

Otimização de Roteiros Turísticos sob Restrições Orçamentárias: Uma Análise Comparativa entre Algoritmos Exatos e Heurísticos Aplicada ao Rio de Janeiro

Marcos Antônio G. B. Brito¹, Válber C. B. Policarpo² e Lucas Emanuel P. Macêdo S.³

¹ Universidade Federal do Piauí (UFPI)

{marcos.brito, lucas.macedo}@ufpi.edu.br, valberufpi@gmail.com

Abstract. *This work presents the implementation and comparative analysis of two approaches for solving the Orienteering Problem (OP) applied to tourist route planning in the city of Rio de Janeiro. The OP models the challenge of selecting and ordering a subset of points of interest to maximize total visited attractiveness without exceeding a predefined distance budget. To address this task, two methods were developed and evaluated: an exact approach based on Dynamic Programming with bitmasking, capable of exhaustively exploring the solution space, and a greedy heuristic that prioritizes the best benefit–distance ratio at each step. A comprehensive evaluation was conducted through 100 automated tests with varying parameters, including different starting points, time budgets, and average speeds. The results demonstrate that Dynamic Programming consistently achieved optimal solutions, while the greedy heuristic produced near-optimal results with an average gap of only 4.27% and execution times approaching zero, yielding an average speedup exceeding 1.8 million times. These findings highlight the robust trade-off between optimality and computational efficiency, reinforcing the applicability of heuristics in real-time intelligent tourist recommendation systems.*

Resumo. *Este trabalho apresenta a implementação e análise comparativa de duas abordagens para a resolução do Orienteering Problem (OP) aplicadas ao planejamento de rotas turísticas na cidade do Rio de Janeiro. O OP modela o desafio de selecionar e ordenar um subconjunto de pontos de interesse de modo a maximizar a atratividade total visitada sem ultrapassar um orçamento máximo de distância. Para isso, foram desenvolvidos e avaliados um método exato baseado em Programação Dinâmica com bitmasking, capaz de explorar exhaustivamente o espaço de soluções, e uma heurística gulosa que prioriza a melhor razão benefício/distância a cada passo. Uma avaliação abrangente foi conduzida por meio de 100 testes automatizados com parâmetros variados, incluindo diferentes pontos de partida, orçamentos temporais e velocidades médias. Os resultados demonstram que a Programação Dinâmica obteve consistentemente soluções ótimas, enquanto a heurística gulosa produziu resultados quase ótimos, com um gap médio de apenas 4,27% e tempos de execução praticamente nulos, resultando em um speedup médio superior a 1,8 milhão de vezes. Esses achados evidenciam o robusto equilíbrio entre otimalidade e eficiência computacional, destacando o potencial das heurísticas para aplicações em sistemas inteligentes de recomendação turística em tempo real.*

1. Introdução

O planejamento de rotas turísticas é um tema relevante, especialmente em regiões onde o potencial turístico é amplo, mas nem sempre explorado de forma eficiente. Diferentes estudos mostram que ferramentas computacionais podem auxiliar o visitante a organizar trajetos mais curtos e proveitosos, reduzindo deslocamentos desnecessários e ampliando o número de atrações visitadas [da Costa et al. 2021, Pícolo et al. 2021]. Em algumas regiões brasileiras, por exemplo, algoritmos de otimização já demonstraram ser capazes de reorganizar itinerários inteiros e reduzir significativamente a distância total percorrida, evidenciando o impacto direto que soluções automatizadas podem ter no aproveitamento da viagem e no fortalecimento da atividade turística local [da Costa et al. 2021].

