

Otimização Multiobjetivo para Planejamento de Rotas Turísticas: Um Dia no Rio de Janeiro

Augusto M. P. de Mendonça¹

IC/UFF, Niterói, RJ

Filipe P. Sousa²

IME/UERJ, Rio de Janeiro, RJ

Igor M. Coelho³

IC/UFF, Niterói, RJ

Resumo. Este trabalho apresenta uma abordagem de otimização multiobjetivo para o planejamento de rotas turísticas de um dia no Rio de Janeiro. O problema foi modelado com quatro objetivos: minimização dos custos e do tempo de deslocamento, e maximização das atrações visitadas e da diversidade de bairros.

Palavras-chave. Otimização Combinatória, Otimização Multiobjetivo, Metaheurística, NSGA-II, Roteirização Turística

1 Introdução

O planejamento de rotas turísticas apresenta desafios de otimização, especialmente quando múltiplos critérios conflitantes precisam ser considerados simultaneamente. Ao estruturar um roteiro que busque minimizar custos e tempos de deslocamento enquanto maximiza o número de atrações visitadas e a diversidade de bairros, observamos objetivos naturalmente conflitantes que exigem técnicas especializadas para sua resolução. Estas características tornam tal problema adequado para abordagens de otimização multiobjetivo, possibilitando que turistas possam explorar eficientemente os pontos turísticos sem comprometer a qualidade da experiência, respeitando restrições práticas como horários de funcionamento, orçamentos disponíveis e limitações de distância.

Os algoritmos evolutivos multiobjetivo surgiram formalmente em 1985 com o VEGA de Schaffer [11], que revolucionou a abordagem para problemas com objetivos conflitantes. No entanto, foi o NSGA-II desenvolvido por Deb et al. em 2002 [3] que trouxe avanços decisivos ao superar limitações críticas: reduziu a complexidade computacional de $O(MN^3)$ para $O(MN^2)$, implementou elitismo ao combinar populações pai e descendente, e eliminou a necessidade de parâmetros de compartilhamento através de um operador de comparação populacional que preserva a diversidade.

O planejamento de rotas turísticas no Rio de Janeiro representa um desafio particular devido à diversidade de atrações distribuídas pela cidade, muitas das quais permanecem pouco exploradas mesmo pelos moradores locais. Como Machado [6] observa em seu estudo sobre a evolução do turismo na cidade, o Rio possui uma longa tradição de visitação que remonta ao período imperial, tendo atraído admiração por sua paisagem desde o século XIX. Atualmente, o problema envolve não apenas identificar pontos turísticos relevantes, mas otimizar roteiros considerando as limitações de tempo e orçamento dos visitantes.

¹augustomp@id.uff.br

²filipe.sousa@pos.ime.uerj.br

³imcoelho@ic.uff.br

Para viabilizar a modelagem do problema de roteirização turística, foram utilizadas duas principais fontes de dados. O TripAdvisor [12] forneceu dados estruturados sobre as atrações cariocas. A partir do ranking de popularidade da plataforma, foram selecionadas as 40 atrações mais bem classificadas do Rio de Janeiro, extraindo-se informações de custos, tempos médios de visitação e horários de funcionamento. Para as informações de deslocamento entre atrações, o estudo utilizou dados do OpenStreetMap [8]. Por meio do OSRM e do serviço Mapbox [7], foram obtidas as matrizes de tempo e distância para deslocamentos de carro e a pé entre todas as atrações mapeadas. A formulação dos custos de transporte baseou-se em uma análise dos preços praticados por empresas de transporte por aplicativo na cidade, chegando-se ao valor médio de R\$6,00 por quilômetro percorrido.

Com base nessas fontes de dados, nosso estudo se propõe a três objetivos fundamentais: implementar e avaliar o algoritmo NSGA-II para resolver o problema de roteirização turística multiobjetivo; quantificar a qualidade das soluções geradas através da métrica de hipervolume; e disponibilizar um aplicativo web de código aberto desenvolvido com Flask [9] e Dash [10] que permita aos usuários navegar intuitivamente entre as diversas soluções otimizadas.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica sobre problemas de otimização multiobjetivo; a Seção 3 detalha a metodologia adotada, incluindo a modelagem do problema; a Seção 4 expõe os resultados obtidos; e, por fim, a Seção 5 apresenta as conclusões.

