

Geração de roteiros turísticos personalizados em tempo real através de heurística GRASP

Lucas G. S. Felix¹, Glauber S. Santos¹, Mateus R. S. Catrinque¹

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

{lucasgsfelix,glaubersoares,mcatrinque}@ufmg.br

Resumo. Um dos setores que mais se beneficiou da expansão da internet foi o turismo. A internet permitiu que as pessoas pudessem compartilhar informações de suas viagens, auxiliando outros turistas a decidir os melhores destinos. Devido à abundância de dados em redes sociais e plataformas especializadas em turismo disponíveis, se tornou cada vez mais difícil escolher quais os melhores locais para se visitar, trazendo aos usuários o problema conhecido como sobrecarga de informação. Assim, considerando que é difícil identificar um local para ser visitado, planejar uma viagem completa pode ser considerado um cenário ainda mais difícil. Visando modelar da melhor maneira este problema, os trabalhos focam na tarefa de planejamento automático de viagens, também conhecido como Problema da Orientação (PO). Contudo, este é considerado um problema computacionalmente custoso, e devido a demandas do usuário, o tempo de resposta para aplicações deve ser abaixo de 1 segundo. Dado estes fatores, neste trabalho, trazemos uma nova proposta baseada em meta-heurísticas de solução única, GRASP. Por meio de nossa proposta, conseguimos avaliar em cenários reais (Rio de Janeiro), com tempo de execução abaixo de 1 segundo, e resultados que se mostram melhores que outras heurísticas comparadas em 60 % dos casos. Link para apresentação: <https://drive.google.com/file/d/1D0BtLFuMh0T98FRcL-syLNRC4fVgUKYT/view>

1. Introdução

Antes da popularização da internet, as fontes predominantes para obter informações sobre turismo eram a troca de experiências pessoais entre indivíduos e recomendações em guias e revistas [Leite et al. 2013, Kotiloglu et al. 2017, Yoo et al. 2016], contudo, essas eram recomendações genéricas e baseadas na opinião de apenas uma pessoa. Com a expansão da internet, a indústria do turismo se revitalizou. Através da internet, os usuários puderam se reunir em comunidades online, blogs de viagens e redes sociais, permitindo a busca por Pontos de Interesse (*Points of Interest - POIs*) e a troca de informações sobre locais e experiências. Contudo, apesar dessas plataformas viagens concentrarem muitas informações relevantes para o planejamento de viagens, os usuários ainda enfrentam dificuldades devido à sobrecarga de informação, tornando a escolha da melhor opção desafiadora [Pang et al. 2008]. Uma das maneiras de lidar com o problema da sobrecarga de informação é via Sistemas de Recomendação (SR), os quais dão aos usuários sugestões personalizadas de lugares, baseado em seu histórico de visitas. No entanto, SR, no geral, não abrangem aspectos fundamentais de uma viagem turística, como preço, horário de funcionamento e distância, que também são cruciais para a experiência do usuário [Kotiloglu et al. 2017].

Para simplificar o planejamento de viagens, muitos estudos buscam automatizar esse processo, abordando-o como um problema de otimização. No geral, este problema turístico pode ser mapeado como uma instância do Problema de Orientação (PO), a qual é descrito como uma combinação do problema do caixeiro-viajante e o problema da mochila [Gavalas et al. 2014]. De maneira geral, o PO busca encontrar uma rota em um grafo que maximize a utilidade, respeitando uma restrição de distância [Vansteenwegen and Van Oudheusden 2007, Gavalas et al. 2014]. Entretanto, ao considerar cenários reais de turismo, apenas essas restrições não são capazes de satisfazer os requerimentos de usuários, assim, diversos trabalhos na literatura expandem

essa modelagem considerando restrições de tempo [Vansteenwegen et al. 2011], múltiplos dias [Vansteenwegen et al. 2011], e categóricas [Sarkar and Majumder 2021].

A formulação do problema pode ser apresentada da seguinte forma: seja $G = (V, E)$ um grafo no qual cada vértice $v \in V$ possui associado um peso $w_v \in \mathbb{R}_+$ de lucros, onde $W = \{w_0, w_1, \dots, w_v\}$. Dado um nó inicial s , um nó terminal f , onde $s, f \in V$, e um orçamento de tempo positivo B , o objetivo é encontrar um caminho de s para f (ou seja, $s = f$) com comprimento no máximo B , de modo que o lucro total dos nós visitados seja maximizado.

Vale ressaltar, que apesar deste problema ter uma grande aplicação prática, o PO é um problema NP-difícil [Gunawan et al. 2016], o que faz encontrar boas soluções em tempo polinomial uma tarefa não trivial. Além disso, visto que este problema pode ser incorporado em aplicações que demandam resposta em tempo real ao usuário (e.g. redes sociais), o desenvolvimento de técnicas que geram boas soluções, mas pecam no tempo computacional, não atende as demandas dos usuários, os quais estão dispostos a aguardar no máximo 1 segundo para terem uma resposta [Headspin 2023]. Dessa maneira, técnicas que possuem um bom desempenho computacional são essenciais, mesmo que isso signifique prescindir da qualidade da solução gerada. Neste cenário, na literatura, heurísticas e meta-heurísticas tem sido extensivamente utilizadas para resolver este problema, aplicando diferentes técnicas.

