

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística



Pedro José Vieira de Souza

**HEURÍSTICA DE BUSCA LOCAL PARA ALTERAÇÃO,
ADIÇÃO E REMOÇÃO DE ESTRADAS EM PROJETOS DE
MALHA VIÁRIA TERRESTRE**

Florianópolis

2021

Pedro José Vieira de Souza

**HEURÍSTICA DE BUSCA LOCAL PARA
ALTERAÇÃO, ADIÇÃO E REMOÇÃO DE
ESTRADAS EM PROJETOS DE MALHA VIÁRIA
TERRESTRE**

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como parte dos requisitos necessários para a obtenção do Título de Cientista da Computação.

Orientador: Prof. Dr. Rafael de Santiago

Universidade Federal de Santa Catarina
Departamento de Informática e Estatística

Florianópolis
2021

Pedro José Vieira de Souza

HEURÍSTICA DE BUSCA LOCAL PARA ALTERAÇÃO, ADIÇÃO E REMOÇÃO DE ESTRADAS EM PROJETOS DE MALHA VIÁRIA TERRESTRE

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Cientista da Computação.

Comissão Examinadora

Prof. Dr. Rafael de Santiago
Universidade Federal de Santa Catarina
Orientador

Elder Rizzon Santos
Universidade Federal de Santa Catarina

Pedro Belin Castelucci
Universidade Federal de Santa Catarina

Florianópolis, 5 de agosto de 2022

Agradecimientos

Resumo

De acordo com (Sindicato Nacional da Indústria de Componentes para Veículos Automotores - Sindipeças, 2021), a frota de veículos aumenta anualmente no Brasil, implicando em cada vez mais congestionamento nas grandes cidades. De acordo com (INRIX, 2020) isto tem um impacto financeiro alto, chegando a mais de \$1.000 por motorista estadunidense. Nesse contexto, este trabalho busca otimizar as malhas viárias urbanas para a redução de congestionamento e a diminuição dos tempos de comutação dos veículos utilizando uma heurística de busca local. Para a realização do mesmo, foi feita uma revisão da literatura recente sobre problemas de design de redes, assim como sobre diferentes implementações de busca local. Foi então proposto o modelo a ser utilizado e realizados testes para trazer um melhor entendimento de como aplicar o modelo sobre malhas existentes.

Palavras-Chave: Network Design Problems; Busca Local; Grafos;

Abstract

According to (Sindicato Nacional da Indústria de Componentes para Veículos Automotores - Sindipeças, 2021), the amount of vehicles in Brazil grows every year, implying in heavier traffic, especially in urban areas. This increase in traffic has a high financial impact, as seen in (INRIX, 2020), costing the average US driver over \$1.000 a year. With this in mind, this work tries to optimize urban road networks aiming to reduce traffic and the average vehicle commute time by using a local search heuristic. To reach this goal, a review of recent literature on RNDPs was made, as well as a review on local search implementations. Then a new model was proposed and applied over a few real urban networks to better visualize how this model can be applied.

Keywords: Network Design Problems; Local Search; Graph Theory;

Lista de figuras

Figura 1 – Busca local presa em um mínimo local encontrado em Ritt (2020) . . .	18
Figura 2 – Exemplo com mais nodos e 10% de penetração - Encontrado em Salman e Alaswad (2018)	21
Figura 3 – Exemplo com mais nodos e 90% de penetração - Encontrado em Salman e Alaswad (2018)	21
Figura 4 – Densidades padrão	31
Figura 5 – Densidades otimizadas	31
Figura 6 – Mapa de chapecó 450m, com 1000m de pista disponíveis para otimização	31
Figura 7 – Mapa de Chapecó com 450m de raio	35
Figura 8 – Mapa de Chapecó com 550m de raio	36
Figura 9 – Mapa da região da UFSC com 450m de raio	37
Figura 10 – Mapa da região da UFSC com 550m de raio	38
Figura 11 – mapa de Nashville com 450m de raio	39
Figura 12 – mapa de Nashville com 550m de raio	40
Figura 13 – Mapa da UFSC otimizado com R\$ 10.000	41
Figura 14 – Mapa da UFSC otimizado com R\$ 50.000	42
Figura 15 – Mapa da UFSC otimizado com R\$ 150.000	43
Figura 16 – Exemplo com mais nodos e 10% de penetração - Encontrado em Salman e Alaswad (2018)	49
Figura 17 – Exemplo com mais nodos e 90% de penetração - Encontrado em Salman e Alaswad (2018)	50

Lista de tabelas

Tabela 1 – UFSC: 450m (48 nodos)	26
Tabela 2 – UFSC: 550m (84 nodos)	26
Tabela 3 – Chapecó: 450m (119 nodos)	26
Tabela 4 – Chapecó: 550m (171 nodos)	26
Tabela 5 – Highway level of service(LOS) and congestion conditions	29
Tabela 6 – Tabela encontrada em Salman e Alaswad (2018)	29
Tabela 7 – Chapecó - 450m (119 nodos, 257 arcos)	30
Tabela 8 – Chapecó - 600m (219 nodos, 482 arcos)	30
Tabela 9 – Nashville - 450m (36 nodos, 89 arcos)	30
Tabela 10 – Nashvile - 600m (75 nodos, 184 arocs)	30
Tabela 11 – UFSC - 450m (48 nodos, 89 arcos)	30
Tabela 12 – UFSC - 600m (115 nodos, 229 arcos)	30
Tabela 13 – UFSC: 450m (48 nodos)	55
Tabela 14 – UFSC: 550m (84 nodos)	55
Tabela 15 – Chapecó: 450m (119 nodos)	55
Tabela 16 – Chapecó: 550m (171 nodos)	55
Tabela 17 – Highway level of service(LOS) and congestion conditions	58
Tabela 18 – Tabela encontrada em Salman e Alaswad (2018)	58
Tabela 19 – Chapecó - 450m (119 nodos, 257 arcos)	59
Tabela 20 – Chapecó - 600m (219 nodos, 482 arcos)	59
Tabela 21 – Nashville - 450m (36 nodos, 89 arcos)	59
Tabela 22 – Nashvile - 600m (75 nodos, 184 arocs)	59

Lista de Siglas e Abreviaturas

IBGE	<i>Instituto Brasileiro de Geografia e Estatística</i>
NDP	<i>Network Design Problems</i>
UFSC	<i>Universidade Federal de Santa Catarina</i>
RNDP	<i>Road Network Design Problem</i>
OSM	<i>Open Street Map</i>

Sumário

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos específicos	14
1.1.3	Metodologia	14
1.1.4	Estrutura do Texto	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Network Design Problems	16
2.1.1	NDP	16
2.1.2	RNDP	16
2.2	Busca Local	17
2.2.1	Vizinhança	17
2.2.2	Ótimos locais e globais	17
2.2.3	Buscas locais não-monótonas	17
2.2.4	Buscas locais monótonas	19
2.3	Cadeias de Markov	19
2.3.1	Cadeias de Markov e o RNDP	19
3	TRABALHOS RELACIONADOS	20
3.1	A bi-level model to optimize road networks for a mixture of manual and automated driving: An evolutionary local search algorithm	20
3.2	Optimisation of transportation service network using k-node large neighbourhood search	22
3.3	Discussão dos trabalhos	22
4	ESPECIFICAÇÃO E MÉTODOS	23
4.1	Busca local em RNDP	23
4.2	Experimentos	24
4.2.1	Otimização de tempo de viagem	25
4.2.2	Otimização de densidade de veículos	28
5	CONSIDERAÇÕES FINAIS	32
5.1	Trabalhos Futuros	32

	REFERÊNCIAS BIBLIOGRÁFICAS	33
6	APÊNDICES	35
6.1	Introdução	47
6.1.1	Metodologia	48
6.2	Trabalhos Relacionados	48
6.2.1	A bi-level model to optimize road networks for a mixture of manual and automated driving: An evolutionary local search algorithm	49
6.2.2	Optimisation of transportation service network using k-node large neighbourhood search	51
6.2.3	Discussão dos trabalhos	51
6.3	Especificação e Métodos	51
6.3.1	Busca local em RNDP	52
6.3.2	Experimentos	52
6.3.3	Otimização de tempo de viagem	54
6.3.4	Otimização de densidade de veículos	57

1 Introdução

A frota circulante de veículos automotores aumenta ano após ano no Brasil, aumentando em 2,5% de 2018 para 2019 e em 0,7% de 2019 para 2020 (Sindicato Nacional da Indústria de Componentes para Veículos Automotores - Sindipeças, 2021). Com este aumento constante da frota lidar com o congestionamento das vias se torna um desafio para as cidades brasileiras e deve-se portanto ser desenvolvida alguma estratégia concreta que consiga otimizar as vias e auxiliar as gestões municipais a lidar com o trânsito.

De acordo com um estudo da INRIX realizado em 2020, o congestionamento nos Estados Unidos custou um média 1.347 dólares por motorista em 2019 (INRIX, 2020), com esta observação, pode-se perceber que o engarrafamento é indesejável não apenas pelo desconforto do motorista, mas por ser economicamente negativo para os cidadãos. Dentro do campo de ciências da computação podemos abordar este problema de diversas maneiras, neste trabalho iremos tratá-lo como um *Network Design Problem* (NDP).

Solucionar um NDP significa encontrar modificações para serem feitas sobre os arcos, buscando algum objetivo bem definido como por exemplo otimizar o fluxo total da rede ou então o fluxo de uma sub-rede.

Um exemplo de otimização viária terrestre utilizando uma abordagem de NDPs pode ser visto na literatura em Salman-Alaswad(2018)(SALMAN; ALASWAD, 2018), onde é utilizada uma meta-heurística genética para buscar uma solução com um fluxo médio da rede otimizado.

Neste trabalho é proposta uma abordagem diferente, utilizando uma heurística de busca local de vizinhança. Algoritmos de busca local são algoritmos que buscam novas soluções para um problema realizando pequenas modificações (chamadas de movimentos) a soluções anteriores. Quando se tem em mente malhas viárias reais devemos sempre buscar realizar o mínimo de movimentos para chegar numa otimização, já que cada mudança na rede acarreta em um aumento do orçamento necessário para concretizar o projeto e, por consequência, grandes mudanças trazem grandes custos, podendo inviabilizar a execução do projeto.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo principal do projeto é o desenvolvimento de uma heurística de busca local capaz de realizar modificações numa dada rede, a fim de otimizar o seu fluxo.

1.1.2 Objetivos específicos

1. implementar em código uma abordagem de busca local para o problema de RNDP através de inversão de pistas
2. expandir a implementação de forma a incluir as novas funções de adição e remoção de pistas.
3. realizar *benchmarks* para verificar a eficiência do novo modelo.
4. disponibilizar código-fonte e aplicação de maneira acessível ao público interessado.

1.1.3 Metodologia

O presente trabalho se trata de uma pesquisa exploratória, adaptando uma heurística de busca local a um novo problema.

Os seguintes passos foram feitos para a realização deste trabalho:

Um levantamento teórico amplo, visando trazer credibilidade a pesquisa realizada. Nesta fase, trabalhos prévios com definições das heurísticas utilizadas serão apresentados para que as definições fiquem claras e consistentes com a literatura. A busca de artigos será feita em portais e repositórios como o Google Scholar, e o SCOPUS.

Após o estudo dos trabalhos anteriores, será feita a definição do novo modelo, apresentando o embasamento por trás das decisões feitas, assim como a especificação da metodologia dos testes. Devem ser definidas as métricas que serão utilizadas para permitir a comparação com os trabalhos prévios, e também os casos de teste que passarão pela nova abordagem.