2 Referencial Teórico

Algoritmos genéticos, formalizados por Holland [5], representam técnicas de otimização inspiradas na evolução biológica que abordam problemas complexos através de populações de soluções que evoluem por seleção, cruzamento e mutação. Esta analogia com processos naturais tem sido aplicada no contexto turístico, como demonstrado por Choi et al. [2], que utilizaram estas técnicas para otimizar o tempo total dos itinerários. Contudo, o problema de roteirização turística mostra-se complexo, propício para o uso de algoritmos multiobjetivos.

A área de otimização multiobjetivo evoluiu significativamente desde o trabalho de Schaffer [11], que desenvolveu o Vector Evaluated Genetic Algorithm (VEGA), até estudos de Zitzler, Deb e Thiele [15], que evidenciou que nenhuma abordagem dominava completamente as demais em todos os cenários, ressaltando a importância de escolher algoritmos adequados para cada contexto específico. De particular relevância, Arbolino et al. [1] utilizaram otimização multiobjetivo no planejamento sustentável do turismo, comprovando a superioridade desta técnica.

O NSGA-II, desenvolvido por Deb e colaboradores [3], consolidou-se como referência em otimização multiobjetivo ao superar limitações críticas: reduziu a complexidade computacional de $O(MN^3)$ para $O(MN^2)$, implementou um operador de comparação de aglomeração que preserva a diversidade populacional sem parâmetros adicionais, e adotou uma estratégia elitista. Seu desempenho competitivo justifica sua ampla adoção.

A avaliação de algoritmos de otimização multiobjetivo apresenta desafios significativos, como demonstrado por Zitzler et al. [16], que identificaram a insuficiência de indicadores unários e a necessidade de métricas binárias para capturar relações de dominância entre soluções. O hipervolume destaca-se como métrica Pareto-conforme, apesar de sua complexidade exponencial tradicional, limitação enfrentada pelo algoritmo HSO (*Hypervolume by Slicing Objectives*) de While et al. [14], que alcança eficiência superior através do processamento sequencial de objetivos, complementado por Wang et al. [13] com técnicas de normalização adaptativa. Em nosso trabalho, implementamos ambas as abordagens para avaliar a qualidade das soluções e aprimorar o processo de busca do NSGA-II, garantindo cobertura mais uniforme da fronteira de Pareto para o problema de roteirização turística.

3 Metodologia

O problema de planejamento de rotas turísticas é formulado como um problema de otimização combinatória multiobjetivo. Uma solução consiste em um roteiro de visitas às atrações turísticas, definindo a sequência de visitas, os modos de transporte entre elas e os horários de chegada e partida em cada atração do Rio de Janeiro.

O modelo tem quatro funções objetivo (Equações 1-4): minimizar custo (Z_1), tempo (Z_2), e maximizar atrações (Z_3) e bairros distintos visitados (Z_4):

$$Z_1 = \sum_{i,j \in A} x_{ij}(e_i + c_{ij}(m_{ij})) \quad (\text{Custo total}) \quad (1)$$

$$Z_2 = \sum_{i,j \in A} x_{ij}(t_{ij}(m_{ij}) + v_i) + \sum_{i \in A} w_i \quad (\text{Tempo total}) \quad (2)$$

$$Z_3 = - \sum_{i \in A} \min(1, \sum_{j \in A} x_{ij}) \quad (\text{Qtd. atrações}) \quad (3)$$

$$Z_4 = -|\{b_i | i \in A, \sum_{j \in A} x_{ij} \geq 1\}| \quad (\text{Diversidade bairros}) \quad (4)$$

Onde $c_{ij}(m_{ij}) = 0$ se caminhada e R\$6/km se carro.

As principais variáveis e parâmetros do modelo são:

t_{ij}^{walk} Tempo de deslocamento a pé entre atrações i e j .

d_{ij} Distância em metros entre atrações i e j .