Apesar das diversas soluções e técnicas presentes na literatura, estas falham em avaliar suas propostas em cenários reais. Por exemplo, há trabalhos que não consideram a seleção de hotéis e múltiplos dias de viagem em suas modelagens [Lu et al. 2012, Wibowo and Handayani 2018, Chen et al. 2020, Sarkar and Majumder 2021], consequentemente tendo pouco uso prático, visto que turistas viajam pelo menos 2 dias [Friggstad et al. 2018]. O trabalho proposto em [Kotiloglu et al. 2017], faz um avanço na literatura preenchendo alguns destes problemas, visto que permite a seleção automática de hotéis, e é modelado para múltiplos dias. Contudo, esta última falha em avaliar cenários práticos para seus usuários, visto que o conjunto de dados utilizados no trabalho não apresenta alguns atributos do local, como, por exemplo, o preço, consequentemente podendo estar avaliando cenários não realistas.

Dado estes fatores, nosso objetivo com este trabalho é propor uma meta-heurística que resolve o problema do turista em um cenário real. Para isso, propomos uma heurística GRASP (*Greedy Randomized Adaptive Search Procedure*), tendo foco em qualidade de solução e desempenho computacional para resolução do problema, visando a aplicação em cenários reais. Visando um cenário real de aplicação, nós avaliamos nossa proposta em uma base de dados da cidade do Rio de Janeiro, coletada da plataforma *TripAdvisor*. A cidade foi escolhida devido ao seu grande apelo turístico, tendo diversos locais, em diferentes regiões, para serem visitados.

Por fim, definimos como principais objetivos:

- Verificar se é possível realizar a recomendação de roteiros turísticos, por meio da utilização de meta-heurísticas, os quais respeitem restrições de tempo dos usuários para uma aplicação em tempo real.
- Verificar se é possível realizar a recomendação de roteiros turísticos, por meio da utilização de meta-heurísticas baseadas em solução única, que gerem boas soluções em termos de utilidade.

Os nossos resultados mostram que por meio de nossa proposta é possível tanto gerar bons roteiros, os quais são melhores que outros *baselines* em 60% dos cenários avaliados, quanto respeitar as restrições de aplicabilidade em tempo real, no nossa proposta possui no pior cenário um tempo de execução de 0.84 segundos.

O restante desse trabalho está dividido da seguinte forma: na seção 2 apresentamos os trabalhos que estão relacionados ao nosso, na seção 3 apresentamos a metodologia proposta para resolver o OP, na seção 4 são apresentados os resultados de nossa proposta, e por último, na seção 5 são apresentadas as conclusões.

2. Trabalhos Relacionados

No contexto do turismo, o Problema de Orientação (PO) é conhecido como Problema de Design de Viagem Turística (*Tourism Trip Design Problem*). Ele visa sugerir roteiros que maximizem a utilidade, respeitando as restrições dos usuários e locais. Na literatura, esse problema foi abordado sob diferentes perspectivas, considerando várias restrições e técnicas para resolvê-lo. Nesta seção, damos uma breve visão dos trabalhos que tratam o problema do turista sobre a perspectiva de meta-heurísticas baseadas em solução única e em meta-heurísticas populacionais.

O trabalho [Vansteenwegen et al. 2011], propõe uma abordagem baseada no *GRASP* para o problema do turista, tendo como principal foco a avaliação com usuários reais. Dentre os principais resultados, os autores realizam avaliação com diferentes usuários por meio de uma aplicação web. Em suas conclusões os autores afirmam a qualidade do *GRASP* para a resolução do problema, tendo qualidade e um bom desempenho computacional, visto que estes não foram pontos de reclamação dos usuários.

As propostas feitas por [Liao and Zheng 2018, Wibowo and Handayani 2018, Zheng and Liao 2019, Yochum et al. 2020], utilizam-se de algoritmos populacionais evolutivos para resolver o problema do turista. Os trabalhos de [Liao and Zheng 2018, Wibowo and Handayani 2018, Yochum et al. 2020], utilizam de algoritmos genéticos para resolver o problema. Enquanto o trabalho de [Zheng and Liao 2019], faz uma comparação entre os algoritmos de diferença evolutiva e otimização por colônia de formigas. Além disso, em alguns cenários, as operações evolutivas podem gerar candidatos que não são viáveis. Além do custo computacional para a avaliação das restrições em cada uma das soluções pode ser muito grande, fazendo consequentemente com que este tipo de técnica não seja a melhor para cenários onde se espera soluções em um tempo muito curto.

Os trabalhos feitos por [Gavalas et al. 2015, Brilhante et al. 2015, Gavalas et al. 2017, Chen et al. 2020], se utilizam de técnicas baseadas em busca local iteradas para resolver o problema do turista. Enquanto os trabalhos [Gavalas et al. 2015, Gavalas et al. 2017], focam em aspectos da iteração do usuário com aplicações para análise. O trabalho de [Brilhante et al. 2015, Chen et al. 2020], tem um foco em comparar as rotas geradas para os usuários, com as rotas feitas previamente por estes turistas. Contudo, vale ressaltar que os usuários, no geral, não realizam escolhas ótimas, o que faz com que as rotas recomendadas sejam melhor quando personalizadas em um passo anterior a otimização [Kotiloglu et al. 2017].