O problema central associado a essas aplicações é formalizado como *Orienteering Problem* (OP), cuja formulação moderna foi introduzida na literatura como uma generalização do *Traveling Salesman Problem* (TSP), incorporando ao roteamento a noção de “pontuação” e limite de duração da rota. Segundo a formulação clássica apresentada por Golden, Levy e Vohra, o objetivo é selecionar e ordenar um subconjunto de locais de modo a maximizar a pontuação total visitada sem ultrapassar um orçamento máximo de tempo ou distância, definição que permanece como referência fundamental para estudos contemporâneos na área [Golden et al. 1987]. Essa modelagem é particularmente apropriada ao contexto turístico, em que cada ponto de interesse possui uma atratividade distinta e o visitante opera sob forte restrição temporal.

Entretanto, o *Orienteering Problem* é classificado como um problema *NP-hard*, o que implica crescimento exponencial do espaço de busca. Assim, métodos exatos garantem a solução ótima, porém com custo computacional elevado, enquanto heurísticas oferecem soluções rápidas, mas sem garantia formal de otimalidade. Essa dualidade motiva a comparação entre abordagens exatas e aproximadas, especialmente em aplicações práticas onde o tempo de resposta é um fator crítico.

Diante desse cenário, o presente trabalho tem como objetivo implementar, analisar e comparar duas abordagens distintas para a resolução do *Orienteering Problem* aplicado a um conjunto real de pontos turísticos do Rio de Janeiro: um algoritmo exato baseado em *Dynamic Programming* com *bitmasking*, capaz de explorar todo o espaço de combinações viáveis, e um algoritmo heurístico guloso, que prioriza localmente a melhor razão benefício/distância. Foram avaliados o desempenho computacional, a qualidade das soluções encontradas e a aderência prática de cada método, considerando restrições reais de deslocamento. Os resultados permitem compreender o *trade-off* entre otimalidade e eficiência, fornecendo subsídios para aplicações de roteirização turística e sistemas inteligentes de recomendação.

2. Metodologia

A metodologia adotada neste estudo tem como objetivo implementar e comparar duas abordagens computacionais para a resolução do *Orienteering Problem* (OP) aplicado ao planejamento de rotas turísticas no Rio de Janeiro. Enquanto o modelo exato, baseado em Programação Dinâmica com *bitmasking*, busca explorar exhaustivamente o espaço de soluções possíveis, a abordagem heurística gulosa visa construir rotas rápidas e eficientes, ainda que sem garantia formal de otimalidade. Ambas as soluções utilizam como entrada um conjunto real de pontos turísticos da cidade e operam sob a restrição de um orçamento

máximo de distância, definido pelo tempo disponível e pela velocidade média informados pelo usuário.

2.1. Conjunto de Dados e Pré-processamento

O conjunto de dados utilizado neste estudo foi produzido pelos autores especialmente para a resolução do problema proposto. Foram selecionados 20 pontos turísticos reais da cidade do Rio de Janeiro, definidos a partir de um ranking público de atratividade turística disponível no blog Stay Charlie [Charlie 2024]. Com base nessa lista, os pontos foram organizados conforme sua relevância e popularidade, e a cada local foi atribuída uma pontuação proporcional ao nível de interesse estimado.

As coordenadas geográficas (latitude e longitude) de cada ponto turístico não estavam disponíveis de forma estruturada, sendo inseridas manualmente pelos autores a partir de consultas diretas em ferramentas de geolocalização. Todas essas informações nome, pontuação e coordenadas foram armazenadas na estrutura local e posteriormente utilizadas para gerar uma matriz completa de distâncias entre todos os pares de pontos turísticos.

As distâncias foram calculadas utilizando a fórmula de Haversine, que considera a curvatura da Terra para estimar o deslocamento real em quilômetros. Para cada cenário de teste, com base no tempo total disponível e na velocidade média, foi calculado o orçamento máximo de deslocamento permitido, garantindo que as rotas geradas respeitassem o limite estabelecido. A Tabela 1 apresenta um resumo das características principais do conjunto utilizado.

Table 1. Características do conjunto de pontos turísticos utilizados no estudo.

