} = 315$, o caminho encontrado atende um subgrupo prioritário e um subgrupo secundário com $R = 650$.

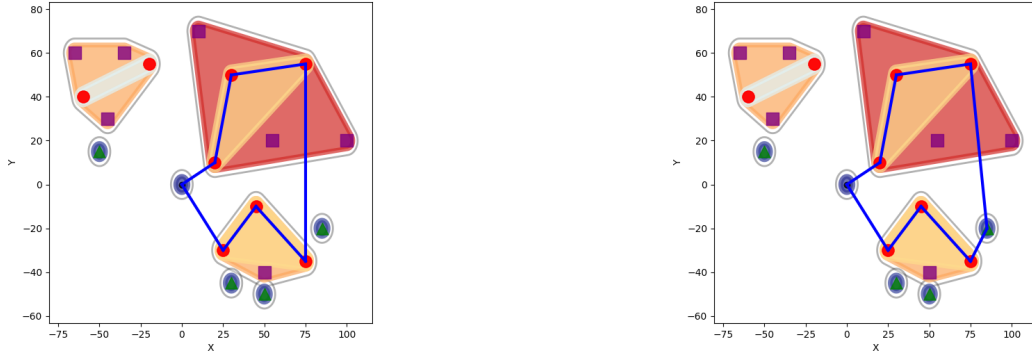
Figura B.3: Outras Rotas para o cenário 1.

Essa modelagem permite que vértices terciários sejam avaliados isoladamente, viabilizando sua visita caso estejam próximos de uma rota já planejada e desde que sua inclusão respeite o orçamento total da rota (em tempo, distância ou custo). Assim, os vértices de baixa prioridade podem ser atendidos de maneira oportunista, sem exigir a visita completa de todos os terciários de uma região.

Essa variação torna o modelo mais flexível e realista para situações em que vértices de baixa recompensa podem ser convenientemente adicionados ao trajeto principal, aumentando o aproveitamento da rota sem comprometer a eficiência logística.

Cenário 3: Subgrupo 1 com Vértices Prioritários, Subgrupo 2 com Vértices Prioritários + Secundários, e Vértices Secundários Isolados

Neste cenário, a estrutura dos clusters e subgrupos mantém-se semelhante aos cenários anteriores, mas com uma abordagem mais flexível para os vértices secundários. Aqui, a principal diferença é que cada vértice secundário é tratado como um subgrupo indepen-



(a) Rota com $T_{\max} = 306$, o caminho encontrado atende dois subgrupos prioritários com $R = 600$ e um comprimento de $L = 305.2$.

(b) Rota com $T_{\max} = 310$, o caminho encontrado atende dois subgrupos prioritários e um cluster terciário isolado com $R = 610$ e um comprimento de $L = 308.9$.

Figura B.4: Rotas para diversos orçamentos no cenário 2.

dente e pode ser visitado isoladamente, desde que haja viabilidade dentro do orçamento total da rota, seja em termos de tempo, distância ou custo.

A organização dos subgrupos dentro de cada cluster é a seguinte:

- **Subgrupo 1 – Vértices Prioritários:** Semelhante aos cenários anteriores, o subgrupo 1 contém apenas os vértices de maior prioridade. A recompensa por visitar um vértice prioritário é de R por vértice. A política da empresa exige que, ao escolher visitar este subgrupo, todos os vértices prioritários do cluster sejam atendidos. Nesse caso, a rota é otimizada para cobrir os vértices prioritários do cluster de forma eficiente, evitando a necessidade de uma visita subsequente.
- **Subgrupo 2 – Vértices Prioritários + Secundários:** Este subgrupo inclui tanto os vértices prioritários quanto os vértices secundários. A recompensa por visitar este subgrupo é de R por cada vértice prioritário mais $R/2$ por cada vértice secundário. A principal vantagem de optar por este subgrupo é que, ao selecioná-lo, todos os vértices prioritários e secundários do cluster são atendidos, evitando a necessidade de retorno a essa região no futuro. A escolha desse subgrupo é preferível quando o orçamento disponível para a visita de todos os pontos dentro do cluster.
- **clusters Secundários Isolados:** Além de serem agrupados com os vértices prioritários dentro de um único subgrupo, cada vértice secundário é tratado como um subgrupo e cluster independentes. A recompensa por visitar um vértice secundário isoladamente é de $R/10$, que é significativamente menor do que a recompensa oferecida dentro do subgrupo 2. No entanto, a vantagem dessa abordagem é que os vértices secundários podem ser visitados de maneira oportunista, caso a rota já passe por sua proximidade e o custo de inclusão esteja dentro do orçamento da via-

gem. Isso permite que a empresa maximize o aproveitamento da rota, atendendo a demandas secundárias sem a necessidade de uma visita completa a todos os pontos de um cluster.