O trabalho de [Kotiloglu et al. 2017] usa técnicas de busca tabu iterativa para recomendar rotas, com um foco semelhante ao nosso: maximizar a utilidade do usuário e o desempenho computacional. Eles utilizam dados do *Foursquare*, mas esses dados, embora sejam abrangentes, não têm detalhes suficientes sobre preços reais, o que os leva a usar dados sintéticos em suas avaliações. No entanto, o trabalho oferece uma modelagem aprofundada dos requisitos dos turistas, considerando aspectos como horários de funcionamento de locais, seleção de hotéis e viagens de múltiplos dias. Como nosso escopo é mais restrito, fazemos uma comparação com a proposta deles ao remover restrições de horário de funcionamento e locais obrigatórios, priorizando soluções que equilibrem utilidade e tempo de resposta rápida.

3. Metodologia

3.1. Restrições do problema

Considere que um turista (usuário) está viajando, por uma cidade, por p dias, visando visitar o máximo de m atrações turísticas todos os dias. O objetivo desse usuário é criar passeios diários em locais que podem ser visitados durante os p dias. Cada local i está associado a uma utilidade $R_i \in \mathbb{R}_+$, que pode ser visto como a nota que aquele usuário deu ao local. Assim, nosso objetivo é encontrar o subconjunto de locais disponíveis que **maximize** a utilidade dos locais visitados, enquanto respeitando as restrições do usuário. As seguintes restrições são consideradas em nosso problema:

- **Restrição Monetária:** Define o quanto o usuário pode gastar ao longo de todos os dias.
- **Restrição de Distância Diária:** Define o quanto o usuário pode andar em quilômetros por dia.
- **Restrição de Quantidade de Atividades Diárias:** Define a quantidade máxima de atividades a serem realizadas em um dia.
- **Restrição de Categoria:** Define a quantidade máxima de locais de uma mesma categoria podem ser visitados sequencialmente por dia.
- **Restrição de Múltiplas Visitas:** Define que nenhum local, com exceção do hotel, pode ser visitado mais de uma vez ao dia.

As restrições apresentadas tem sua formulação apresentada no trabalho [da Silva Felix et al. 2022].

3.2. Solução proposta

Dado o fato que técnicas populacionais são abordagens que aprofundam nos quesitos de intensidade e diversidade, em troca de desempenho computacional, tais apesar de gerar bons resultados para o problema do turista [Liao and Zheng 2018, Wibowo and Handayani 2018, Zheng and Liao 2019, Yochum et al. 2020], no geral não atendem os requisitos necessários para aplicações em tempo real. Neste caso, técnicas baseadas em soluções únicas como busca local, busca tabu, e *GRASP* possuem um desempenho computacional melhor [Talbi 2009], apesar de perderem na diversidade das soluções geradas. Contudo, dentre todas essas técnicas de solução única, *GRASP* se destaca por ter uma natureza aleatória que permite um aprofundamento, e consequentemente gerar soluções que possuem mais diversidade que outras técnicas de solução única que não possuem nenhum princípio estocástico. Com isso, dado estes fatores, neste trabalho, utilizamos o *GRASP* para a resolução do problema do turista.

O *GRASP* é uma meta-heurística proposta por [Feo and Resende 1995], que combina elementos de construção gulosa com busca local para resolver problemas de otimização combinatória [Talbi 2009]. Na Figura 1, é apresentada a maneira que cada uma das soluções geradas pela heurísticas é representada, tendo de maneira fixa em sua estrutura, na primeira posição, o hotel e nas outras posições os locais que serão visitados.

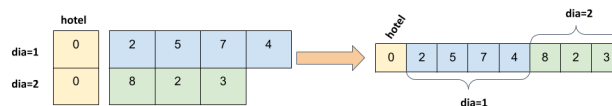


Figura 1. Representação da solução

De maneira geral, o *GRASP* pode ser descrito da seguinte forma:

1. **Construção Gulosa Aleatorizada:** Começa com uma solução vazia e, de maneira gulosa, adiciona elementos ao conjunto de soluções, mas com um grau de aleatoriedade controlado pelo parâmetro α . Isso permite explorar diferentes soluções iniciais. A cada iteração, os elementos que podem ser incluídos na solução parcial são ordenados na lista (em ordem decrescente de valor) usando uma heurística local. A partir dessa lista, é gerado um subconjunto que representa a lista de candidatos restritos (LCR).
2. **Busca Local:** Após a construção, aplica-se uma busca local para melhorar a solução encontrada na etapa anterior. A busca local explora soluções vizinhas em busca de melhorias. Em nosso cenário, a busca local visa trocar todos os locais que não possuam a nota máxima (5). Visando manter a viabilidade da solução gerada, selecionamos para substituição locais que possuem a mesma categoria, e que são próximos ao local que será substituído, mas que tenham uma nota maior. Neste caso, ordenamos as melhores soluções pela nota do local, testando os novos locais que podem

ser atribuídos. Este processo é repetido até um novo local, que melhora a qualidade da solução, seja encontrado. Caso nenhum dos candidatos selecionados melhore a solução e a deixe viável, então passamos para o próximo local que possui nota menor que 5.

3. **Iterações:** Repete as etapas de construção e busca local por um número definido de iterações, mantendo a melhor solução encontrada.