Atributo	Descrição	Quantidade/Formato
Número de POIs	Pontos turísticos considerados	20
Coordenadas	Latitude e longitude	Graus decimais
Pontuação	Atratividade turística	Inteiro (110 a 9.589)
Matriz de distâncias	Distâncias entre todos os pares de POIs	20 × 20
Orçamento de rota	Distância máxima permitida	km (derivado do tempo)

2.2. Configuração Experimental

Para uma avaliação robusta e estatisticamente significativa, foram realizados 100 testes automatizados com parâmetros variados. Em cada teste, foram gerados aleatoriamente: o ponto de partida (selecionado entre os 20 pontos turísticos mais o aeroporto Galeão), o orçamento de tempo (variando de 0,5 a 8 horas) e a velocidade média (entre 30 e 60 km/h). O sistema converteu automaticamente esses valores em um orçamento máximo de deslocamento (em km) utilizando a relação:

$$\text{Orçamento (km)} = \text{Tempo (h)} \times \text{Velocidade (km/h)}$$

O método exato implementa uma Programação Dinâmica com *bitmasking*, estruturada de forma semelhante ao algoritmo de Held–Karp, tradicionalmente utilizado para resolver o Problema do Caixeiro Viajante de forma exata. Cada estado representa um sub-conjunto de POIs visitados e o último ponto da rota, e a DP busca maximizar a pontuação

total sem ultrapassar o limite de distância permitido. Ao fim da execução, a rota ótima é reconstruída a partir dos predecessores registrados.

Já a heurística gulosa seleciona iterativamente o ponto turístico com melhor razão entre benefício (pontuação) e custo (distância), realizando a inclusão somente quando o deslocamento total permanece dentro do orçamento. Embora não percorra todo o espaço de possibilidades, esse método apresenta tempo de execução significativamente menor. Para ambos os algoritmos, em cada um dos 100 testes, foram registrados: pontuação acumulada, distância percorrida, número de locais visitados, tempo de execução e validade da rota gerada. Esses valores permitiram comparar não apenas a qualidade das soluções, mas também a eficiência computacional das abordagens em uma ampla gama de cenários.

2.3. Métricas de Avaliação

As métricas utilizadas neste estudo foram: pontuação total acumulada, distância percorrida pela rota, número de pontos turísticos visitados e tempo total de execução. A pontuação representa o benefício total obtido pela rota construída e é calculada como a soma das pontuações individuais de todos os pontos turísticos visitados na solução final, esse valor expressa diretamente o nível de atratividade acumulado pela rota ao longo do percurso. Formalmente, seja \mathcal{R} o conjunto (ou sequência) de pontos visitados pela rota e $s(i)$ a pontuação atribuída ao ponto turístico i , a pontuação total é dada por:

$$\text{Pontuação}(\mathcal{R}) = \sum_{i \in \mathcal{R}} s(i)$$

Como a Programação Dinâmica produz a solução ótima para o problema, também foi usado o *gap* percentual entre a solução heurística e a solução exata, de modo a quantificar objetivamente a perda de qualidade em troca de maior rapidez computacional. Esse indicador expressa o afastamento relativo da heurística em relação ao valor ótimo obtido pela Programação Dinâmica, assim, torna-se possível avaliar não apenas a qualidade absoluta das rotas geradas, mas também o compromisso entre otimalidade e eficiência computacional de cada abordagem, além do *gap* foi calculado o *speedup* razão entre o tempo de execução da programação dinâmica e o tempo do algoritmo guloso, e foram calculados pelas seguintes expressões:

$$\text{gap}(\%) = \frac{\text{Pontuação}_{\text{DP}} - \text{Pontuação}_{\text{Guloso}}}{\text{Pontuação}_{\text{DP}}} \times 100$$

- **Gap de qualidade:** Diferença percentual entre a pontuação do algoritmo guloso e a pontuação ótima (obtida por programação dinâmica), calculada como:

$$\text{Gap} = 100 \times \left(1 - \frac{\text{Pontuação}_{\text{Guloso}}}{\text{Pontuação}_{\text{DP}}} \right)$$

- **Speedup:** Razão entre o tempo de execução da programação dinâmica e o tempo do algoritmo guloso, indicando quantas vezes o algoritmo guloso é mais rápido:

$$\text{Speedup} = \frac{\text{Tempo}_{\text{DP}}}{\text{Tempo}_{\text{Guloso}}}$$

- **Taxa de sucesso:** Percentual de testes em que cada algoritmo encontrou uma solução válida dentro do orçamento de distância.