$0, n$ Índices especiais representando ponto de partida e chegada do roteiro.

k Índice das atrações intermediárias (exclui o ponto de partida 0 e o destino final n).

x_{ij} Variável binária indicando se o turista vai da atração i para a j (1 sim, 0 não).

m_{ij} Variável binária para modo de transporte entre i e j : caminhada (0) ou carro (1).

a_i Horário de chegada na atração i .

d_i Horário de partida da atração i .

t_{ij} Tempo de deslocamento entre i e j com transporte escolhido.

c_{ij} Custo do deslocamento entre i e j com transporte escolhido.

v_i Tempo de visita da atração i .

w_i Tempo de espera na atração i quando chegada ocorre antes da abertura.

e_i Custo de entrada na atração i .

b_i Bairro onde está localizada a atração i .

O_i Horário de abertura da atração i .

F_i Horário de fechamento da atração i .

O modelo é sujeito às oito restrições a seguir (Equações 5-11), apresentadas na Tabela 1:

3.1 NSGA-II

Nossa implementação do Algoritmo 1 utiliza: (i) seleção por torneio com operador de comparação por dominância e aglomeração (linhas 6-7); (ii) cruzamento específico para sequências de atrações (linha 7); (iii) mutação que adiciona/remove/reordena atrações (linha 7); (iv) mecanismo de penalização para rotas inviáveis durante a avaliação (linha 2). O algoritmo evolui ao longo de G gerações (linha 5), combinando populações pai e descendente (linha 8) e aplicando ordenação não-dominada rápida (linha 9) para produzir um conjunto diverso de roteiros turísticos Pareto-ótimos (linha 17).

Tabela 1: Conjunto de restrições do modelo de roteirização turística (Equações 5-11).

(5)	$\sum_{i,j \in A} x_{ij}(t_{ij}(m_{ij}) + v_i) + \sum_{i \in A} w_i \leq 840 \quad (\text{Tempo diário})$
(6)	$O_i \leq a_i \leq d_i \leq F_i, \quad d_i - a_i \geq v_i \quad \forall i \in A : \sum_{j \in A} x_{ij} \geq 1 \quad (\text{Funcionamento})$
(7)	$a_j \geq d_i + t_{ij}(m_{ij}) \quad \forall i, j \in A : x_{ij} = 1 \quad (\text{Sequencialidade})$
(8)	$\sum_{j \in A} x_{ij} \leq 1, \quad \sum_{i \in A} x_{ij} \leq 1 \quad \forall i, j \in A \quad (\text{Unicidade})$
(9)	$m_{ij} = \begin{cases} 0, & t_{ij}^{\text{walk}} \leq 15, \\ 1, & \text{caso contrário,} \end{cases} \quad \forall i, j \in A : x_{ij} = 1 \quad (\text{Preferência transporte})$
(10)	$c_{ij}(m_{ij}) = \begin{cases} 0, & m_{ij} = 0, \\ 6 \cdot \frac{d_{ij}}{1000}, & m_{ij} = 1, \end{cases} \quad \forall i, j \in A : x_{ij} = 1 \quad (\text{Custo carro})$
(11)	$\sum_{i \in A} x_{i0} = 1, \quad \sum_{j \in A} x_{nj} = 1, \quad \sum_{i \in A} x_{ik} = \sum_{j \in A} x_{kj} \quad \forall k \in A \setminus \{0, n\} \quad (\text{Continuidade})$