Neste trabalho propomos uma abordagem para a seleção de locais em cenários urbanos complexos, como grandes cidades, visando resolver o problema do turista. A proposta consiste em uma ordenação da Lista de Candidatos Restritos (LCR) com base na distância entre locais e suas notas de interesse. Essa estratégia prioriza locais próximos com notas mais altas, buscando evitar soluções inviáveis em trajetos longos na cidade, dado a seguinte fórmula $c_j = \frac{D_{i,j}}{R_j}$. Assim, essa técnica visa encontrar um equilíbrio entre proximidade e valor dos locais, proporcionando soluções mais viáveis e relevantes para o planejamento de viagens em ambientes urbanos complexos, o qual é dado pela fórmula [1]

$$\beta = c^{min} + \alpha(c^{max} - c^{min}) \quad (1)$$

A seleção dos valores de α é realizada adaptativamente em um intervalo determinado por valores de α e β , calculados conforme uma abordagem proposta anteriormente na literatura [Talbi 2009].

3.3. Propostas de controle (Baselines)

Visando comparar os resultados da heurística proposta neste trabalho com diferentes técnicas para resolver o mesmo problema, implementamos 4 diferentes técnicas, e também apresentamos um limite superior teórico para as instâncias que temos. De maneira geral, os algoritmos comparados ao nosso podem ser descritos da seguinte maneira:

- **Aleatório (Random):** O algoritmo aleatório proposto gera soluções de forma estocástica até encontrar uma solução viável. O algoritmo gera uma solução aleatória considerando todos os hotéis e locais disponíveis. Desses hotéis, 1 é selecionado. Dos locais, são selecionados de maneira aleatória a quantidade máxima de locais que podem ser visitados pelo usuário a cada dia. Após feito isso a nova solução é avaliada, caso ela não seja viável, são removidos para cada um dos dias inviáveis os últimos locais a serem visitados. Esse processo é feito enquanto a solução não é viável. Embora essa abordagem seja eficiente, é considerada ingênua, pois não garante a geração de boas soluções. No entanto, ela serve para estabelecer um limite inferior para a qualidade das soluções produzidas.
- **Guloso (Greedy):** O algoritmo guloso proposto segue a lógica de um turista se perguntando "qual é o melhor lugar viável que posso visitar agora?". Sendo assim, o algoritmo guloso é uma abordagem construtiva que seleciona a melhor solução candidata viável disponível no momento. No momento em que ao adicionar um local a solução deixa de ser viável, então o turista volta para o hotel. Assim como o algoritmo aleatório, essa abordagem também é considerada ingênua e oferece um limite comparativo para a qualidade da solução gerada.
- **Solver Comercial:** Modelamos a função objetivo e restrições no *Solver* CPLEX. Este pode ser considerado um limite superior para qualidade de soluções, visto que *solvers* comerciais tendem a gerar próximas ao ótimo, porém, com um alto custo computacional.
- **Baseline da Literatura:** Por fim, implementamos o ITS proposto no trabalho [Kotiloglu et al. 2017]. Implementamos uma versão remove as restrições de janela de tempo, e locais obrigatórios proposta no *baseline*.

Por último temos um *baseline* teórico, o qual nos permite calcular um limite superior para qualidade de soluções, dado pela fórmula [2]. Nela temos que o limite teórico para a rota gerada é dada se o usuário visitasse a quantidade máxima de locais em um dia ao longo (m) de todos seus dias de viagem (p), e que todos os locais que o mesmo visitou tivessem

nota máxima (nota 5). Esse valor é acrescido de 5, sendo esta a nota do hotel que é contado apenas uma vez.

$$teo = (5 \times p \times m) + 5 \quad (2)$$

3.4. Procedimentos experimentais

Coleta de dados

Para nossa avaliação experimental, escolhemos a cidade do Rio de Janeiro, uma metrópole com uma vasta gama de pontos turísticos. Desenvolvemos um crawler para coletar dados do TripAdvisor, incluindo informações (preço, categoria) sobre hotéis, restaurantes e atrações, como praias, teatros e pontos de interesse como o Morro da Urca e o Corcovado. A coleta resultou em 648 hotéis, 564 restaurantes e 959 atrações, abrangendo 110 subcategorias de atrações diferentes. Esses dados inicialmente não estruturados foram convertidos em um formato semi-estruturado (por exemplo, CSV) após o parsing e pré-processamento das páginas coletadas. Para calcular as distâncias entre esses locais, utilizamos a biblioteca OpenStreet Maps, que considera rotas reais em vez de distâncias geodésicas. Em média, a distância entre os pontos é de cerca de 27 quilômetros. Este processamento foi realizado offline para a preparação dos dados.

Perfis de usuários

Para realização da avaliação experimental, foi considerado um turista viajando a cidade do Rio de Janeiro. Para isso consideramos diferentes combinações de parâmetros que chamamos de perfis. Os parâmetros podem ser categorizados em 2: fixos, e variáveis. Os fixos incluem a duração da viagem (2 dias), número máximo de locais diários (5 locais) e visitas consecutivas por categoria (3 visitas). Já os variáveis são os quilômetros por dia e o orçamento, criando diferentes perfis de usuário, os quais são apresentados na Tabela 1.