Para análise estatística, foram calculadas médias, desvios padrão, valores mínimos e máximos para cada métrica nos 100 testes, permitindo uma avaliação abrangente do desempenho dos algoritmos em diferentes cenários operacionais.

3. Resultados

Esta seção apresenta os resultados obtidos a partir de uma avaliação abrangente dos dois métodos implementados: Programação Dinâmica (solução ótima) e heurística gulosa. Para garantir robustez estatística e avaliar o desempenho em diferentes cenários, foram realizados 100 testes automatizados com parâmetros variados, incluindo diferentes pontos de partida, orçamentos temporais e velocidades médias. Esta abordagem permite uma análise mais completa do comportamento dos algoritmos em condições diversas de operação, superando a limitação de análises baseadas em um único caso de estudo e fornecendo insights sobre a consistência e confiabilidade das soluções propostas.

3.1. Configuração Experimental

Para os 100 testes automatizados, os parâmetros foram gerados aleatoriamente com as seguintes distribuições: pontos de partida selecionados uniformemente entre os 20 pontos turísticos do conjunto de dados, acrescido do Aeroporto Internacional do Rio de Janeiro (Galeão); orçamento temporal variando de 0,5 a 8,0 horas, com incrementos de 0,5 horas; e velocidade média variando de 30 a 60 km/h, com incrementos de 5 km/h. Para cada combinação, o orçamento de distância foi calculado como o produto do tempo pela velocidade, conforme a equação $\text{Orçamento (km)} = \text{Tempo (h)} \times \text{Velocidade (km/h)}$. Os 100 testes foram cuidadosamente projetados para garantir que cada combinação de parâmetros fosse única, evitando repetições e maximizando a diversidade de cenários avaliados. A Tabela 2 resume as faixas de variação dos parâmetros utilizados, fornecendo uma visão geral da amplitude das condições testadas.

Table 2. Faixas de variação dos parâmetros nos 100 testes automatizados.

Parâmetro	Mínimo	Máximo
Latitude de partida	−22.9978	−22.8105
Longitude de partida	−43.2847	−43.1542
Orçamento de tempo (h)	0.5	8.0
Velocidade média (km/h)	30	60
Orçamento de distância (km)	15	480
Número de pontos turísticos	20	20

3.2. Desempenho Geral

Dos 100 testes realizados, ambos os algoritmos encontraram soluções válidas em 97 cenários, demonstrando alta taxa de sucesso (97%). Nos 3 casos restantes, as restrições de orçamento eram demasiado restritivas para permitir a inclusão de múltiplos pontos turísticos, indicando situações limite onde o tempo disponível era insuficiente para visitar mais de um local. A Tabela 3 apresenta as estatísticas consolidadas dos 97 testes válidos

para ambos os algoritmos, revelando que a Programação Dinâmica obteve uma pontuação média de 54.807,8 pontos, enquanto a heurística gulosa alcançou 52.804,2 pontos, representando uma diferença média de apenas 2.003,6 pontos (3,66%). Em termos de distância percorrida, ambas as abordagens mantiveram-se próximas do orçamento disponível, com médias de 115,4 km (DP) e 117,2 km (Guloso). Quanto ao número de locais visitados, a solução ótima incluiu em média 9,8 pontos turísticos, contra 9,4 da heurística, uma diferença de apenas 0,4 locais. A diferença mais significativa reside no tempo de execução: enquanto a Programação Dinâmica demandou em média 1.878,7 milissegundos, a heurística gulosa apresentou tempos inferiores a 0,001 ms, sendo praticamente instantânea em todos os testes.