Em seguida será desenvolvida uma aplicação para implementar a nova abordagem, assim como os casos de testes. Ambas estas aplicações devem ser disponibilizadas junto com este trabalho, garantindo assim a transparência quanto a legitimidade dos testes, assim como a possibilidade de replicação dos mesmos.

O último passo, é a análise dos resultados, buscando conclusões sobre a eficiência e eficácia da nova abordagem. A partir das quais será possível observar em quais situações esse método pode ser utilizado.

1.1.4 Estrutura do Texto

Este documento está estruturado em cinco capítulos, sendo o primeiro o capítulo de introdução acima.

No Capítulo 2 serão apresentados os conceitos fundamentais necessários para o entendimento do trabalho. Estes conceitos serão fundamentados em trabalhos prévios encontrados na literatura.

O Capítulo 3 apresentará trabalhos correlatos, mostrando outros trabalhos similares e relevantes que foram desenvolvidos, e o que se espera deste trabalho quando comparado com os trabalhos relacionados.

No Capítulo 4 serão especificados os novos modelos propostos, também serão apresentadas as redes que servirão de casos de testes.

No Capítulo 5 teremos as conclusões atuais do trabalho, comentando brevemente sobre o que já foi feito e o que ainda deve ser feito para a finalização do mesmo. Também deve ser apresentado um cronograma, demonstrando o plano de desenvolvimento do trabalho.

2 Fundamentação Teórica

Neste capítulo serão abordados assuntos relacionados a conceitos básicos utilizados durante este trabalho. Serão apresentadas definições de conceitos estabelecidos na literatura, para um embasamento do trabalho.

2.1 Network Design Problems

Segundo Leblanc (1975) e Poorzahedy e Turnquist (1982), um Network Design Problem (Problema de Design de Rede), ou de forma abreviada um NDP genérico pode ser definido como um grafo com as seguintes características

2.1.1 NDP

Com isso podemos definir um NDP da seguinte forma:

- $G = (V, A)$, uma rede;
- $A = A_1 \cup A_2$, um conjunto de arcos;
- A_1 , um conjunto de arcos que não podem sofrer mudanças;
- A_2 , um conjunto de arcos que podem sofrer mudanças e;
- V , um conjunto de nodos.

Além de possuir essas características, um NDP requer um objetivo para ser alcançado, por exemplo encontrar a configuração dos arcos que atinge o maior fluxo possível para esta rede. A partir desse objetivo e a rede associada temos um NDP.

2.1.2 RNDP

Um Road Network Design Problem (ou RNDP) possui algumas características mais específicas quando comparado com um NDP genérico. Em primeiro lugar para estes problemas devemos sempre considerar que a quantidade de mudanças realizadas a nossa rede inicial deve ser mínima, visando reduzir custos. Também deve-se pensar, de acordo com Farahani et al. (2013) em estratégias de longo prazo, já que mudanças de malha viária não devem ser feitas com alta frequência. Além disso devemos considerar a reação da população as mudanças, já que nem todas as otimizações criadas serão utilizadas (e.g. uma rua que é vantajosa para o transporte, mas atravessa uma região perigosa de uma cidade). Por último devemos lembrar que dada a natureza do problema a criação ou modificação de certos arcos é impossível devido a geografia local.

2.2 Busca Local

De acordo com Ritt (2020) e Pirlot (1996), uma busca local é uma abordagem para problemas de otimização onde são realizadas mudanças a uma solução inicial, chamamos essas pequenas modificações de movimentos. Estes movimentos fazem com que a solução evolua para que seja obtida no fim uma solução com o valor de uma função de otimização melhorado. Esta função de otimização demonstra a adequação da solução ao problema, e varia de acordo com o problema que está sendo abordado. Em certos casos busca-se um valor cada vez menor (e.g. uma função de otimização que calcula o custo de uma dada solução) e em outros casos o objetivo é maximizar este valor (e.g. uma função de otimização que calcula o fluxo máximo de um grafo). Um algoritmo genérico para buscas locais pode ser encontrado em 1, para um melhor entendimento devem ser observadas algumas definições.

2.2.1 Vizinhança

Uma vizinhança se trata de de um conjunto de soluções separadas por um movimento, ou seja, são soluções similares, onde podemos observar as diferenças de otimização destes movimentos e descobrir movimentos que trazem maior vantagem para uma solução. Quando se trata de uma busca local sobre um problema de grafos, como é o caso dos NDPs, cada vizinho dentro da vizinhança é um grafo que soluciona o problema proposto com uma eficiência melhor ou pior, porém nunca deve ser permitido que um dado movimento nos leve a um grafo que não solucione o problema.

2.2.2 Ótimos locais e globais

O uso de vizinhanças nos possibilita encontrar ótimos locais, já que para toda solução \mathbf{s} temos um conjunto de vizinhos N contendo os vizinhos \mathbf{s}' com os quais podemos comparar a eficácia da solução (e.g. capacidade de fluxo da rede, média de fluxo em um dado arco). Conhecendo estes ótimos locais, podemos fazer uma comparação mais ampla onde comparamos os ótimos locais de cada vizinhança para encontrar um ótimo global. Pode-se observar um exemplo de mínimo local em 1 onde para uma função de otimização $F(\mathbf{x})$ sobre um espaço de busca X , podemos observar que x_n é um ótimo local, e dependendo do tipo de busca utilizada será a melhor otimização encontrada. Diferentes soluções para atravessar esses máximos locais existem, porém dependendo da complexidade do problema analisado a execução destas soluções pode ser computacionalmente inviável.

2.2.3 Buscas locais não-monótonas

Ainda em Ritt (2020) vemos definições de monotonia para buscas locais. As buscas locais não-monótonas nos permitem piorar a solução atual. Embora no primeiro momento

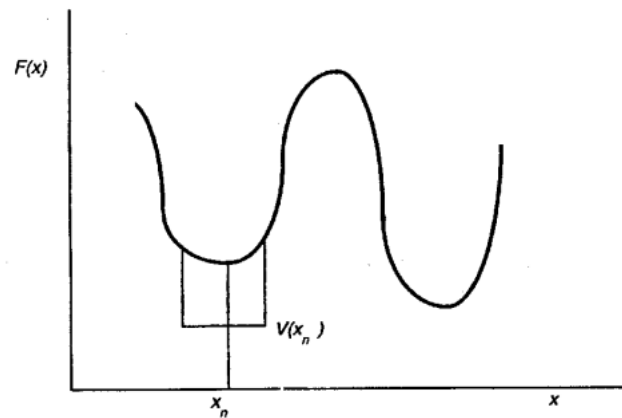


Figura 1 – Busca local presa em um mínimo local encontrado em Ritt (2020)

Algorithm 1: Algoritmo Genérico de Busca Local

Data: Solução inicial s , vizinhança N , uma distribuição P , uma função de otimização α

Result: Saída: Uma solução s^* tal que $\alpha(s^*) \leq \alpha(s)$

```

1  $s^* \leftarrow s$ ;
2 while  $\neg(\text{critério de parada})$  do
3   | seleciona  $s' \in N(s)$  de acordo com  $P(s)$ ;
4   |  $s \leftarrow s'$ ;
5   | if  $\alpha(s) < \alpha(s^*)$  then
6   |   |  $s^* \leftarrow s$ ;
7   | end
8 end
9 return  $s^*$ 

```

não pareça uma vantagem, essa característica nos permite potencialmente encontrar soluções com um maior grau de otimização, já que podem ser necessários diversos movimentos negativos para alcançar os requisitos para um movimento com um impacto positivo em uma escala maior. Isto pode ser visualizado na figura 1, onde uma abordagem não-monótona poderia atravessar os valores que trazem um aumento em $F(x)$ e encontrar o vale seguinte, onde é encontrado um mínimo local melhor. Porém, para serem utilizadas heurísticas não-monótonas é necessário definir um critério de parada, já que existe possibilidade de melhorar o mínimo local encontrado. Dentro do contexto de NDPs podemos definir diversos critérios de parada dependendo do problema abordado, seguem alguns exemplos de critérios de parada em NDP:

- Fluxo médio da rede $\geq x$
- Fluxo de uma sub-rede $R \geq x$
- Capacidade somada das arestas de uma sub-rede $R \geq x$

2.2.4 Buscas locais monótonas

Já as buscas monótonas são aquelas que não permitem movimentos que piorem a solução, logo algoritmos que tem essa característica sempre acabam em ótimos locais, porém não há garantia que serão ótimos globais, já que para alcançar o ótimo global pode ser necessário passar por movimentos que piorariam a solução atual. No caso de uma algoritmo que implemente esta estratégia, a melhor solução encontrada para o problema representado na figura 1 seria x_n , caso a solução inicial estivesse entre o máximo local à sua esquerda e o máximo local à sua direita.0

Para garantir este comportamento consideramos o conjunto $B \in N$, onde estão apenas os vizinhos melhores da nossa solução. Além disso, a função de distribuição P assume valor 0 para qualquer solução s' onde $s' \notin B$.

Por conta desse comportamento de apenas descer dentro da vizinhança local, as buscas locais monótonas podem ser muito úteis quando o problema abordado possui restrições de modificação, como minimização do custo ou recursos limitados.

2.3 Cadeias de Markov

Uma cadeia de Markov é um processo estocástico com estados discretos, onde o próximo estado na cadeia depende apenas do estado presente, ou seja, a probabilidade de uma variável possuir um valor em um dado estado depende unicamente do seu estado anterior, e não de todo um caminho percorrido (SALMAN; ALASWAD, 2018). Uma maneira comum de se definir cadeias de Markov é através de grafos direcionados, onde os nodos representam os diferentes estados possíveis e os arcos representam a probabilidade do estado atual transitar para um dado estado futuro. Porém, uma cadeia de Markov também pode ser facilmente definida através de uma matriz de transições, onde os estados não imediatamente alcançáveis possuem uma probabilidade $p = 0$.

2.3.1 Cadeias de Markov e o RNDP

Como visto em Salman e Alaswad (2018) cadeias de Markov podem facilitar a validação de uma heurística utilizada sobre o RNDP, já que elas nos permitem fazer uma estimativa do comportamento do transito em uma malha viária sem a necessidade de criar e executar simulações computacionalmente caras. Além disso é possível estimar uma quantidade específica de passos discretos no tempo, e observar os impactos que estes causam na malha que está sendo observada.

3 Trabalhos Relacionados

Neste capítulo, iremos selecionar trabalhos relacionados a este, tendo como objetivo trazer uma visão do que foi desenvolvido nesta área nos últimos anos, e qual contribuição este trabalho pode trazer.

Para encontrar os trabalhos prévios com um bom nível de correlação, utilizaremos duas palavras chave no portal SCOPUS: "Network Design Problem" e "Local Search". Ao realizar esta busca apenas 104 documentos são encontrados, este número razoavelmente baixo nos mostra que nossas palavras chave foram suficientemente limitantes.

O critério de seleção dos trabalhos será a relevância de acordo com o SCOPUS, passando por uma filtragem do autor, buscando garantir que os trabalhos sejam realmente relacionados com a pesquisa aqui desenvolvida.

3.1 A bi-level model to optimize road networks for a mixture of manual and automated driving: An evolutionary local search algorithm

O trabalho desenvolvido por Madadi et al. (2020), aborda pontos importantes sobre a integração do transporte autônomo com as modalidades de veículos tradicionais. São definidos conceitos de valor do tempo de viagem (VoTT) para veículos autônomos e tradicionais, sendo o VoTT de transporte por veículos autônomos mais baixo, já que o tempo do transporte pode ser utilizado para realizar outras tarefas e é portanto menos limitante que o transporte tradicional. Os autores citam trabalhos mostrando a improbabilidade da existência de veículos autônomos funcionais em todo tipo de estrada no futuro próximo e definem como objetivo determinar uma sub-rede para veículos autônomos dentro de uma dada rede de forma que apenas esta sub-rede precise estar dentro do padrão necessário para o funcionamento dos veículos autônomos. Esta limitação permite uma redução no custo de atualização das vias. No trabalho são comparadas três abordagens, sendo duas delas algoritmos genéticos (GA e MGA) e uma por busca local evolutiva (ELS). Também é utilizada uma taxa de penetração variável, que indica a porcentagem da rede que deveria ser adaptada para capacitar o trânsito de veículos autônomos.