Algorithm 1 Algoritmo NSGA-II adaptado para roteirização turística

```

1: Inicializar população  $P_0$  de tamanho  $N$ 
2: Avaliar objetivos (custo, tempo, atrações, bairros)
3: Classificar  $P_0$  por não-dominância:  $F = (F_1, F_2, \dots)$ 
4: Calcular distância de aglomeração em cada  $F_i$ 
5: for  $t = 0$  até  $G - 1$  do
6:   Selecionar pais de  $P_t$  via torneio binário
7:   Gerar  $Q_t$  (tamanho  $N$ ) via cruzamento e mutação
8:    $R_t = P_t \cup Q_t$  (tamanho  $2N$ )
9:    $F = (F_1, F_2, \dots) = \text{Ordenação-Não-Dominada}(R_t)$ 
10:   $P_{t+1} = \emptyset$  e  $i = 1$ 
11:  while  $|P_{t+1}| + |F_i| \leq N$  do
12:    Calcular distância de aglomeração em  $F_i$ 
13:     $P_{t+1} = P_{t+1} \cup F_i$  e  $i = i + 1$ 
14:  end while
15:  Ordenar  $F_i$  por distância de aglomeração decrescente
16:  Incluir primeiros  $(N - |P_{t+1}|)$  elementos de  $F_i$  em  $P_{t+1}$ 
17: end for
18: return Soluções não-dominadas de  $P_G$ 
    
```

4 Resultados

Os experimentos foram realizados utilizando dados reais de 40 atrações turísticas do Rio de Janeiro, selecionadas com base no ranking do *TripAdvisor* [12]. Esta base inclui pontos turísticos distribuídos por 22 bairros, com tempos de visita entre 60 e 240 minutos e custos variando de gratuitos a R\$ 844,37. As matrizes de distância e tempo foram construídas utilizando o *Open Source Routing Machine* (OSRM) [8] e o serviço *Mapbox* [7], fornecendo dados precisos para deslocamentos a pé e de carro.

Os experimentos foram executados em um processador Intel i7-11700K com 16GB de RAM no sistema operacional Ubuntu 22.04 LTS. O algoritmo NSGA-II foi implementado em C++17. Após testes iniciais, foi configurado com população de 100 indivíduos evoluindo por 100 gerações, utilizando taxa de cruzamento de 0,9 e taxa de mutação de 0,1. Realizamos 30 execuções independentes para garantir robustez estatística.

Em média a execução do NSGA-II resultou em conjuntos de cerca de 70 soluções não-dominadas. A qualidade foi avaliada através da métrica de hipervolume, implementada conforme o algoritmo HSO [14]. O conjunto final alcançou um hipervolume normalizado médio de 0,163, após aplicação de técnicas de normalização adaptativa de Wang et al. [13].

As soluções obtidas apresentam uma ampla gama de resultados: desde roteiros econômicos com

média de R\$ 230,21 visitando 8 atrações em 7 bairros, até roteiros mais dispendiosos com média de R\$ 512,24 que incluem 9 atrações em 8 bairros. A diversidade de soluções permite aos turistas escolherem roteiros alinhados com suas preferências (Figura 1).

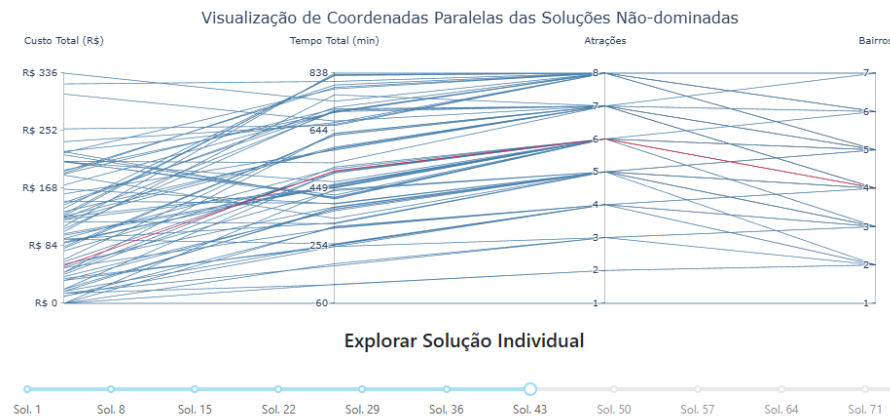


Figura 1: Visualização de coordenadas paralelas das soluções não-dominadas. Fonte: Os autores.

Para facilitar a exploração das soluções geradas, desenvolvemos um protótipo de aplicativo web denominado "Um Dia no Rio", que permite ao usuário visualizar detalhadamente cada roteiro otimizado, como exemplificado na Figura 2.