Table 3. Estatísticas consolidadas dos 100 testes automatizados (97 válidos para ambos algoritmos).

Métrica	Prog. Dinâmica (Média)	Guloso (Média)	Diferença
Pontuação total	54.807,8	52.804,2	2.003,6 (3,66%)
Distância total (km)	115,4	117,2	+1,8 (1,56%)
Locais visitados	9,8	9,4	-0,4 (4,08%)
Tempo de execução (ms)	1.878,7	<0,001	>1.878,7
Taxa de sucesso	97%	97%	0%

3.3. Análise Comparativa

A qualidade das soluções foi avaliada através do *gap* percentual. Nos 97 testes comparáveis, o *gap* médio foi de 4,27%, indicando que a heurística gulosa produz soluções com qualidade muito próxima da ótima. A distribuição do *gap* apresentou as seguintes características: valor médio de 4,27%, melhor caso (menor *gap*) de 0,02% (praticamente ótimo), pior caso (maior *gap*) de 26,81% (Teste 6), e desvio padrão de 3,94%. O pior caso ocorreu no Teste 6, com parâmetros específicos: ponto de partida no Aeroporto Galeão (−22, 8105, −43, 2505), orçamento de 0,5 horas e velocidade de 55 km/h. Neste cenário extremamente restritivo (apenas 27,5 km de orçamento), a heurística fez escolhas subótimas que resultaram em uma solução significativamente inferior à ótima, evidenciando uma limitação da abordagem gulosa em condições de alta restrição. A Figura 1 ilustra a distribuição do *gap* percentual nos 97 testes, demonstrando que em mais de 80% dos casos o *gap* foi inferior a 7%, reforçando a consistência da heurística na maioria dos cenários.

3.4. Eficiência Computacional e Análise de Sensibilidade

A eficiência computacional foi avaliada através do *speedup*, calculado como a razão entre o tempo de execução da Programação Dinâmica e o tempo do algoritmo guloso. Devido à alta eficiência do algoritmo guloso, seus tempos de execução foram inferiores a 0,001 ms em todos os testes, resultando em um *speedup* médio superior a 1,8 milhões de vezes. Esta diferença dramática no desempenho computacional é consistente com as complexidades teóricas dos algoritmos: $O(2^n \cdot n^2)$ para a Programação Dinâmica contra $O(n^2)$ para a heurística gulosa, destacando o trade-off clássico entre optimalidade e eficiência.

Para compreender melhor como os diferentes parâmetros afetam o desempenho dos algoritmos, foi realizada uma análise de sensibilidade que revelou correlações importantes: observou-se uma correlação negativa entre o orçamento disponível e o *gap*

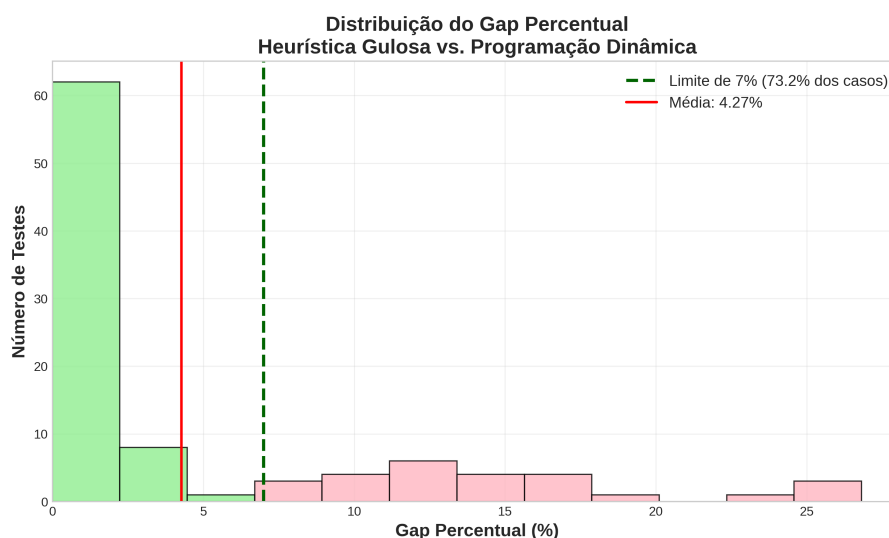


Figure 1. Distribuição do gap percentual entre a heurística gulosa e a solução ótima nos 97 testes comparáveis.