Podemos ver nas figuras (16 e 17) a diferença entre os algoritmos, em verde temos as vias a serem adaptadas buscando minimizar o custo de viagem na rede fornecida, enquanto tenta-se também minimizar o custo de adaptação. O tempo computacional dos algoritmos avaliados é consideravelmente alto, com a execução mais rápida sendo a do ELS-10% em 76 minutos, e a mais lenta sendo a MGA-50% em 968 minutos. Vale notar que existe no

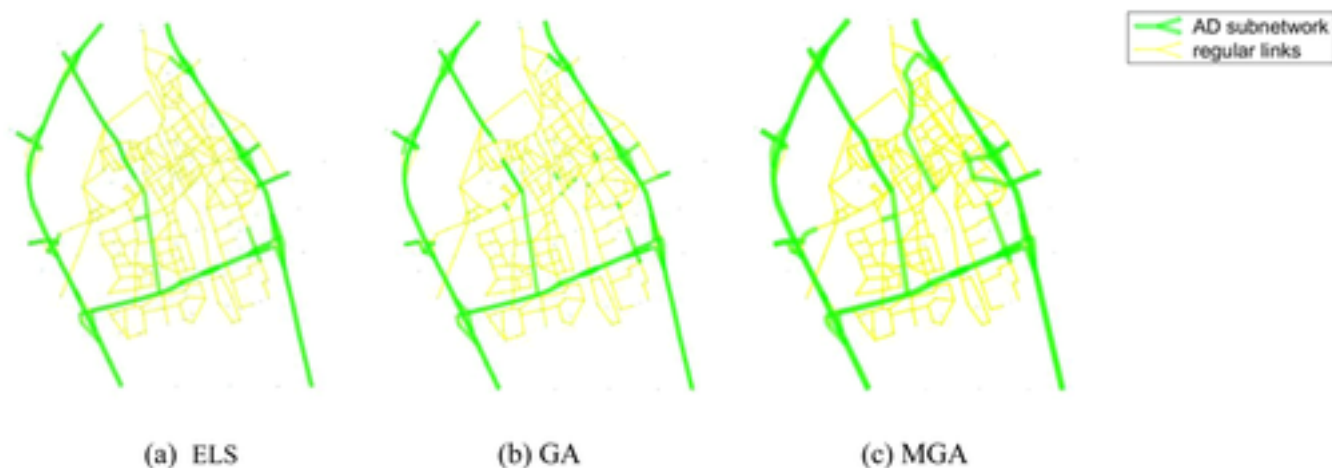


Figura 2 – Exemplo com mais nodos e 10% de penetração - Encontrado em Salman e Alaswad (2018)

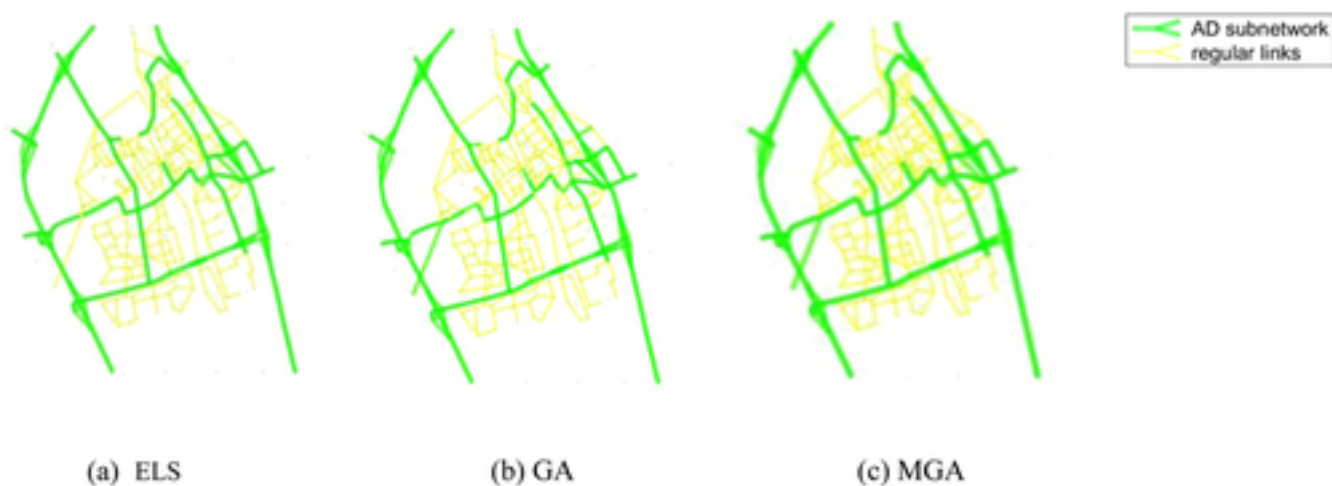


Figura 3 – Exemplo com mais nodos e 90% de penetração - Encontrado em Salman e Alaswad (2018)

trabalho uma métrica de "Função objetivo" que toma em tanto o custo total de viagem e o custo total de adaptação onde todas as abordagens tiveram resultados similares. A razão pela qual esta métrica é notável dentro do trabalho que está sendo desenvolvido é que ela também pode ser utilizada aqui. Mesmo com os trabalhos abordando problemas significativamente diferentes esta métrica serve como um bom ajuste para o custo-benefício de uma solução, tendo em vista o retorno a longo prazo que um custo de viagem reduzido traz.

3.2 Optimisation of transportation service network using k-node large neighbourhood search

Este trabalho levanta pontos interessantes sobre a pesquisa de Service Network Design Problems, que podem ser expandidos para NDPs de forma geral. De acordo com Bai et al. (2018) mesmo existindo uma grande quantidade de pesquisa sobre NDPs e otimização de rotas dentro das redes, a maior parte dos trabalhos se limita a encontrar uma nova heurística para abordar uma mesma modelagem de vizinhança independente da abordagem. Os autores propõem então uma modificação vista como mais baixo nível, buscando otimizar a própria função de vizinhança. Embora o problema abordado de SNDPs seja significativamente diferente dos problemas RNDP abordados neste trabalho, é necessário observar a possível utilidade de fazer mudanças nas vizinhanças dentro de RNDPs também, pois potencialmente existem otimizações a serem feitas neste campo que poderiam afetar diversos problemas, não importando qual for o problema abordado ou implementação específica. Os autores por fim utilizam uma implementação híbrida, utilizando diferentes passos para encontrar a melhor solução, onde um dos passos é a otimização de vizinhanças para SNDPs proposta no trabalho, demonstrando a possibilidade de utilizar otimizações de vizinhança junto a estratégias meta-heurísticas. Os resultados para os testes com essa abordagem híbrida foram melhores que ambas as estratégias separadas, devido ao alto custo computacional do método apresentado, e da acurácia mais baixa da busca Tabu demonstrada no trabalho.

3.3 Discussão dos trabalhos

A partir dos trabalhos apresentados nas seções anteriores, percebe-se um padrão quando se trata de buscas locais que é o uso de mais de uma estratégia, organizando-as em etapas para melhor aproveitar as vantagens trazidas por cada uma. Os artigos também levantam pontos importantes, como diferentes funções de objetivo que não o fluxo médio da rede (como o método baseado em densidade de veículos em Salman e Alaswad (2018)) sendo então interessante para trabalhos que tragam novas abordagens sobre RNDPS observarem se os métodos que estão desenvolvendo podem ser generalizados para qualquer variável de objetivo e até mesmo se outras variáveis não são mais compatíveis com o modelo sendo desenvolvido.

4 Especificação e Métodos

Neste capítulo, serão apresentados o modelo desenvolvido neste trabalho, bem como os métodos utilizados para realizar os testes.

4.1 Busca local em RNDP

Observando as definições do capítulo 2, consegue-se definir um modelo de busca local para abordar RNDPs. A primeira e mais importante definição é quanto a monotonia do modelo que será utilizado. Ao analisar as vantagens e desvantagens dos modelos, estipula-se que heurísticas monótonas se encaixem melhor ao escopo do problema. Isso se deve a minimização de mudanças que estas trazem, já que os RNDPs não podem sofrer modificações muito drásticas em suas soluções.

Um algoritmo genérico de busca local monótona pode ser encontrado em Ritt (2020)

Algorithm 2: Algoritmo Genérico de Busca Local Monótona
<p>Data: Solução inicial s, vizinhança N, uma distribuição P, uma função de otimização α</p> <p>Result: Saída: Uma solução com valor de otimização no máximo $\alpha(s)$</p> <pre> 1 $s^* \leftarrow s$; 2 while $\neg(P(s) = 1)$ do 3 seleciona $s' \in N(s)$ de acordo com $P(s)$; 4 $s \leftarrow s'$; 5 if $\alpha(s) < \alpha(s^*)$ then 6 $s^* \leftarrow s$; 7 end 8 end 9 return s^* </pre>

Este algoritmo pode ser adequado a uma RNPD utilizando o seguinte mapeamento para as variáveis:

- $s \leftarrow$ solução inicial do caso de teste;
- $N \leftarrow$ conjunto de soluções a um movimento de distância, para todos os possíveis movimentos;
- $P \leftarrow$ uma função de distribuição a ser desenvolvida;
- $\alpha \leftarrow$ o tempo médio das viagens dentro da rede

4.2 Experimentos

Para os experimentos, foi desenvolvida uma aplicação em *python* implementando a heurística a ser testada. À priori foi construída uma abordagem com uma heurística de busca local simples, como o visto em 1, tendo como critério de parada o gasto máximo permitido para mudanças no projeto de malhas, também parando a execução caso após uma iteração não seja encontrada nenhuma melhoria.

Algorithm 3: Algoritmo para calculo de média de tempo

```

Data: Grafo direcionado  $G$ , valor minimo  $V$ 
Result: Saída: média de tempo de viagem  $M$ 
1  $nodes \leftarrow G.nodes$ ;
2  $total\_time \leftarrow 0$ ;
3  $total\_routes \leftarrow 0$ ;
4 for  $origin \in nodes$  do
5   for  $dest \in nodes$  do
6     if  $origin \neq dest$  then
7        $route\_time \leftarrow shortest\_path(origin, dest)$ ;
8       if  $route\_time \geq V$  then
9          $total\_time \leftarrow total\_time + route\_time$ ;
10         $total\_routes \leftarrow total\_routes + 1$ ;
11      end
12    end
13  end
14 end
15  $M \leftarrow total\_time/total\_routes$ ;
16 return  $M$ ;

```

Foram propostos casos de testes com diferentes tamanhos, orçamentos e regiões. Para todos os casos propostos, foi gerado um grafo do mapa a partir de dados coletados do OpenStreetMap (OSM). Sobre esse grafo é analisado o tempo médio dos trajetos, utilizando o algoritmo 16. Tendo este valor salvo, podemos fazer uma busca local exaustiva para todas as possíveis mudanças, no escopo deste trabalho foram inclusas a inversão de pista, a remoção de uma pista e a adição de uma pista. Para cada mudança feita este tempo é recalculado e, caso seja melhor que o mapa inicial e o candidato atual de mudança, é eleito como a próxima alteração a ser feita ao mapa. Ao final de um ciclo têm-se a melhor mudança viável para o mapa.