Neste cenário, a principal decisão estratégica gira em torno da escolha entre visitar apenas os vértices prioritários (Subgrupo 1), todos os vértices prioritários e secundários (Subgrupo 2), ou ainda realizar visitas isoladas aos vértices secundários. A estrutura permite uma flexibilidade maior, onde a empresa pode optar por um equilíbrio entre a cobertura total de um cluster e a otimização do trajeto com a visita de vértices secundários de forma independente.

Percebe que ao escolher visitar o subgrupo 2 todos os outros clusters também são visitados. Ou seja, ao escolher visitar o subgrupo 2 a recompensa obtida será R para cada vértice prioritário, $R/2 + R/10$ para cada vértice secundário. O fato dos vértices secundários valerem mais agrupados garante o atendimento a todos caso esteja dentro do orçamento.

A escolha entre o Subgrupo 1 e o Subgrupo 2 é guiada por um trade-off entre a recompensa obtida e a cobertura desejada. Para que seja preferível a visita do Subgrupo 2 como um todo (com todos os vértices prioritários e secundários), a recompensa adicional por cada vértice secundário ($R/2$) precisa justificar o custo de visitá-los em conjunto. Por outro lado, a visita isolada a um vértice secundário, com recompensa de $R/10$, é uma opção quando a proximidade ao trajeto e a flexibilidade de incluir vértices adicionais são vantajosas, sem comprometer o orçamento da rota.

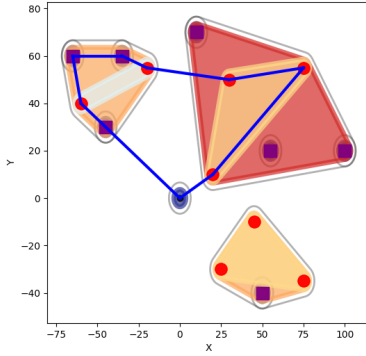
Este cenário oferece uma dinâmica mais rica para a tomada de decisões estratégicas na roteirização, pois permite uma combinação de abordagem rígida (ao escolher Subgrupo 1 ou 2) com uma abordagem mais flexível (ao incluir vértices secundários isoladamente), proporcionando um equilíbrio entre a cobertura e a eficiência do trajeto.

Cenário 4 – Turismo Cultural com Preferências de Grupos

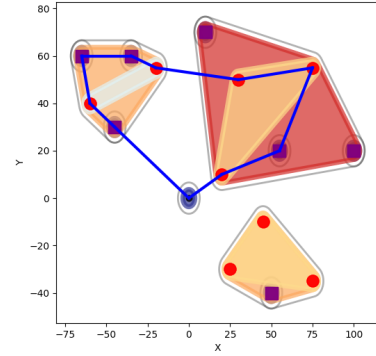
Neste cenário, uma agência de turismo cria roteiros personalizados para pequenos grupos de turistas em uma cidade histórica. Os pontos turísticos são organizados em regiões, com base em sua localização geográfica e valor cultural, como centros históricos, bairros modernos e áreas naturais.

Dentro de cada região, os pontos de interesse são agrupados em diferentes **clusters**, com finalidades distintas:

- **Cluster A – Atrações principais:** inclui museus, monumentos e marcos históricos. Como essas atrações são consideradas essenciais, todas devem ser visitadas ao explorar a região. Por esse motivo, o Cluster A possui apenas um subgrupo contendo todos esses pontos, garantindo que, ao visitar a região, o roteiro abranja integralmente as atrações principais;



(a) Rota com $T_{\max} = 328$, o caminho encontrado atende um subgrupo secundário, um subgrupo primário + 3 clusters secundários unitários $R = 680$ e um comprimento de $L = 327.5$.



(b) Rota com $T_{\max} = 335$, o caminho encontrado atende um subgrupo secundário, um subgrupo primário + 4 clusters secundários unitários $R = 690$ e um comprimento de $L = 333.1$.

Figura B.5: Rotas para diversos orçamentos no cenário 3.

- **clusters B – Atrações com visita opcional e exclusiva por categoria:** abrangem locais como cafeterias, galerias alternativas, lojas de artesanato, pequenos parques urbanos ou feiras de rua. Neste caso, deseja-se visitar apenas uma opção por categoria, evitando redundância de experiências. Cada subgrupo dentro desses clusters contém uma alternativa específica, por exemplo, uma cafeteria.