Perfil	Orçamento Monetário	Quantidade de KMs diários	Descrição
A	500 ¹	10	Gasto pouco, anda pouco
B	1000	10	Gasto médio, anda pouco
C	2500	10	Gasto muito, anda pouco
D	500	20	Gasto pouco, anda muito
E	1000	20	Gasto médio, anda muito
F	2500	20	Gasto muito, anda muito

Tabela 1. Diferentes perfis testados

Parametrização das técnicas Para o *GRASP*, no geral temos dois parâmetros principais, o número de iterações, e o α para a seleção na LRC. Dado que usamos um alfa adaptativo, para quantidade de iterações nós testamos com 10, 50, e 100 iterações, porém, para as duas avaliações, caso tenhamos 10 iterações onde não haja uma melhoria na qualidade da melhor solução, e o algoritmo é finalizado mais cedo.

No método proposto por [Kotiloglu et al. 2017], a busca tabu utiliza dois parâmetros principais: o tamanho da lista tabu e o mecanismo de perturbação. O tamanho da lista tabu evita revisitar locais já visitados por um número fixo de iterações, enquanto o mecanismo de perturbação remove aleatoriamente um subconjunto de locais com base em uma probabilidade. O estudo de [Kotiloglu et al. 2017] sugere valores padrão de lista tabu igual a 40 e probabilidade de remoção igual a 0.5. Para este trabalho, além desses valores padrão, foram utilizados um tamanho de lista tabu de 20 e probabilidade de remoção de 0.25.

Avaliação das técnicas Avaliamos as técnicas nos orientando por duas métricas principais: a qualidade das soluções geradas por cada uma das técnicas, e pelo tempo gasto para gerar cada uma das técnicas. Para realizar a avaliação, cada um dos algoritmos foram executados 100 vezes, e são comparados utilizando um teste-t pareado com 95% de confiança,

além da correção de Bonferroni devido à comparação de múltiplos algoritmos. Vale ressaltar que iremos realizar tal teste para cada um dos perfis apontados acima. Além disso, para o algoritmo *Greedy*, por ser um algoritmo exato, não temos variação na utilidade, apenas no tempo de execução.

4. Resultados

Nesta seção nós apresentamos os resultados alcançados através da pesquisa proposta. Inicialmente mostramos a parametrização e como realizamos a seleção dos melhores parâmetros. Seguindo, passamos para etapa de avaliação das técnicas que utilizamos para resolver o problema do turista, apresentando resultados com relação a tempo e a utilidade. Por último realizamos uma breve discussão sobre os resultados alcançados.

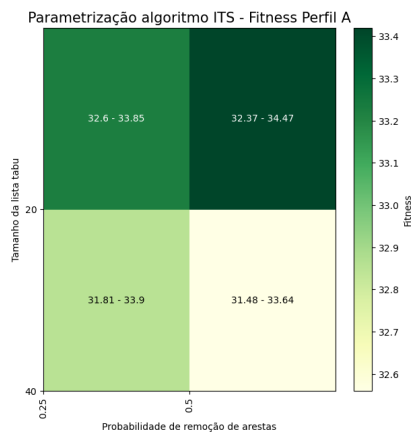
4.1. Experimento de ajuste de parâmetros

Dentre as técnicas que utilizamos neste trabalho apenas duas necessitam de parametrização: *Iterated Tabu Search* e *GRASP*. Em ambos os casos, apresentamos os resultados da parametrização com relação a utilidade para os perfis A e F visto que são os dois perfis mais distintos dentre os avaliados.

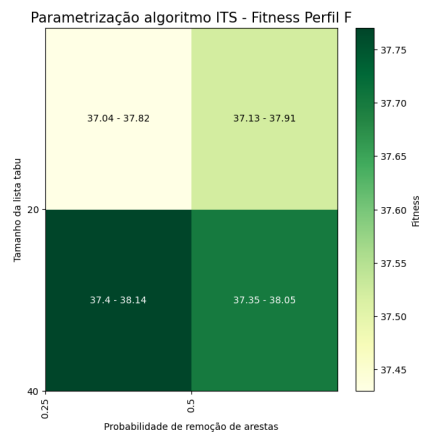
A técnica ITS é baseada na proposta de [Kotiloglu et al. 2017], e tem como principais parâmetros o tamanho da lista tabu e a probabilidade de remoção das arestas. A Figura 2, apresenta a comparação entre a combinação de todos os valores que utilizados em termos de utilidade. Em ambos os cenários, quanto mais escuro o quadro, melhor é o algoritmo. Contudo, como apresentamos na seção de metodologia, avaliamos estes resultados com relação ao intervalo de confiança. Neste cenário, o que percebemos é que para todas as combinações de parâmetros os resultados são estatisticamente equivalentes com 95% de confiança, logo, não podemos afirmar que os parâmetros possuem pouco impacto nos resultados do ITS. Acreditamos que este baixo impacto se deve ao fato das operações de busca local (e.g. *intra e inter 2opt*, *1-0 Relocate* e *1-1 Exchange*) terem um pouco efeito dentro da instância que testamos. Por exemplo, técnicas como o *2-opt* apenas realizam a redução no custo de distância da rota, o não é um objetivo para nós. A técnica de *1-0 Relocate* remove um local da rota, o que apenas reduz a mesma em custo, distância e utilidade. Por último, a técnica de *1-1 Exchange* adiciona um ponto qualquer a rota, sendo a única abordagem em pode haver um ganho em utilidade. Contudo, em um cenário como o do Rio de Janeiro, essa pode não ser a melhor estratégia, visto que a maior distância que consideramos é de 20 quilômetros por dia (Perfil F), enquanto a distância média entre todos os pontos disponíveis é de 27 quilômetros, o que faz com que em grande maioria dos casos os resultados sejam inviáveis. Por último, vale ressaltar que, dentro do contexto do trabalho de [Kotiloglu et al. 2017] as técnicas de busca local fazem total sentido, uma vez que restrições de janela de tempo também são tratadas dentro de sua proposta. Visto que os resultados são estatisticamente equivalentes, selecionamos como parâmetros padrão em nossas próximas avaliações a lista tabu com tamanho 20 e a probabilidade de remoção igual a 0.25.