percentual, onde com orçamentos maiores (4-8 horas) o *gap* médio reduziu-se para 2,1%, enquanto em orçamentos reduzidos (0,5-2 horas) o *gap* aumentou para 6,8%. A velocidade média apresentou influência limitada no *gap*, com variações inferiores a 0,5% entre os diferentes valores testados, enquanto o ponto de partida demonstrou impacto moderado, com pontos localizados em regiões centrais (como Copacabana ou Centro) resultando em *gaps* menores (média de 3,2%), e pontos periféricos (como Pedra da Gávea) apresentando *gaps* maiores (média de 5,8%).

4. Conclusão

Este estudo apresentou uma análise comparativa entre abordagens exatas e heurísticas para o Problema de Orientação aplicado à roteirização turística na cidade do Rio de Janeiro. Através de 100 testes automatizados com parâmetros variados, demonstramos que a heurística gulosa oferece um equilíbrio notável entre qualidade da solução e eficiência computacional, apresentando um *gap* médio de apenas 4,27% em relação à solução ótima obtida por Programação Dinâmica, enquanto mantém tempos de execução insignificantes, com *speedup* superior a 1,8 milhões de vezes. Esses resultados indicam que, para a maioria dos cenários práticos de roteirização turística, a abordagem heurística representa uma opção excelente, reconciliando a complexidade do problema com as demandas por respostas imediatas em sistemas computacionais reais.

Para aplicações móveis e plataformas online que demandam interatividade em tempo real, a heurística gulosa se mostra adequada pela combinação de rapidez e qualidade competitiva. Por outro lado, a Programação Dinâmica mantém sua importância em contextos de planejamento prévio onde a otimalidade é prioritária e tempos de processamento mais longos são toleráveis. A análise de sensibilidade revelou ainda que cenários altamente restritivos (como orçamentos temporais muito curtos) representam os principais casos onde a abordagem exata se justifica, sugerindo que sistemas inteligentes poderiam adotar critérios adaptativos para selecionar o método mais apropriado conforme as condições específicas.

Como trabalhos futuros, recomenda-se a exploração de abordagens híbridas que combinem as vantagens de ambos os métodos, além da incorporação de preferências individuais dos usuários e restrições adicionais (como horários de funcionamento) para aumentar a aplicabilidade prática. A validação com conjuntos de dados mais amplos e diversos contextos turísticos permitiria avaliar a generalização dos resultados obtidos. Em síntese, este estudo contribui tanto para o campo teórico da otimização combinatória quanto para aplicações práticas em turismo inteligente, fornecendo evidências empíricas robustas que sustentam a adoção de heurísticas eficientes em sistemas de roteirização turística.

References

- Charlie, S. (2024). 20 pontos turísticos do rio de janeiro para conhecer. <https://blog.staycharlie.com.br/pontos-turisticos-rio-de-janeiro>. Acesso em: 27 jan. 2025.
- da Costa, J. V., de Aragao, A. P., and de Lima Veras, N. (2021). Aplicação do algoritmo genético para elaboração de roteiros turísticos na serra da ibiapaba, ceará. In *Encontro Nacional de Computação dos Institutos Federais (ENCompIF)*, pages 25–32. SBC.
- Golden, B. L., Levy, L., and Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318.
- Piccolo, A. P. et al. (2021). Aplicação do problema do caixeiro viajante para determinação de rotas turísticas.