Essa estrutura permite uma combinação entre rotas fixas e decisões flexíveis, adaptando o itinerário às preferências do grupo e aos recursos disponíveis. Além disso, ela evita sobrecarga de atividades similares (como múltiplas cafeterias), ao mesmo tempo em que garante a cobertura completa dos pontos de maior relevância cultural.

Cenário 5 – Manutenção de Equipamentos Industriais

Neste caso, uma empresa de manutenção realiza inspeções técnicas em equipamentos distribuídos em diversas fábricas da cidade. Cada fábrica é modelada como um **cluster**, e os equipamentos são classificados conforme sua criticidade operacional.

Cada cluster (fábrica) possui os seguintes subgrupos:

- **Subgrupo 1 – Equipamentos críticos:** inspeções obrigatórias mensais;
- **Subgrupo 2 – Críticos + importantes:** verificações trimestrais;
- **Subgrupo 3 – Todos os equipamentos:** inclui também os de menor prioridade, com manutenção semestral.

A visita a uma fábrica exige a seleção de **um único subgrupo**, cujos equipamentos devem ser todos inspecionados. A decisão depende do tempo de deslocamento, da urgência e da disponibilidade de recursos da equipe técnica.

Diferencial: o modelo permite priorizar fábricas com equipamentos críticos, enquanto outras são atendidas completamente apenas quando há orçamento ou conveniência logística.

Cenário 6 – Coleta de Dados Ambientais com Restrições Climáticas

Neste cenário, considera-se uma equipe de pesquisadores encarregada de coletar dados ambientais em diferentes regiões geográficas antes da chegada de uma frente climática adversa, como o início da estação chuvosa. A coleta deve ser realizada dentro de um limite de tempo ou orçamento pré-estabelecido, exigindo decisões estratégicas sobre quais pontos visitar.

As regiões de interesse contêm diversos pontos de medição, classificados segundo seu nível de acessibilidade:

- **Pontos de fácil acesso:** localizados próximos a centros urbanos e acessíveis por vias pavimentadas. Estes pontos são agrupados por região em **clusters**, e todos os pontos de fácil acesso de uma região formam um único **subgrupo**. Ao escolher atender a uma região, todos esses pontos devem ser visitados.
- **Pontos de difícil acesso:** localizados em áreas remotas, exigindo deslocamento por estradas secundárias, trilhas ou até mesmo acesso fluvial. Dada a dificuldade logística de alcançar esses locais, cada ponto de difícil acesso é modelado como um **subgrupo e um cluster isolado**, permitindo que sejam considerados individualmente na formulação do problema.

A estratégia adotada neste cenário é atender integralmente os pontos de fácil acesso sempre que uma região for selecionada, garantindo a eficiência da visita. Já os pontos de difícil acesso são tratados de forma oportunista, podendo ser incluídos na rota apenas se houver orçamento disponível.

Diferencial: Este cenário combina dois tipos de decisões. De um lado, há o atendimento obrigatório e coletivo de pontos agrupados. De outro, os pontos de difícil acesso são tratados como instâncias separadas, lembrando o OP clássico, onde cada vértice, de forma isolada, pode ou não ser visitado. Essa modelagem híbrida reflete a natureza prática do problema, em que a incerteza e a complexidade logística dos deslocamentos exigem um tratamento flexível e adaptativo, mostrando o potencial de exploração da COPS.

Cenário 7 – Atendimento Domiciliar em Saúde Pública

Neste cenário, uma equipe de agentes comunitários de saúde precisa realizar visitas domiciliares em uma cidade. As regiões são agrupadas em **clusters**, representando bairros ou áreas de cobertura, e dentro de cada região existem diferentes tipos de pacientes com necessidades distintas de acompanhamento.

Subgrupos por cluster:

- **Subgrupo 1 – Pacientes de risco elevado:** idosos, gestantes e pessoas com doenças crônicas;
- **Subgrupo 2 – Risco elevado + pacientes em monitoramento:** pacientes em acompanhamento pós-alta ou controle de medicação;
- **Subgrupo 3 – Todos os pacientes cadastrados na região:** visita geral a todos os pacientes cadastrados.

A visita a um cluster exige a escolha de um único subgrupo para atender. Quando uma equipe é designada para um bairro, ela deve realizar todas as visitas do subgrupo selecionado. A seleção do subgrupo pode depender da criticidade dos casos, do tempo disponível e da quantidade de recursos da equipe.