A técnica *GRASP*, a qual segue nossa proposta apresentada na seção 3, possui como parâmetros o alfa, e a quantidade de iterações. Visto que o alfa é calculado de maneira automática, a quantidade de iterações é o único parâmetro que impacta em nossa proposta. A Figura 4, apresenta o impacto da quantidade de iterações nos resultados alcançados no perfil A e F. É possível perceber que ao aumentar a quantidade de iterações temos um aumento na utilidade. Neste cenário, isso ocorre dado a partida múltipla do *GRASP*, a qual com uma maior quantidade de iterações consegue avaliar melhor o espaço de busca e consequentemente encontrar soluções melhores. Como veremos na próxima seção 4.2, as resultados do *GRASP* são menos que o tempo estipulado como ideal (1 segundo), em todos os cenários. Logo, utilizamos como parâmetro em nossas próximas avaliações a quantidade de iterações igual a 100.

4.2. Avaliação das heurísticas

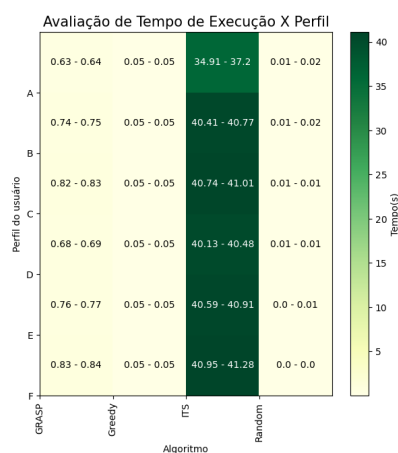


(a) Avaliação do tempo de execução pelo perfil de usuários

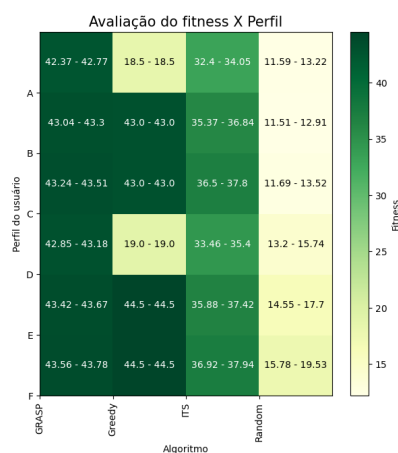


(b) Avaliação do perfil de usuários na utilidade das técnicas

Figura 2. Impacto da parametrização algoritmo ITS - Intervalo de confiança para a utilidade



(a) Avaliação do tempo de execução pelo perfil de usuários



(b) Avaliação do perfil de usuários na utilidade das técnicas

Figura 3. Avaliação da utilidade pelo perfil de usuários

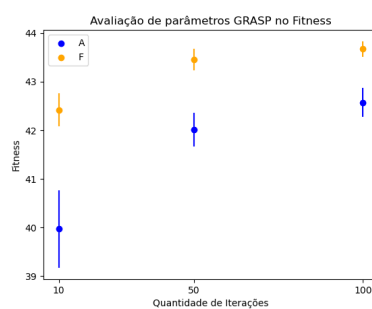


Figura 4. Impacto da parametrização algoritmo GRASP - Intervalo de confiança para a utilidade

Nesta subseção discutimos os resultados alcançados pelos algoritmos em cada um dos perfis avaliados. As Figuras 3 mostram os intervalos de confiança em relação à utilidade 3b, e tempo de execução 3a para cada uma das técnicas avaliadas em nosso trabalho. É possível perceber que os resultados dos solvers comerciais (CPLEX) não estão apresentados nas figuras. Isso é motivado pelo fato que esta técnica não foi capaz de gerar roteiros em um tempo hábil para cidade do Rio de Janeiro. Em nossa avaliação foi possível identificar que com uma quantidade de locais máxima de 400, ainda é possível gerar roteiros, contudo, ao passar disso o tempo de execução se torna muito alto, consequentemente nos motivando ainda mais pelo uso de meta-heurísticas.

Avaliação de Desempenho Computacional: Ao avaliarmos a primeira Figura 3a a qual apresenta o tempo de execução gasto por cada um dos algoritmos, podemos ver que com exceção da técnica ITS, todos os algoritmos possuem tempo de execução menor que 1 segundo, consequentemente podendo serem utilizadas em uma aplicação em tempo real. Dentre os fatores que levam a técnica ITS a ter um tempo de execução tão acima estão o fato de ser uma técnica de inicialização múltipla, mas também, o alto custo para as técnicas de busca local, e o fato de termos que avaliar todas as restrições após feita a operação de busca local. Isso faz com que o custo sob esta execução esteja mais associado a reavaliação da solução. Isso poderia ser mitigado caso fosse feita a atualização dos atributos da rota (e.g. custos, distância, quantidade de locais por categoria) ao longo da execução. Contudo, dentro de nosso entendimento, isso não é apontado explicitamente pelos autores como algo feito ao longo da busca local, logo, pode ser um ponto de futura melhoria.