5 Conclusão

Este trabalho apresentou uma abordagem de otimização multiobjetivo para o planejamento de rotas turísticas no Rio de Janeiro utilizando o algoritmo NSGA-II. Os três objetivos propostos foram alcançados: implementamos e avaliamos o NSGA-II para roteirização turística multiobjetivo, adaptando os operadores genéticos para o contexto de sequenciamento de atrações; quantificamos a qualidade das soluções através da métrica de hipervolume; e desenvolvemos o aplicativo web "Um Dia no Rio" que permite aos usuários navegarem entre as soluções otimizadas.

O método proposto mostrou-se capaz de gerar roteiros turísticos de um dia que equilibram quatro objetivos conflitantes: minimização dos custos e do tempo de deslocamento, e maximização das atrações visitadas e da diversidade de bairros. Os roteiros respeitam restrições práticas essenciais. Os resultados demonstraram a existência de diversas soluções não-dominadas, oferecendo ao turista opções desde roteiros econômicos até itinerários que maximizam quantidade e variedade de locais visitados.

Todo o código-fonte desenvolvido neste trabalho está disponível publicamente no repositório GitHub <https://github.com/augustompm/OM-de-Rotas-Turisticas-Um-Dia-no-Rio>, incluindo a implementação do algoritmo NSGA-II, as estruturas de dados para representação do problema, as ferramentas de análise de hipervolume e o aplicativo web interativo. O arquivo README.md contém instruções detalhadas para execução tanto do aplicativo web (pasta app/) quanto do algoritmo de otimização.

Como trabalhos futuros, identificamos diversas direções promissoras, tais como explorar metodologias alternativas de otimização multiobjetivo, como MOEA/D ou SPEA2, e comparar seu desempenho com o NSGA-II ou incorporar objetivos adicionais, como a preferência pessoal do turista por tipos específicos de atrações.