Diferencial: Este cenário busca otimizar a cobertura da equipe de saúde, priorizando áreas e casos de maior necessidade e evitando visitas redundantes ou desnecessárias.

Cenário 8 – Inspeção de Segurança em Instalações Críticas

Neste cenário, uma empresa de infraestrutura realiza inspeções periódicas em suas instalações críticas, como postos elétricos, subestações ou gasodutos. As instalações são agrupadas por região geográfica ou por tipo de estrutura (e.g., torres de transmissão, transformadores).

Subgrupos por cluster:

- **Subgrupo 1 – Pontos com alerta de risco:** locais que apresentaram problemas em inspeções anteriores ou apresentam sinais de falha;
- **Subgrupo 2 – Pontos com alerta + pontos com inspeção vencida:** combina os pontos de risco com aqueles cuja inspeção não foi realizada dentro do prazo recomendado;
- **Subgrupo 3 – Todas as instalações da região:** inclui todos os pontos da instalação, independentemente do risco ou prazo de inspeção.

Para uma visita a uma instalação, a equipe seleciona um único subgrupo para inspeção, com base no nível de risco, urgência e requisitos operacionais. A decisão sobre

o subgrupo a ser atendido depende da gravidade dos problemas identificados e da janela de tempo disponível.

Diferencial: Esse cenário oferece flexibilidade para priorizar instalações com maior risco, garantindo a segurança e a continuidade das operações, ao mesmo tempo que minimiza a necessidade de múltiplas visitas.

Cenário 9 – Entrega de Cestas Básicas em Programa Social

Neste cenário, uma prefeitura organiza a entrega de cestas básicas em comunidades carentes. Cada comunidade é representada como um **cluster**, e os beneficiários são classificados de acordo com critérios sociais que definem a prioridade das entregas.

Subgrupos por cluster:

- **Subgrupo 1 – Famílias com crianças pequenas e idosos:** prioridade para famílias com vulnerabilidade social mais grave;
- **Subgrupo 2 – Subgrupo 1 + famílias em situação de desemprego recente:** adiciona famílias que estão em situação de desemprego ou com baixo poder aquisitivo;
- **Subgrupo 3 – Todas as famílias cadastradas na comunidade:** inclui todos os beneficiários cadastrados, abrangendo as entregas para toda a comunidade.

Ao selecionar uma comunidade, a equipe de entrega deve atender todos os beneficiários do subgrupo escolhido. A seleção do subgrupo depende de critérios sociais e logísticos, com o objetivo de cobrir as áreas de maior necessidade primeiro, de forma eficiente.

Diferencial: Este cenário otimiza as entregas com base nas necessidades mais urgentes da população, ao mesmo tempo que minimiza o número de deslocamentos, garantindo que as áreas mais necessitadas sejam atendidas com maior rapidez.

Os cenários apresentados nesta seção ilustram de forma clara a flexibilidade e adaptabilidade do modelo COPS, que se destaca pela sua capacidade de lidar com problemas complexos de otimização em diversos contextos. Ao incorporar a noção de clusters e subgrupos, o COPS permite uma gestão eficiente de recursos, levando em consideração não apenas a localização geográfica dos pontos de interesse, mas também a prioridade e as restrições específicas de cada subgrupo de pontos.

Cada cenário descrito oferece um exemplo de como o COPS pode ser aplicado a problemas reais, desde a logística de entregas até a saúde pública e inspeções de segurança em instalações críticas. A escalabilidade do modelo e sua capacidade de incorporar múltiplos critérios de otimização o tornam uma ferramenta poderosa para enfrentar desafios que exigem uma solução balanceada entre custo, tempo e complexidade operacional.

A flexibilidade do COPS é um de seus maiores diferenciais, pois permite que diferentes tipos de restrições sejam incorporados à solução, ajustando-se facilmente às necessidades de cada contexto. A possibilidade de escolher entre diferentes níveis de prioridade para os subgrupos, bem como a opção de tratar pontos de visita de maneira individual ou coletiva, amplia as possibilidades de aplicação e torna o modelo altamente versátil.

Em síntese, o COPS se mostra uma ferramenta robusta e prática para a solução de problemas de roteirização com múltiplas restrições e prioridades, oferecendo uma abordagem inovadora e eficaz para otimizar processos em diferentes áreas, com uma ampla gama de potenciais aplicações em logística, saúde, segurança e outros campos. A flexibilidade e adaptabilidade do modelo o posicionam como uma solução promissora para o futuro de sistemas de otimização em ambientes dinâmicos e complexos.