As técnicas ingênuas (*Greedy*, *Random*) comparadas em nossa proposta possuem tempo de execução baixo em comparação as outras técnicas. Isso se dá pelo fato de realizarem a exploração sobre apenas uma solução. Já a técnica que propomos, *GRASP*, apesar de ter um custo de tempo elevado em comparação as técnicas ingênuas, possui tempo de execução dentro do esperado. Um dos fatores que impactam em um maior tempo de execução do *GRASP* é o fato de ser uma técnica de inicialização múltipla, que neste cenário faz com que o *GRASP* avalie 100 soluções. Além disso, podemos perceber que a medida que a quantidade de quilômetros deslocados por dia aumenta, o tempo de execução de nossa proposta também tem um acréscimo. Isso ocorre, pois, a medida que aumentamos essa característica do perfil, consequentemente elevamos a lista de soluções candidatas, o que impacta no custo computacional de nossa técnica. Por fim, para evitarmos problemas de tempo de execução em cenários em que a quantidade de quilômetros deslocados por dia aumenta, seria possível realizar a avaliação dentro do critério de seleção de locais para a LRC baseada também em custo, ou algum fator de preferência do usuário.

Avaliação da Função Objetivo (Utilidade): A Figura 3b apresenta o intervalo de confiança da utilidade de cada uma das técnicas avaliadas em nossa proposta. Neste cenário, apresentamos também como valor de referência para um ótimo teórico o resultado da Equação 2, o qual é igual a 55, sendo este o valor para uma solução ótima teórica, visto que não sabemos se há uma solução que satisfaça todas as restrições e tenha nota igual a 5.

Ao avaliarmos os resultados é possível ver que a técnica *Random* possui um baixo valor de utilidade. Isto se deve ao fato da escolha aleatória de locais levar a distâncias percorridas muito altas, dado a alta média de distância entre locais (27 quilômetros). O mesmo ocorre com as buscas locais realizadas pela técnica ITS, como discutido na subseção de parametrização. A diferença é a inicialização da meta-heurística ITS é baseada em uma abordagem aleatorizada-gulosa, assim como o *GRASP*. Logo, a solução inicial gerada já possui um resultado melhor que uma solução inicial totalmente aleatorizada. A diferença entre os resultados do ITS e do *GRASP* se dá justamente pelo fato da busca local aliada a seleção automática de alfa conseguirem gerar soluções viáveis, as quais focam nos itens que possuem uma nota baixa, substituindo-os por candidatos viáveis que possuem nota maior. Já a busca local do ITS realiza seleções aleatorizadas, as quais não levam necessariamente a substituição de locais que possuem a utilidade menor.

Por último, o algoritmo *Greedy*, apesar de ser uma heurística simples, apresenta bons resultados para cenários em que o custo monetário não é um problema. Os resultados ruins

nos perfis *A* e *D*, que possuem um orçamento de 500 reais, estão relacionados a seleção do hotel realizada pelo algoritmo. Neste caso, grande parte do orçamento é gasto na seleção do hotel, o qual é de 214 reais a diária. Neste caso, 85.6% do orçamento é gasto em hospedagem, quase não sobrando dinheiro para atrações e restaurantes. Contudo, vale ressaltar que em cenários em que um orçamento maior é disponibilizado, o algoritmo guloso consegue ter resultados um pouco superiores ao algoritmo *GRASP* para os perfis *E* e *F*, realizando um teste-t pareado com 95% de confiança.

Apesar de um bom desempenho em cenários menos restritivos, e um excelente desempenho computacional por parte da técnica *Greedy*, acreditamos no potencial de nossa técnica visto que a mesma, apesar de não obter os melhores resultados em todas as instâncias, chega a valores os quais são muito próximos do melhor resultado encontrado, e não são muito piores que o ótimo. Além disso, essa abordagem pode trazer ao usuário uma diversificação das soluções oferecidas, gerando diversas soluções que possuem ótima qualidade, e não ficando preso a apenas uma solução, como a técnica *Greedy*. Por fim, ressaltamos que a técnica proposta por nós, ainda pode ser facilmente modificada para executar em paralelo, o que pode nos resultar em tempos de execução ainda menores.

5. Conclusões

Neste trabalho, propomos uma técnica baseada em meta-heurísticas para o Problema de Orientação, também conhecido como Problema do Turista. Este problema consiste em uma combinação do problema da mochila e do problema do caixeiro-viajante e tem sido amplamente estudado no contexto do turismo. Neste cenário, o objetivo é encontrar uma rota que maximize as preferências dos usuários, respeitando as restrições de local e do usuário. O estudo propôs a meta-heurística *GRASP* e a comparou com outras abordagens como técnica exata, por meio de solver comercial(*CPLEX*) e outras heurísticas (*Greedy*, *Random*, *ITS*), utilizando dados reais da cidade do Rio de Janeiro do TripAdvisor.

Em nossos resultados, com uma base de dados contendo mais de 2000 locais, ficou claro que métodos exatos baseados em *solvers* comerciais falhavam quando o número de locais ultrapassava 400, destacando a necessidade de abordagens meta-heurísticas para resolver eficientemente esse problema em ambientes complexos e com grandes conjuntos de dados. Neste cenário, as heurísticas se destacaram com um tempo de processamento viável. Com exceção do *baseline ITS*, todas as outras três meta-heurísticas avaliadas possuem um desempenho computacional que respeita o tempo de execução imposto para aplicações reais.