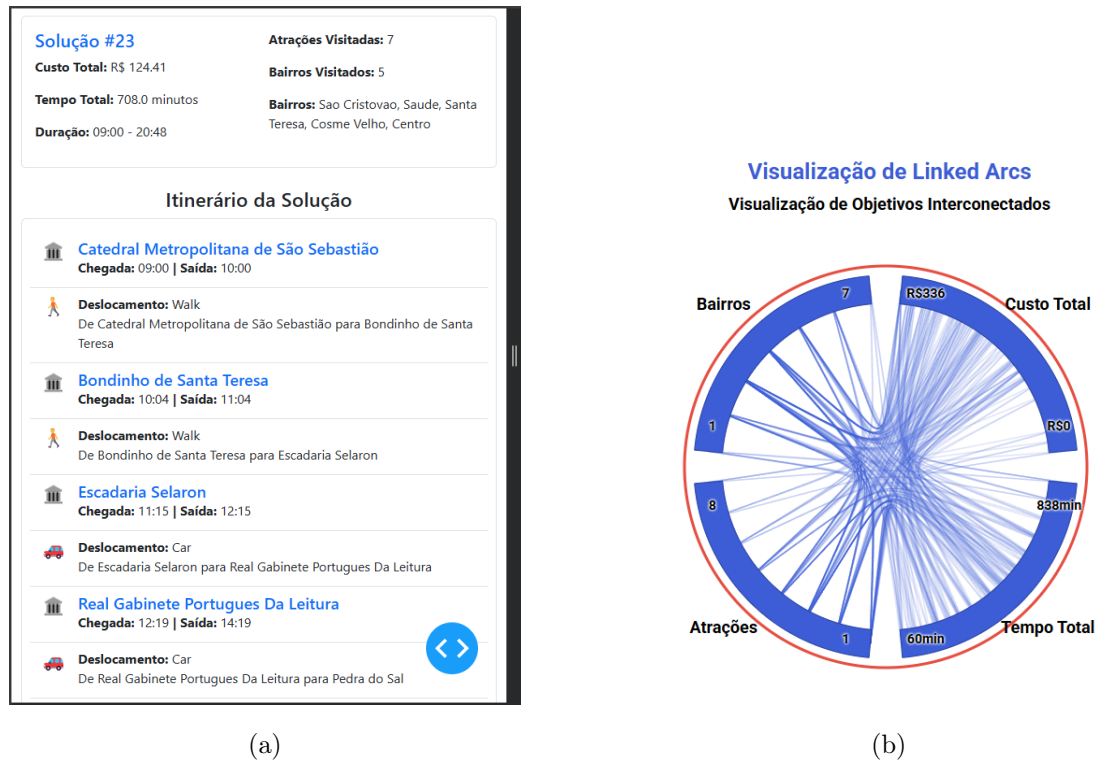


Figura 2: Interface do aplicativo "Um Dia no Rio": (a) apresentação de um itinerário otimizado; (b) visualização de arcos vinculados entre os objetivos, adaptado de [4]. Fonte: Os autores.

Referências

- [1] R. Arbolino, R. Boffardi, L. De Simone e G. Ioppolo. "Multi-objective optimization technique: A novel approach in tourism sustainability planning". Em: **Journal of Environmental Management** 285 (2021), p. 112016.
- [2] K.-C. Choi, S. Li, C.-T. Lam, A. Wong, P. Lei, B. Ng e K.-M. Siu. "Genetic Algorithm For Tourism Route Planning Considering Time Constrains". Em: **International Journal of Engineering Trends and Technology** 70.1 (2022), pp. 170–178.
- [3] K. Deb, A. Pratap, S. Agarwal e T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". Em: **IEEE Transactions on Evolutionary Computation** 6.2 (2002), pp. 182–197.
- [4] R. Haghazadeh. **Harmony: A Visualization Tool for Multi-Objective Optimization Results**. <https://github.com/RoozbehH/Harmony>. Acessado em: 1 março 2025. 2023. URL: <https://github.com/RoozbehH/Harmony>.
- [5] J. H. Holland. "Genetic Algorithms and the Optimal Allocation of Trials". Em: **SIAM Journal on Computing** 2.2 (1973), pp. 88–105.
- [6] M. de B. T. Machado. "As etapas evolutivas do turismo: Um estudo sobre o Rio de Janeiro (séculos XVIII-XX)". Em: **CULTUR: Revista de Cultura e Turismo** 7.1 (2013), pp. 105–127.

- [7] Mapbox. **Mapbox: Maps and location for developers**. Acesso em: 1 mar. 2025. 2024. URL: <https://www.mapbox.com/>.
- [8] OpenStreetMap Contributors. **OpenStreetMap**. Acesso em: 1 mar. 2025. 2024. URL: <https://www.openstreetmap.org/>.
- [9] Pallets. **Flask: Web development, one drop at a time**. Acesso em: 1 mar. 2025. 2024. URL: <https://flask.palletsprojects.com/>.
- [10] Plotly Technologies Inc. **Dash: Analytical Web Apps for Python**. Acesso em: 1 mar. 2025. 2024. URL: <https://dash.plotly.com/>.
- [11] J. Schaffer. “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.” Em: **Proceedings of the First Int. Conference on Genetic Algorithms**, Ed. **G.J.E Grefenstette, J.J. Lawrence Erlbraum** (jan. de 1985), pp. 93–100.
- [12] TripAdvisor. **TripAdvisor**. Acesso em: 1 mar. 2025. 2024. URL: <https://www.tripadvisor.com.br/>.
- [13] B. Wang, H. K. Singh e T. Ray. “Adjusting normalization bounds to improve hypervolume based search for expensive multi-objective optimization”. Em: **Complex & Intelligent Systems** 9.2 (2023), pp. 1193–1209.
- [14] L. While, P. Hingston, L. Barone e S. Huband. “A faster algorithm for calculating hypervolume”. Em: **Evolutionary Computation, IEEE Transactions on** 10 (mar. de 2006), pp. 29–38.
- [15] E. Zitzler, K. Deb e L. Thiele. “Comparison of multiobjective evolutionary algorithms: Empirical results”. Em: **Evolutionary computation** 8.2 (2000), pp. 173–195.
- [16] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca e V. G. da Fonseca. “Performance assessment of multiobjective optimizers: an analysis and review”. Em: **IEEE Transactions on Evolutionary Computation** 7.2 (2003), pp. 117–132.