4.2.1 Otimização de tempo de viagem

Para a primeira bateria de testes, definimos três localidades (UFSC, Chapecó e Nashville), bem como dois possíveis raios (450m e 550m) e três níveis de orçamento (10, 50 e 150 mil reais). O custo de modificação de cada km de via foi utilizada uma tabela de custos do MT/DNER (2007). A partir deste custo e do comprimento da via a ser alterada é determinado o preço da obra para uma dada modificação.

Os resultados destes testes podem ser observados nas Tabelas 13, 14, 15, 16. Pode-se observar a média inicial do tempo das viagens, a média otimizada, o número de modificações feitas e o tempo de execução. Rapidamente percebe-se que essa otimização não parece muito grande, sendo em média uma redução de apenas 0.05% no tempo médio de viagem na malha. Isso é natural, visto que qualquer mudança feita em um único arco não vai afetar a maior parte das viagens. Também percebe-se um alto tempo de execução, sendo em média 52.8s por passo dado nas malhas de pequeno porte (450m), e de 883s (14m43s) por passo nas malhas de porte maior (600m). Este crescimento rápido com o aumento dos nodos é também natural das buscas locais exaustivas, porém a heurística se mostra inviável para malhas de grandes portes.

As Figuras 7, 8, 9, 10, 11 12 mostram os mapas antes de serem otimizados (com um *colormap* somente ilustrativo).

As Figuras 13, 14, e 15 nos mostram os mapas de otimização da UFSC: 550m em diferentes níveis de orçamento, com as pistas modificadas em vermelho, e as não modificadas em azul. Especificamente na figura 15 podemos observar uma pista onde uma de suas faixas foi modificada e a outra não, gerando uma estrada em roxo. Podemos observar que são mudanças pequenas em relação ao tamanho total da malha.

Tabela 1 – UFSC: 450m (48 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	98.20553420445817	98.10107692307687	R\$ 3.913,15	2	127,6629s
R\$ 50.000	98.10107692307687	98.10107692307706	R\$ 45.315,20	2	137,3100s
R\$ 150.000	98.10107692307687	98.10107692307680	R\$ 146,452.20	1	92,3190s

Tabela 2 – UFSC: 550m (84 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	116.62899875363496	116.54137931034467	R\$ 1.417,15	2	1379,3913s
R\$ 50.000	116.62899875363496	116.54137931034452	R\$ 33,375.20	3	1900,8265s
R\$ 150.000	116.62899875363496	116.54137931034467	R\$ 37,624.60	3	1850,0693s

Tabela 3 – Chapecó: 450m (119 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	75.0378384426048	75.0378384426048	R\$ 10.000	0	30,3279
R\$ 50.000	75.0378384426048	75.0378384426048	R\$ 50.000	0	31,6178
R\$ 150.000	75.0378384426048	75.0378384426048	R\$ 150.000	0	29,7678

Tabela 4 – Chapecó: 550m (171 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	81.17417908178834	81.15966246008828	R\$ 8,611.75	1	1654,0852s
R\$ 50.000	81.17417908178834	81.15966246008819	R\$ 39,802.70	1	1142,8250s
R\$ 150.000	81.17417908178834	81.15966246008806	R\$ 98,554.09	2	1122,2729s

Pode-se observar na diferença dos tempos de execução, por exemplo entre as tabelas 14, e 16, que o custo computacional desta abordagem é bastante alta para mapas com uma quantidade elevada de nodos, este comportamento é esperado da heurística de busca local, porém ele limita o escopo do trabalho para pequenas regiões. Para lidar com essa limitação, o mapa pode ser redesenhado como uma série de regiões pequenas (com menos de 150 nodos por exemplo) interconectadas por um mapa maior que inclua apenas estradas centrais da cidade, como as SCs na grande Florianópolis. Ao reimaginar o mapa desta forma, podem ser feitas múltiplas execuções mais curtas do algoritmo, ao invés de uma longa execução.

O tempo de execução cresce muito rapidamente ao adicionar mais nodos ao grafo, isso fica claro no algoritmo 16, onde a função *shortest_path* implementa a busca de menor rota através do algoritmo de Dijkstra, tendo sozinha uma complexidade de $O(A+V\log V)$, onde A é o número de arcos e V o número de Nodos. Além disso, o algoritmo busca a rota mais curta para todos os pares de nodos, aumentando essa complexidade por V^2 vezes. Finalmente, esse cálculo de média de tempo deve ser executado cada vez que uma mudança é feita nas vias, para verificar se houve melhoria. No caso dos experimentos atuais, são feitas três mudanças por arco em cada ciclo, aumentando então a complexidade em $3A$ vezes trazendo a nossa complexidade final para $O(V^2A^2 + AV^3\log V)$. Sabendo da complexidade do algoritmo, pode-se afirmar que as execuções de diferentes regiões

menores, e interconectadas seriam sim mais rápidas do que uma execução de um único grande mapa, devido a natureza cúbica do problema.

Também pode-se perceber que o número de mudanças relevantes encontradas pelo método exerce uma influência no tempo de execução. Isso fica bastante claro ao comparar as Tabelas 14 e 15 onde mesmo com mais nodos o mapa da segunda tabela tem um tempo de execução curtíssimo. Isso se deve ao fato de que não existiam mudanças relevantes para o tempo médio de viagem na malha, causando o algoritmo a realizar apenas uma passagem inicial sobre a malha.

4.2.2 Otimização de densidade de veículos

Propõe-se uma segunda próxima bateria de testes, onde a variável a ser otimizada é alterada, ao invés de trabalhar sobre o *traveltime* médio da malha, o foco será a densidade média de veículos da malha (como visto em Salman e Alaswad (2018)), esta decisão de mudança é devido ao fato de que a otimização da média do tempo de viagem é, de forma geral, bastante baixa e por vezes pode não realmente demonstrar uma melhoria na malha de forma geral (e.g. um cenário onde uma melhoria grande suficiente de um único *traveltime* A->B melhora a média porém piora o *traveltime* de diversas outras rotas), já o método de densidade de veículos tem como motivação a dispersão do congestionamento, trazendo uma melhoria mais ampla para a malha viária observada.

Para o cálculo dessas densidades foi utilizado o método também encontrado em Salman e Alaswad (2018) utilizando cadeias de Markov, onde a probabilidade de um veículo depende do comprimento da via onde ele está, bem como das vias para onde ele pode ir, onde uma maior quilometragem aumenta a probabilidade de ser uma via destino. Pode-se pensar nessa definição como a seguinte função de probabilidade

$$\forall e, f \in A$$

$$P(e, f) = \begin{cases} 0 & \text{quando } v(e) \neq u(f) \\ \frac{(length(f)*lanes(f))}{\sum_{i=0}^n P(e,x(i))} & \text{quando } v(e) = u(f) \end{cases}$$

- A o conjunto de arcos do grafo
- x uma função que retorna um arco de acordo com seu índice
- *length* uma função que retorna o comprimento do arco
- *lanes* uma função que retorna o numero de pistas do arco
- u e v retornam os nodos de origem e destino de um arco, respectivamente

Para a definição de classes de ocupação de uma pista (visualizada nas imagens) podemos observar a tabela 18:

Tabela 5 – Highway level of service(LOS) and congestion conditions

LOS	Traffic Conditions	Volume/Capacity Ratio	Maximum Density (vehichle/km/lane)
A	Free	0.35	7
B	Stable	0.54	11
C	Stable	0.77	16
D	High Density	0.93	22
E	Near Capacity	1.00	28
F	Breakdown	Unstable	>28

Tabela 6 – Tabela encontrada em Salman e Alaswad (2018)

A partir desta tabela e do método da cadeia de Markov acima, foram geradas as densidades de veículos para cada arco dos mapas tanto antes quanto depois das modificações serem realizadas. A variável a ser otimizada é a soma dos dez maiores valores dessa lista, assim evita-se que o algoritmo simplesmente alterne a via onde se encontra o congestionamento. Além disso, foi removido o calculo de custos para essa segunda bateria de testes, essa mudança foi motivada tanto para redução de operações em máquina sendo feitas (trazendo mais velocidade para a execução de uma análise) e também tendo em vista de que o custo estimado poderia ser longe da realidade dependendo do local sendo analisado e portanto ter o custo como um critério de parada não auxiliava realmente a solucionar o problema em questão. Foi utilizado então somente quantidades fornecidas de metros que o algoritmo poderia modificar, para os testes foram utilizados 200, 500 e 1000 metros de orçamento.

Este método obteve resultados de maior qualidade, como podemos ver nas Tabelas 19, 20, 21, 22, 23, 24. Obtendo reduções da métrica observada magnitudes acima do que as reduções dos testes realizados na primeira bateria, com uma média de redução de 10.60% nos mapas de 450 metros e de 3.85% nos mapas de 600 metros. Pode-se observar na Tabela 20 um comportamento inesperado onde foi obtido um tempo total de execução de duas horas para um único mapa, mas ao realizar uma análise mais profunda observando o tempo de execução por passo, encontramos uma média de 361s por passo noas mapas de 600m, uma redução de 59% em relação ao método testado na primeira bateria.

Analisando mais a fundo o algoritmo utilizado, é perceptível o motivo de uma redução tão significativa no tempo de execução. O algoritmo utilizado para a segunda bateria de testes não depende do numero de nodos, dependendo apenas do numero de arcos de um dado mapa. Além disso sua complexidade é bastante menor, tendo uma complexidade total de $O(A^3 + A^2)$ ou seja, seu crescimento é cúbico em relação as arestas, enquanto na primeira bateria tinha-se um de grau muito maior ($A^2V^2 + AV^3 \log V$).

Tabela 7 – Chapecó - 450m (119 nodos, 257 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	784.9107000000004	764.9800000000006	2	447,502033s
500	784.9107000000004	731.3361750000001	10	1482,678092s
1000	784.9107000000004	711.4766749999999	11	1543,163062s

Tabela 8 – Chapecó - 600m (219 nodos, 482 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	1447.8596000000014	1427.8835000000013	2	1934,328336s
500	1447.8596000000014	1393.3651000000011	5	3901,393258s
1000	1447.8596000000014	1356.712125000001	13	9660,576185s

Tabela 9 – Nashville - 450m (36 nodos, 89 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	450.4251749999999	428.9938749999999	4	32,995950s
500	450.4251749999999	396.6446499999999	4	30,193408
1000	450.4251749999999	372.6997249999994	5	33,621320

Tabela 10 – Nashville - 600m (75 nodos, 184 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	1012.3540999999999	1010.4519750000001	2	215,784380s
500	1012.3540999999999	1004.4859499999999	3	289,140887s
1000	1012.3540999999999	980.3352499999999	5	421,830091

Tabela 11 – UFSC - 450m (48 nodos, 89 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	322.1485750000001	307.3986250000001	3	28,588393s
500	322.1485750000001	287.534475	3	29,138282s
1000	322.1485750000001	233.7517750000004	3	25,793708s

Tabela 12 – UFSC - 600m (115 nodos, 229 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	795.5606749999998	789.6622999999997	3	484,99126s
500	795.5606749999998	753.9375249999996	3	475,906789s
1000	795.5606749999998	691.0352249999995	3	469,407217s

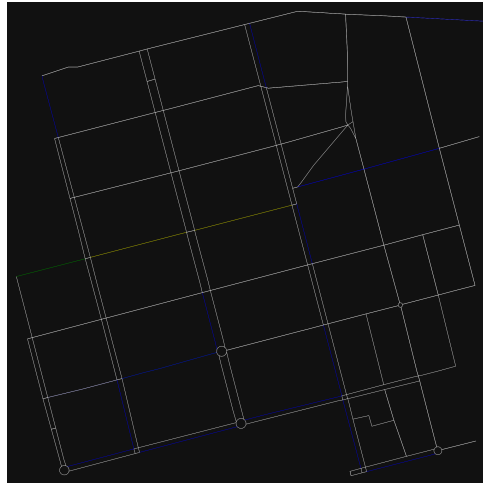


Figura 4 – Densidades padrão

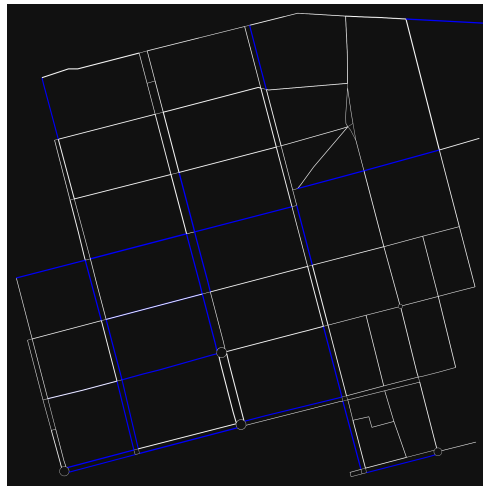


Figura 5 – Densidades otimizadas

Figura 6 – Mapa de chapecó 450m, com 1000m de pista disponíveis para otimização

Podemos ver na figura 20 um exemplo de otimização. Onde as cores do *heatmap* foram mapeadas para as classes definidas em 18. Como podemos ver, as estradas em amarelo (classe D) foram reduzidas para azul (classe B). É necessário ressaltar que certas vias brancas (classe A) também se tornam azuis. Isso se deve a distribuição do congestionamento que está sendo diluído para vias alternativas e é um bom sinal de que não há veículos sendo apagados durante o processo de otimização.