Dentro dos cenários avaliados no que tange as duas principais perguntas de nosso trabalho, foi possível verificar a heurística que propomos consegue ter um tempo de execução inferior a 1 segundo, o que nos permite a aplicação em cenários reais. Contudo, outras duas heurísticas ingênuas possuem um resultado ainda melhor no quesito desempenho computacional. Entretanto, ao avaliarmos a utilidade, a qual tange o nosso segundo objetivo, os testes com cinco perfis diferentes de turistas revelaram que dois algoritmos se destacam: o *Greedy* e nossa proposta, *GRASP*. O *Greedy* funciona bem em cenários de abundância de recursos, com excelente desempenho computacional. No entanto, em situações com maior restrição financeira, o *Greedy* tende a não gerar rotas eficientes, pois gasta grande parte do orçamento em hospedagem. Por outro lado, o *GRASP*, embora não seja o melhor em todos os cenários, demonstra potencial ao apresentar os melhores resultados em três dos cinco cenários avaliados e resultados estatisticamente próximos ao melhor desempenho nos outros dois cenários. Em resumo, as heurísticas baseadas em uma única solução mostram potencial na resolução desse problema, com resultados que a candidatam a utilização em tempo real, e utilidade muito próxima ao ótimo teórico.

Como direções para trabalhos futuros, identificamos a possibilidade de explorar a eficiência da técnica *GRASP* com paralelização. Além disso, aprofundar os testes da heurística *Greedy* para identificar cenários onde essa técnica pode não ser eficaz é um caminho importante. Explorar extensões nas restrições usadas no problema do turista, como restrições de janelas de tempo, também se mostra relevante. Por fim, acreditamos que avaliar cidades maiores e avaliações personalizadas para cada local visitado.

Referências

- Brilhante, I. R., Macedo, J. A., Nardini, F. M., Perego, R., and Renso, C. (2015). On planning sightseeing tours with tripbuilder. *Information Processing & Management*, 51(2):1–15.
- Chen, L., Zhang, L., Cao, S., Wu, Z., and Cao, J. (2020). Personalized itinerary recommendation: Deep and collaborative learning with textual information. *Expert Systems with Applications*, 144:113070.
- da Silva Felix, L. G. et al. (2022). Planet caravan: full trip planner.
- Feo, T. A. and Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6:109–133.
- Friggstad, Z., Gollapudi, S., Kollias, K., Sarlos, T., Swamy, C., and Tomkins, A. (2018). Orienteering algorithms for generating travel itineraries. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 180–188.
- Gavalas, D., Kasapakis, V., Konstantopoulos, C., Pantziou, G., Vathis, N., and Zaroliagis, C. (2015). The ecompass multimodal tourist tour planner. *Expert systems with Applications*, 42(21):7303–7316.
- Gavalas, D., Kasapakis, V., Konstantopoulos, C., Pantziou, G., and Vathis, o. (2017). Scenic route planning for tourists. *Personal and Ubiquitous Computing*, 21(1):137–155.
- Gavalas, D., Konstantopoulos, C., Mastakas, K., and Pantziou, G. (2014). A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3):291–328.
- Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332.
- Headspin (2023). Why measuring and optimizing response time is critical for applications success. <https://shorturl.at/aTX56>. Accessed: 2023-10-17.
- Kotiloglu, S., Lappas, T., Pelechrinis, K., and Repoussis, P. (2017). Personalized multi-period tour recommendations. *Tourism Management*, 62:76–88.
- Leite, A. H., Benevenuto, F., and Moro, M. M. (2013). Triptag: Ferramenta de planejamento de viagens baseada em experiências de usuários de redes sociais. *28o Simpósio Brasileiro de Banco de Dados, Recife*, page 37.
- Liao, Z. and Zheng, W. (2018). Using a heuristic algorithm to design a personalized day tour route in a time-dependent stochastic environment. *Tourism Management*, 68:284–300.
- Lu, E. H.-C., Chen, C.-Y., and Tseng, V. S. (2012). Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 209–218.
- Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.
- Sarkar, J. L. and Majumder, A. (2021). A new point-of-interest approach based on multi-itinerary recommendation engine. *Expert Systems with Applications*, 181:115026.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Vansteenwegen, P., Souffriau, W., Berghe, G. V., and Van Oudheusden, D. (2011). The city trip planner: an expert system for tourists. *Expert Systems with Applications*, 38(6):6540–6546.
- Vansteenwegen, P. and Van Oudheusden, D. (2007). The mobile tourist guide: an or opportunity. *OR insight*, 20:21–27.

- Wibowo, B. S. and Handayani, M. (2018). A genetic algorithm for generating travel itinerary recommendation with restaurant selection. In *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 427–431. IEEE.
- Yochum, P., Chang, L., Gu, T., and Zhu, M. (2020). An adaptive genetic algorithm for personalized itinerary planning. *IEEE Access*, 8:88147–88157.
- Yoo, K.-H., Sigala, M., and Gretzel, U. (2016). Exploring tripadvisor. *Open tourism: open innovation, crowdsourcing and co-creation challenging the tourism industry*, pages 239–255.
- Zheng, W. and Liao, Z. (2019). Using a heuristic approach to design personalized tour routes for heterogeneous tourist groups. *Tourism Management*, 72:313–325.