5 Considerações Finais

Tendo em mente o objetivo de otimizar o fluxo de uma malha viária de forma geral, e não apenas um único trajeto, este trabalho utilizou de métodos de busca local, e a implementação de um algoritmo para o uso dessa heurística.

O algoritmo apresentado é capaz de fazer alterações de uma malha, para um dado local, de tamanho pequeno melhorando o fluxo desta localidade. Para que o mesmo pudesse ser implementado, foi realizado um levantamento de trabalhos com temas similares utilizando outras heurísticas, bem como a definição do que deveria ser otimizado e o desenvolvimento em si.

Após realizar os experimentos, foi obtido um plano viário ótimo para cada conjunto de localidade, orçamento e tamanho de mapa. Com estes dados, se tornou possível observar o comportamento do algoritmo em diferentes situações.

Os objetivos deste trabalho foram alcançados, demonstrando uma implementação funcional da heurística, bem como uma análise sobre sua eficiência para a resolução de RNDPs.

5.1 Trabalhos Futuros

Alguns possíveis temas podem ser explorados em trabalhos futuros:

- Uma análise de heurística mista, com busca local como um componente. Isso permitiria a otimização da busca local operar dentro de um algoritmo geral com maior eficiência temporal.
- Uma análise de RNDPs utilizando o relevo para o algoritmo, trazendo um maior grau de realidade.
- Uma análise com adição de novas estradas (não apenas novas pistas). Assim como a opção acima, esta também poderia aproximar da realidade e trazer otimizações maiores, como por exemplo uma adição de um túnel, ou uma ponte.
- Uma análise desta heurística utilizando diferentes valores nos seus parâmetros. Poderia ser criada uma fórmula que retorna os parâmetros ideais para esta heurística a partir do tamanho do caso de estudo.
- Um estudo de heurística mista utilizando busca local para pequenas áreas.

Referências Bibliográficas

- BAI, R. et al. Optimisation of transportation service network using -node large neighbourhood search. **Computers Operations Research**, v. 89, p. 193–205, 2018. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054817301466>>. 22, 51
- FARAHANI, R. Z. et al. A review of urban transportation network design problems. **European Journal of Operational Research**, v. 229, n. 2, p. 281–302, 2013. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221713000106>>. 16
- INRIX. **Global Traffic Scorecard**. 2020. [Online; acessado em 28/04/2021]. Disponível em: <<https://inrix.com/scorecard/>>. 6, 7, 13, 45, 47
- LEBLANC, L. J. An algorithm for the discrete network design problem. **Transportation Science**, INFORMS, v. 9, n. 3, p. 183–199, 1975. Disponível em: <<http://www.jstor.org/stable/25767791>>. 16
- MADADI, B. et al. A bi-level model to optimize road networks for a mixture of manual and automated driving: An evolutionary local search algorithm. **Computer-Aided Civil and Infrastructure Engineering**, v. 35, n. 1, p. 80–96, 2020. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12498>>. 20, 49
- MT/DNER. **CUSTOS UNITÁRIOS MÉDIOS DAS OPERAÇÕES DE MANUTENÇÃO E REABILITAÇÃO DA MALHA RODOVIÁRIA FEDERAL**. 2007. [Online; acessado em 19/07/2022]. Disponível em: <<http://geipot.gov.br/>>. 25, 54
- PIRLOT, M. General local search methods. **European Journal of Operational Research**, v. 92, n. 3, p. 493–511, 1996. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0377221796000070>>. 17
- POORZAHEDY, H.; TURNQUIST, M. A. Approximate algorithms for the discrete network design problem. **Transportation Research Part B: Methodological**, v. 16, n. 1, p. 45–55, 1982. ISSN 0191-2615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0191261582900406>>. 16
- RITT, M. Busca heurística. In: _____. [s.n.], 2020. p. 13–42. Disponível em: <<https://www.inf.ufrgs.br/~mrpritt/lib/exe/fetch.php?media=cmp268:notas-11158.pdf>>. 8, 17, 18, 23, 52
- SALMAN, S.; ALASWAD, S. Alleviating road network congestion: Traffic pattern optimization using markov chain traffic assignment. **Computers Operations Research**, v. 99, p. 191–205, 2018. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054818301692>>. 8, 9, 13, 19, 21, 22, 28, 29, 47, 49, 50, 51, 57, 58
- Sindicato Nacional da Indústria de Componentes para Veículos Automotores - Sindipeças. **Relatório da Frota Circulante**. 2021. [Online; acessado em

28/04/2021]. Disponível em: <https://www.sindipecas.org.br/sindinews/Economia/2021/RelatorioFrotaCirculante_Marco_2021.pdf>. 6, 7, 13, 45, 47

6 Apêndices

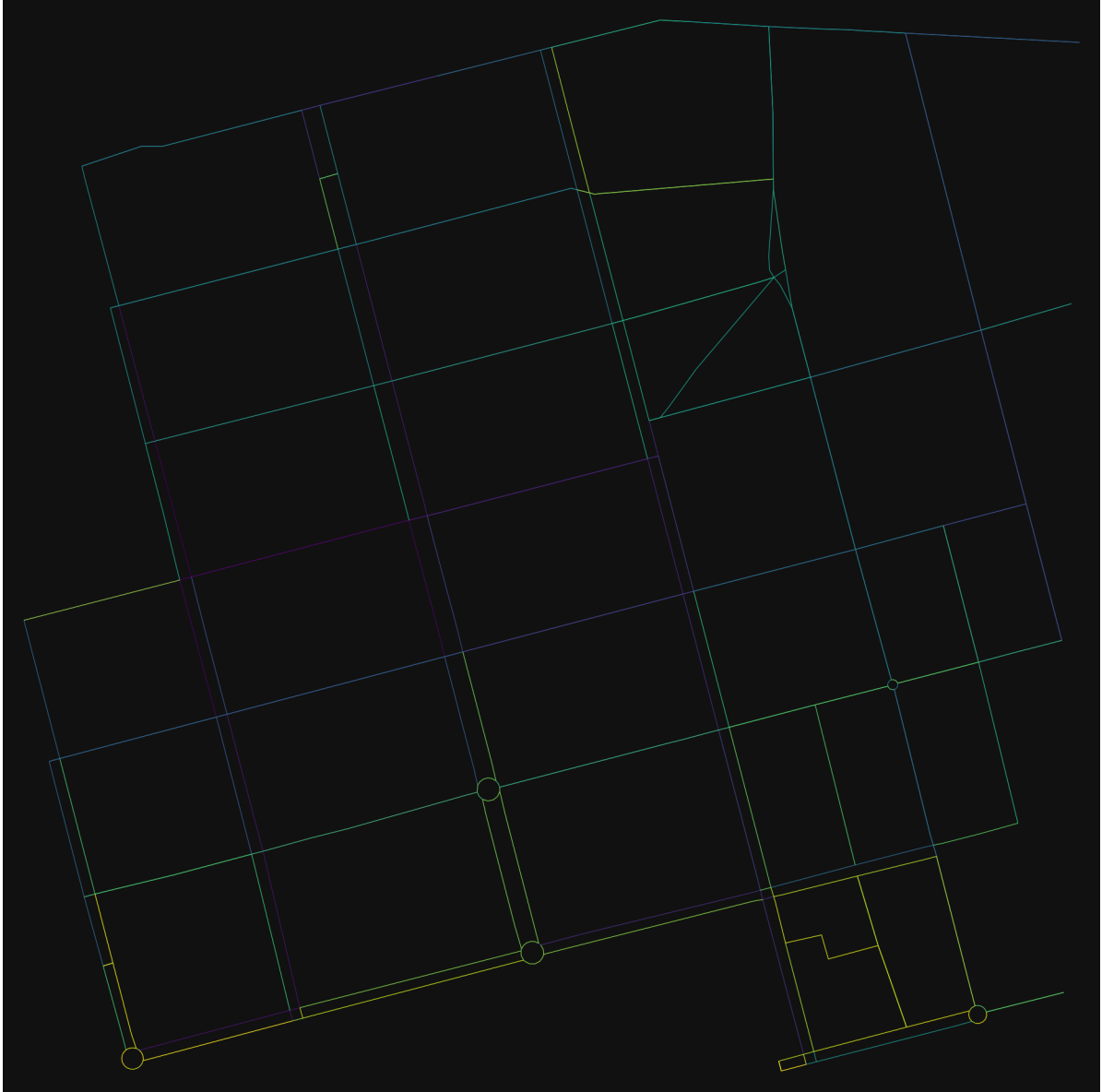


Figura 7 – Mapa de Chapecó com 450m de raio

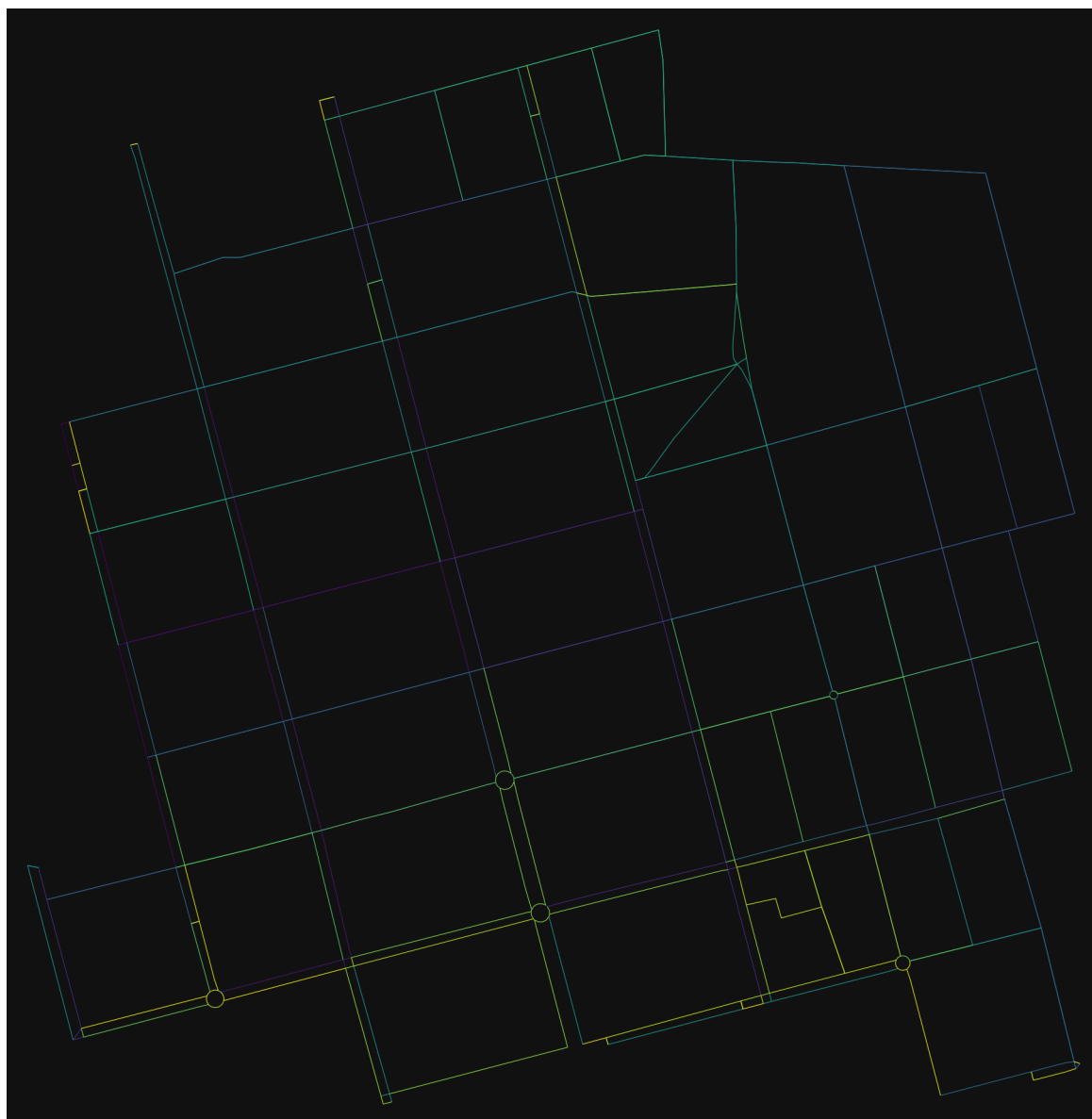


Figura 8 – Mapa de Chapecó com 550m de raio



Figura 9 – Mapa da região da UFSC com 450m de raio

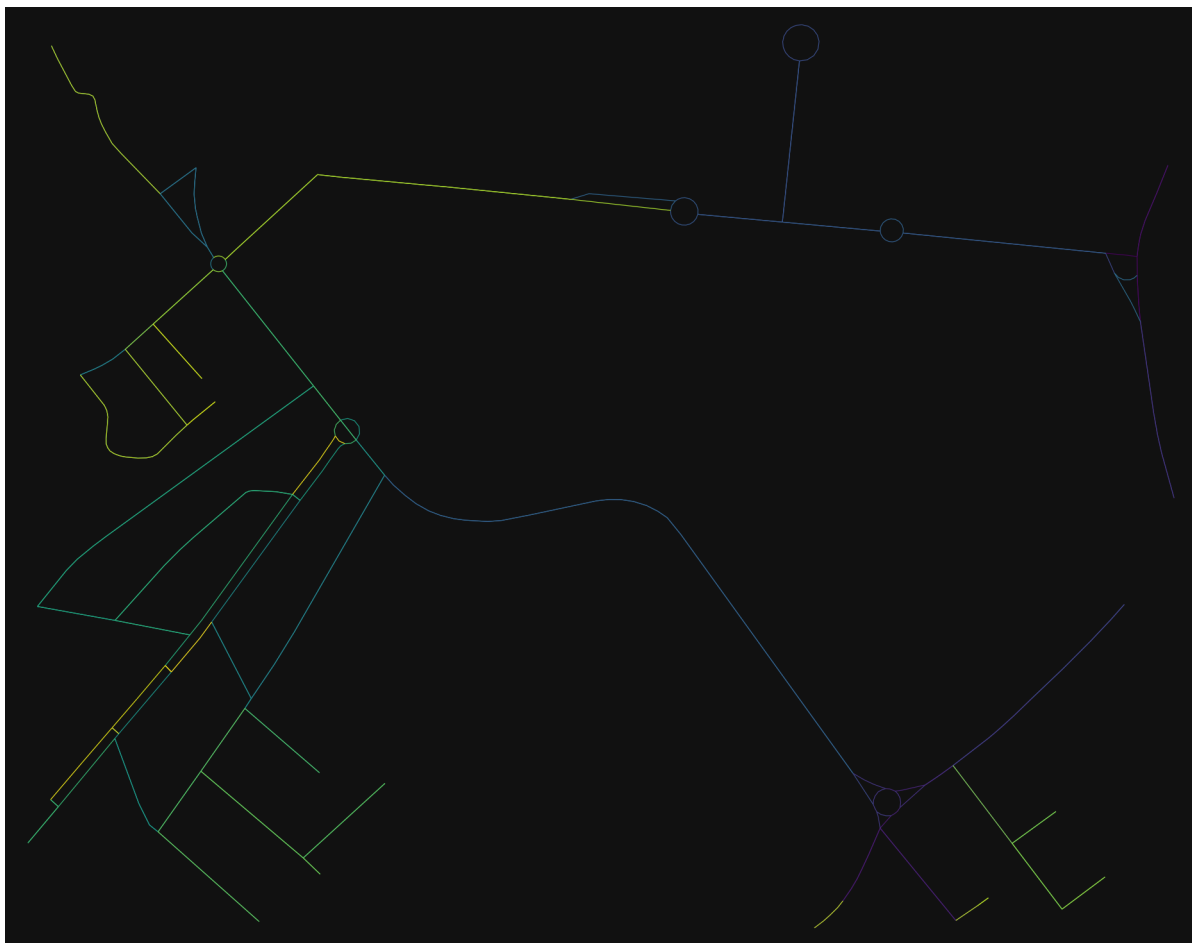


Figura 10 – Mapa da região da UFSC com 550m de raio

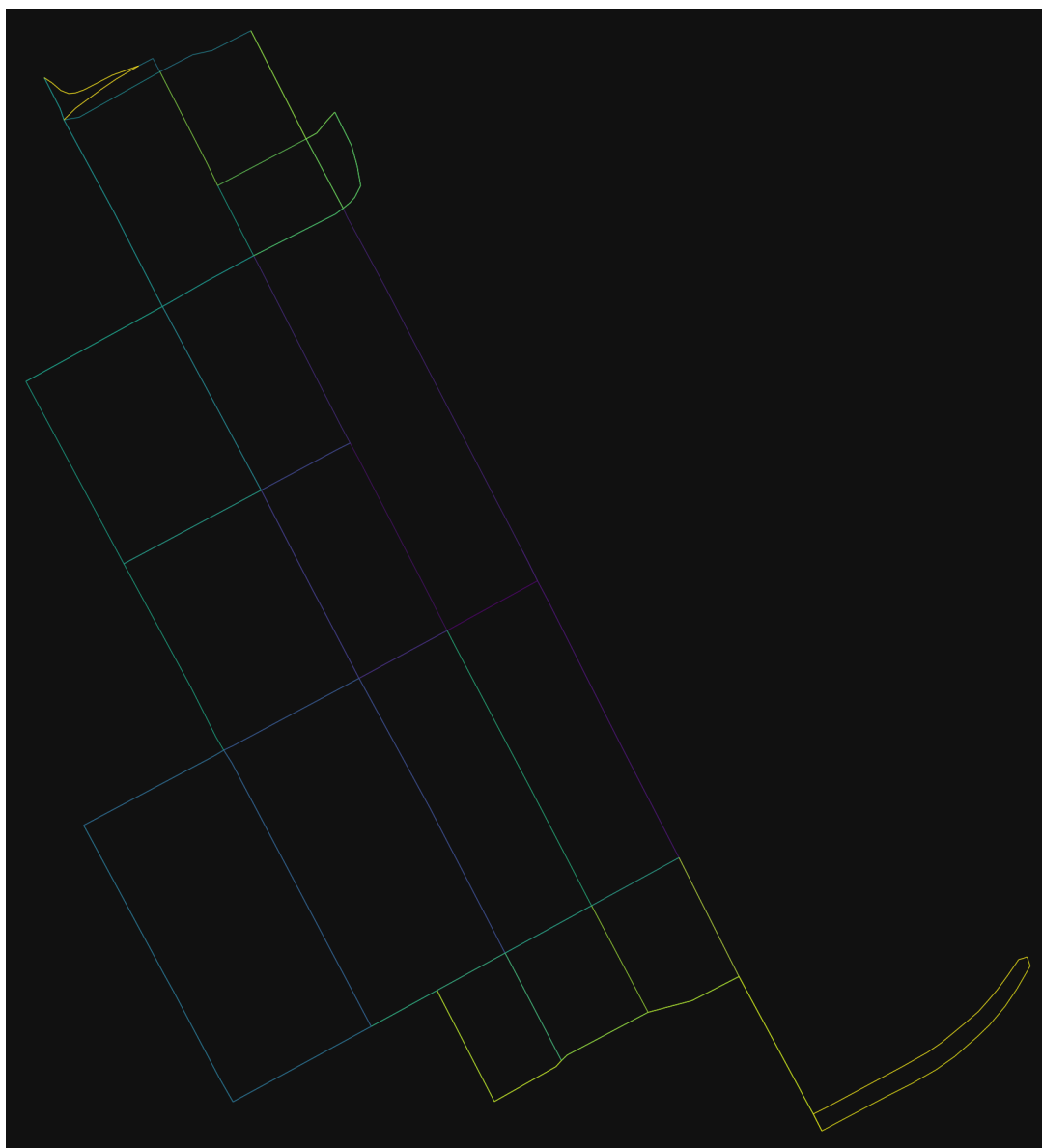


Figura 11 – mapa de Nashville com 450m de raio

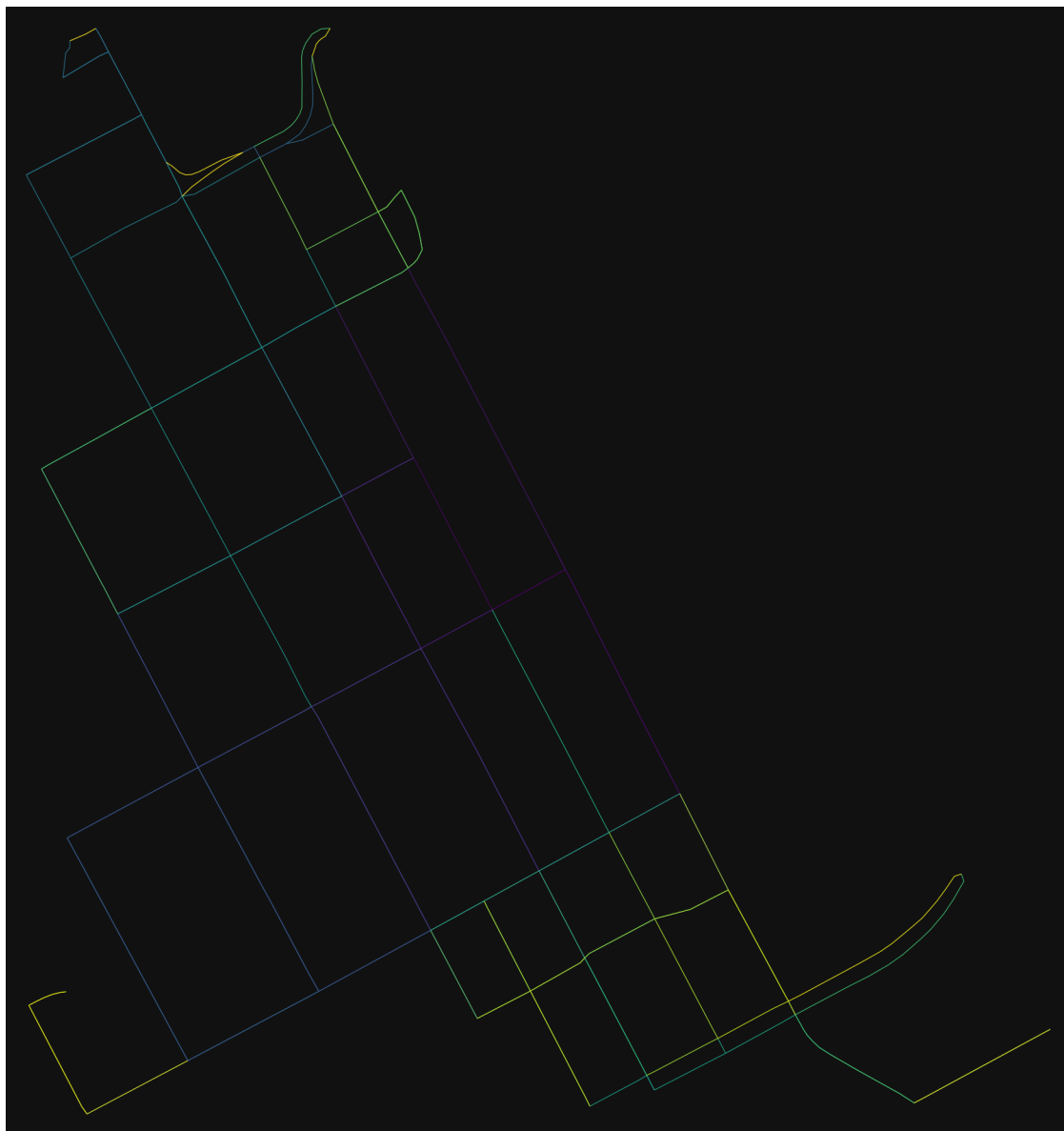


Figura 12 – mapa de Nashville com 550m de raio

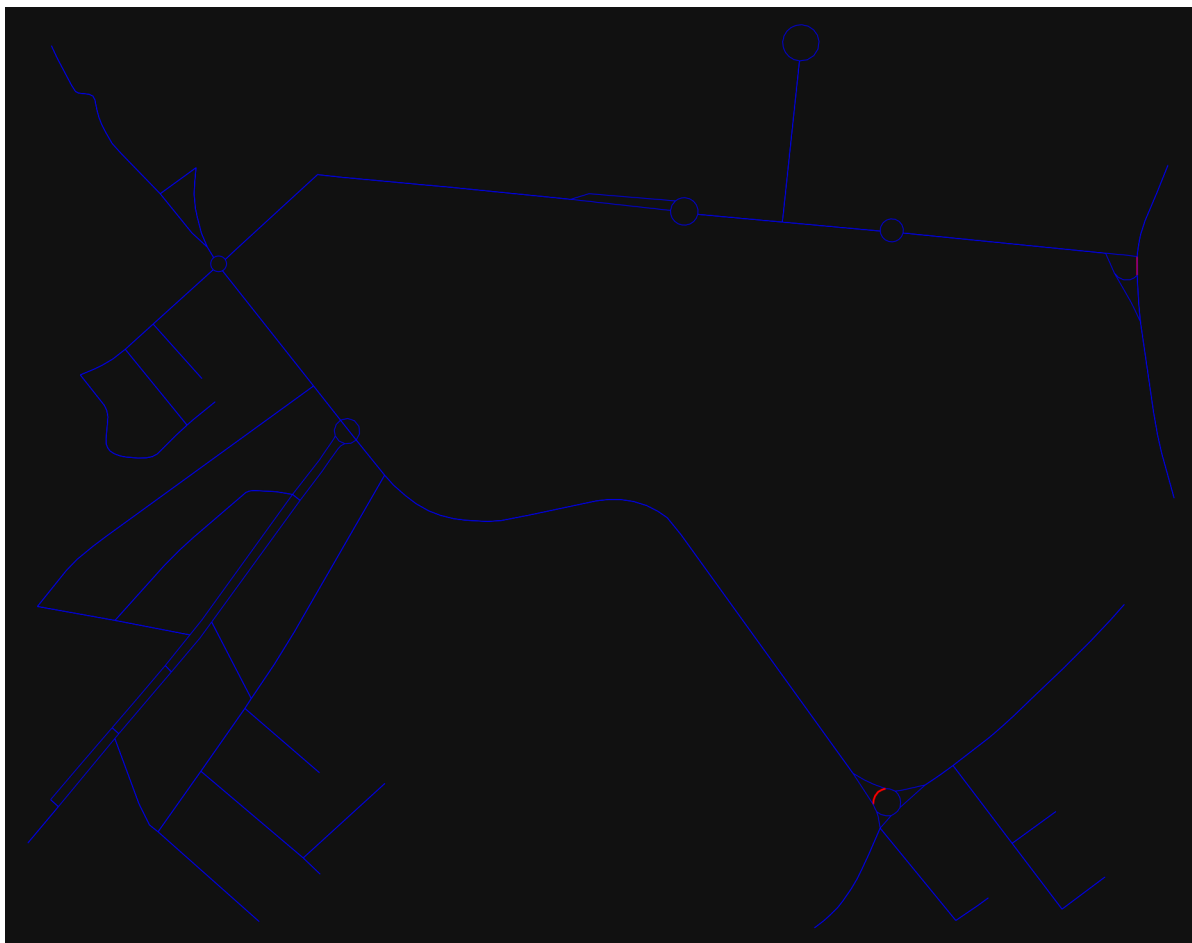


Figura 13 – Mapa da UFSC otimizado com R\$ 10.000

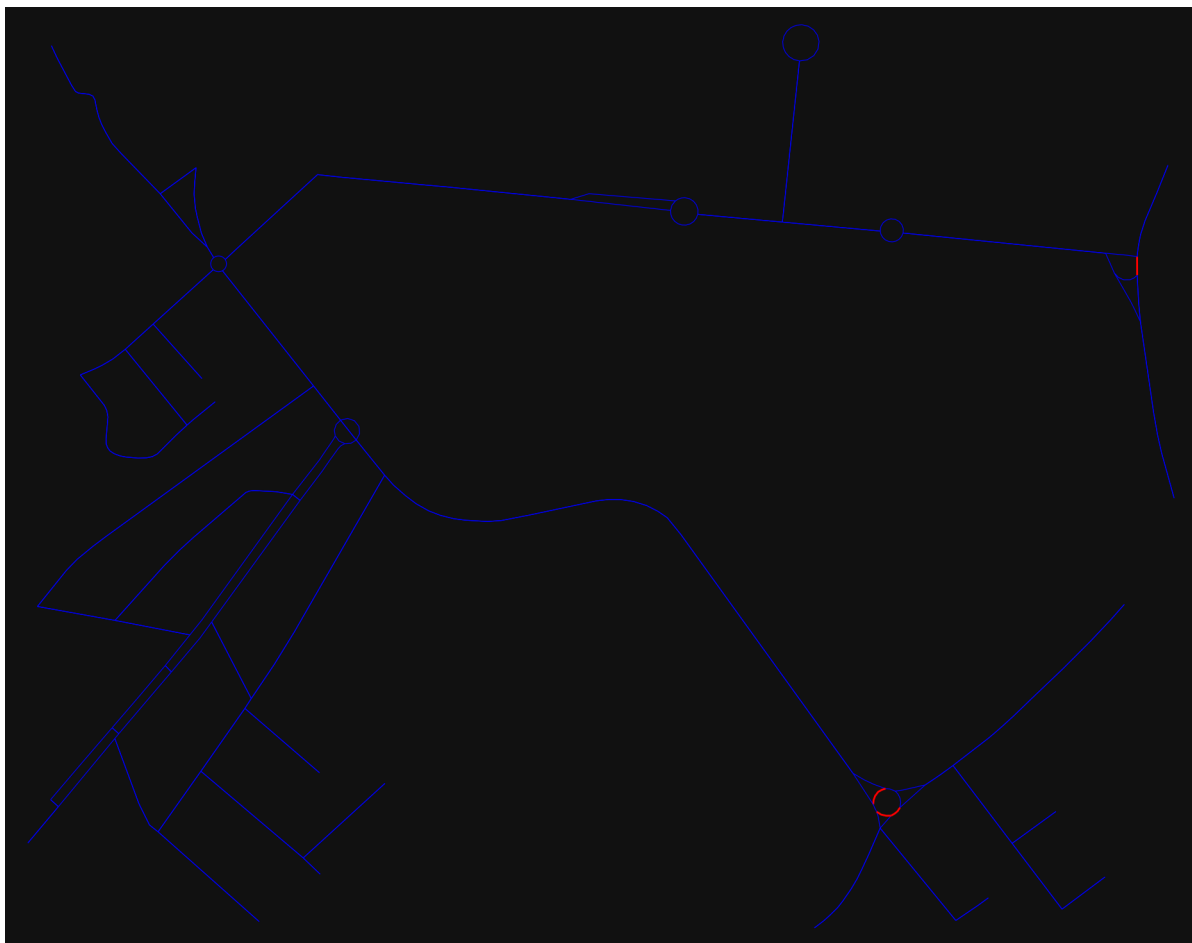


Figura 14 – Mapa da UFSC otimizado com R\$ 50.000

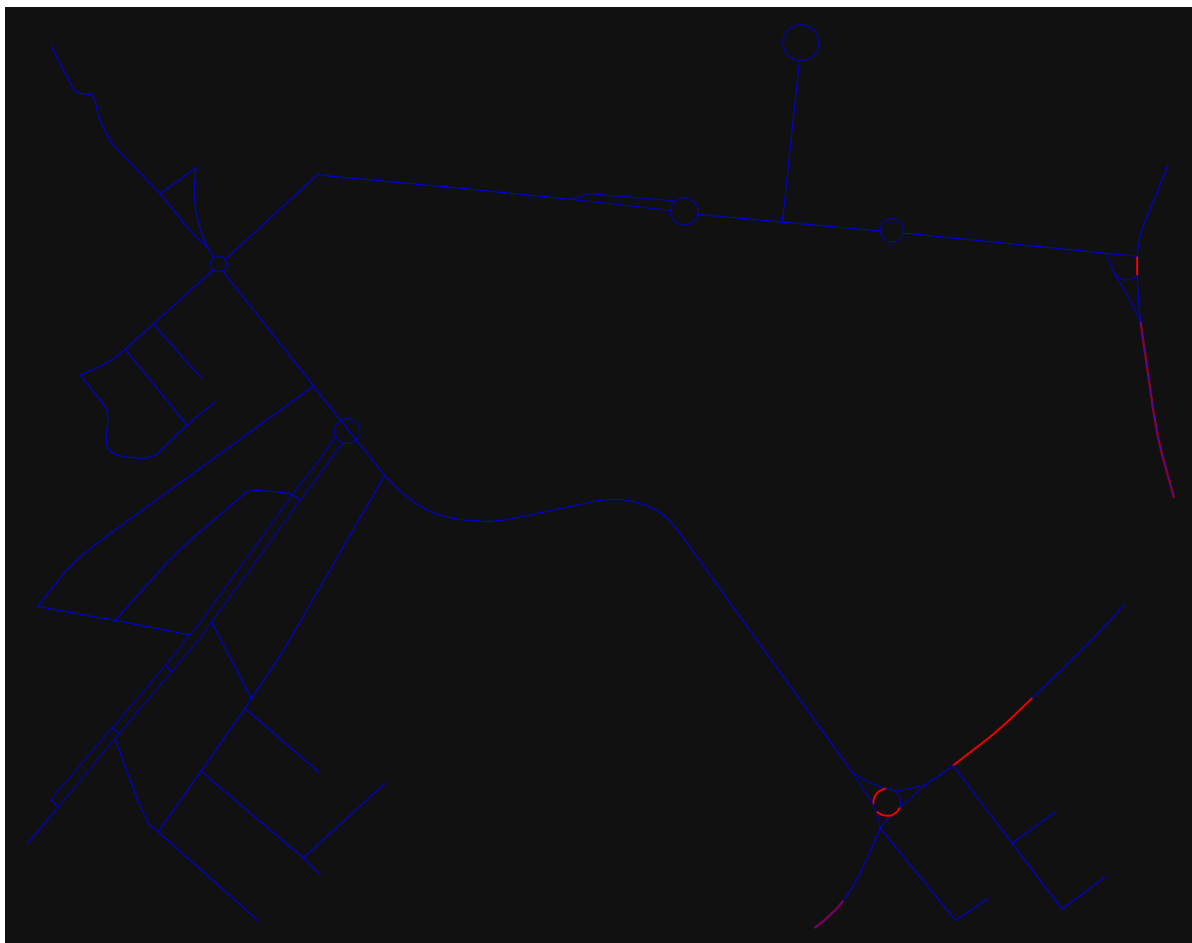


Figura 15 – Mapa da UFSC otimizado com R\$ 150.000

[12pt]article

sbc-template

graphicx,url

[utf8]inputenc

UFSC - Universidade Federal de Santa Catarina pedro.j.v.souza@grad.ufsc.br,
r.santiago@ufsc.br elder.santos@ufsc.br, pedro.castellucci@ufsc.br

HEURÍSTICA DE BUSCA LOCAL PARA ALTERAÇÃO, ADIÇÃO E REMOÇÃO DE ESTRADAS EM PROJETOS DE MALHA VIÁRIA TERRESTRE

Pedro José Vieira de Souza, Rafael de Santiago, Elder Rizzon Santos, Pedro Belin

2021

Resumo

According to (Sindicato Nacional da Indústria de Componentes para Veículos Automotores - Sindipeças, 2021), the amount of vehicles in Brazil grows every year, implying in heavier traffic, especially in urban areas. This increase in traffic has a high financial impact, as seen in (INRIX, 2020), costing the average US driver over \$1.000 a year. With this in mind, this work tries to optimize urban road networks aiming to reduce traffic and the average vehicle commute time by using a local search heuristic. To reach this goal, a review of recent literature on RNDPs was made, as well as a review on local search implementations. Then a new model was proposed and applied over a few real urban networks to better visualize how this model can be applied.

Resumo

De acordo com (Sindicato Nacional da Indústria de Componentes para Veículos Automotores - Sindipeças, 2021), a frota de veículos aumenta anualmente no Brasil, implicando em cada vez mais congestionamento nas grandes cidades. De acordo com (INRIX, 2020) isto tem um impacto financeiro alto, chegando a mais de \$1.000 por motorista estadunidense. Nesse contexto, este trabalho busca otimizar as malhas viárias urbanas para a redução de congestionamento e a diminuição dos tempos de comutação dos veículos utilizando uma heurística de busca local. Para a realização do mesmo, foi feita uma revisão da literatura

recente sobre problemas de design de redes, assim como sobre diferentes implementações de busca local. Foi então proposto o modelo a ser utilizado e realizados testes para trazer um melhor entendimento de como aplicar o modelo sobre malhas existentes.

6.1 Introdução

A frota circulante de veículos automotores aumenta ano após ano no Brasil, aumentando em 2,5% de 2018 para 2019 e em 0,7% de de 2019 para 2020 (Sindicato Nacional da Indústria de Componentes para Veículos Automotores - Sindipeças, 2021). Com este aumento constante da frota lidar com o congestionamento das vias se torna um desafio para as cidades brasileiras e deve-se portanto ser desenvolvida alguma estratégia concreta que consiga otimizar as vias e auxiliar as gestões municipais a lidar com o trânsito.

De acordo com um estudo da INRIX realizado em 2020, o congestionamento nos Estados Unidos custou um média 1.347 dólares por motorista em 2019(INRIX, 2020), com esta observação, pode-se perceber que o engarrafamento é indesejável não apenas pelo desconforto do motorista, mas por ser economicamente negativo para os cidadãos. Dentro do campo de ciências da computação podemos abordar este problema de diversas maneiras, neste trabalho iremos tratá-lo como um *Network Design Problem* (NDP).

Solucionar um NDP significa encontrar modificações para serem feitas sobre os arcos, buscando algum objetivo bem definido como por exemplo otimizar o fluxo total da rede ou então o fluxo de uma sub-rede.

Um exemplo de otimização viária terrestre utilizando uma abordagem de NDPs pode ser visto na literatura em Salman-Alaswad(2018)(SALMAN; ALASWAD, 2018), onde é utilizada uma meta-heurística genética para buscar uma solução com um fluxo médio da rede otimizado.

Neste trabalho é proposta uma abordagem diferente, utilizando uma heurística de busca local de vizinhança. Algoritmos de busca local são algoritmos que buscam novas soluções para um problema realizando pequenas modificações (chamadas de movimentos) a soluções anteriores. Quando se tem em mente malhas viárias reais devemos sempre buscar realizar o mínimo de movimentos para chegar numa otimização, já que cada mudança na rede acarreta em um aumento do orçamento necessário para concretizar o projeto e, por consequência, grandes mudanças trazem grandes custos, podendo inviabilizar a execução do projeto.

6.1.1 Metodologia

O presente trabalho se trata de uma pesquisa exploratória, adaptando uma heurística de busca local a um novo problema.

Os seguintes passos foram feitos para a realização deste trabalho:

Um levantamento teórico amplo, visando trazer credibilidade a pesquisa realizada. Nesta fase, trabalhos prévios com definições das heurísticas utilizadas serão apresentados para que as definições fiquem claras e consistentes com a literatura. A busca de artigos será feita em portais e repositórios como o Google Scholar, e o SCOPUS.

Após o estudo dos trabalhos anteriores, será feita a definição do novo modelo, apresentando o embasamento por trás das decisões feitas, assim como a especificação da metodologia dos testes. Devem ser definidas as métricas que serão utilizadas para permitir a comparação com os trabalhos prévios, e também os casos de teste que passarão pela nova abordagem.

Em seguida será desenvolvida uma aplicação para implementar a nova abordagem, assim como os casos de testes. Ambas estas aplicações devem ser disponibilizadas junto com este trabalho, garantindo assim a transparência quanto a legitimidade dos testes, assim como a possibilidade de replicação dos mesmos.

O último passo, é a análise dos resultados, buscando conclusões sobre a eficiência e eficácia da nova abordagem. A partir das quais será possível observar em quais situações esse método pode ser utilizado.

6.2 Trabalhos Relacionados

Neste capítulo, iremos selecionar trabalhos relacionados a este, tendo como objetivo trazer uma visão do que foi desenvolvido nesta área nos últimos anos, e qual contribuição este trabalho pode trazer.

Para encontrar os trabalhos prévios com um bom nível de correlação, utilizaremos duas palavras chave no portal SCOPUS: "Network Design Problem" e "Local Search". Ao realizar esta busca apenas 104 documentos são encontrados, este número razoavelmente baixo nos mostra que nossas palavras chave foram suficientemente limitantes.

O critério de seleção dos trabalhos será a relevância de acordo com o SCOPUS, passando por uma filtragem do autor, buscando garantir que os trabalhos sejam realmente relacionados com a pesquisa aqui desenvolvida.

6.2.1 A bi-level model to optimize road networks for a mixture of manual and automated driving: An evolutionary local search algorithm

O trabalho desenvolvido por Madadi et al. (2020), aborda pontos importantes sobre a integração do transporte autônomo com as modalidades de veículos tradicionais. São definidos conceitos de valor do tempo de viagem (VoTT) para veículos autônomos e tradicionais, sendo o VoTT de transporte por veículos autônomos mais baixo, já que o tempo do transporte pode ser utilizado para realizar outras tarefas e é portanto menos limitante que o transporte tradicional. Os autores citam trabalhos mostrando a improbabilidade da existência de veículos autônomos funcionais em todo tipo de estrada no futuro próximo e definem como objetivo determinar uma sub-rede para veículos autônomos dentro de uma dada rede de forma que apenas esta sub-rede precise estar dentro do padrão necessário para o funcionamento dos veículos autônomos. Esta limitação permite uma redução no custo de atualização das vias. No trabalho são comparadas três abordagens, sendo duas delas algoritmos genéticos (GA e MGA) e uma por busca local evolutiva (ELS). Também é utilizada uma taxa de penetração variável, que indica a porcentagem da rede que deveria ser adaptada para capacitar o trânsito de veículos autônomos.

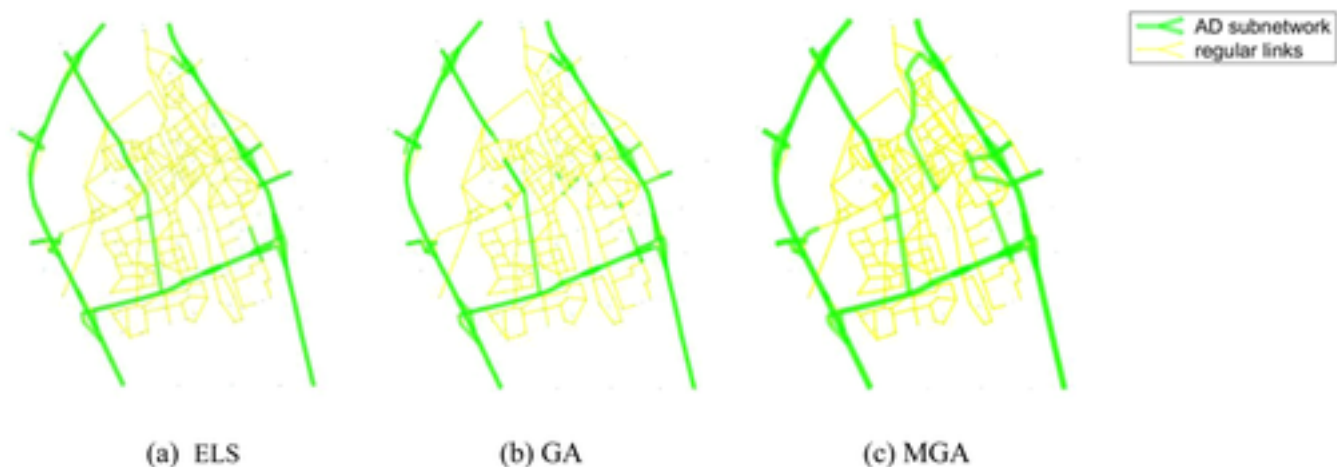


Figura 16 – Exemplo com mais nodos e 10% de penetração - Encontrado em Salman e Alaswad (2018)

Podemos ver nas figuras (16 e 17) a diferença entre os algoritmos, em verde temos as vias a serem adaptadas buscando minimizar o custo de viagem na rede fornecida, enquanto tenta-se também minimizar o custo de adaptação. O tempo computacional dos algoritmos avaliados é consideravelmente alto, com a execução mais rápida sendo a do ELS-10% em 76 minutos, e a mais lenta sendo a MGA-50% em 968 minutos. Vale notar que existe no trabalho uma métrica de "Função objetivo" que toma em tanto o custo total de viagem e o custo total de adaptação onde todas as abordagens tiveram resultados similares. A razão pela qual esta métrica é notável dentro do trabalho que está sendo desenvolvido é

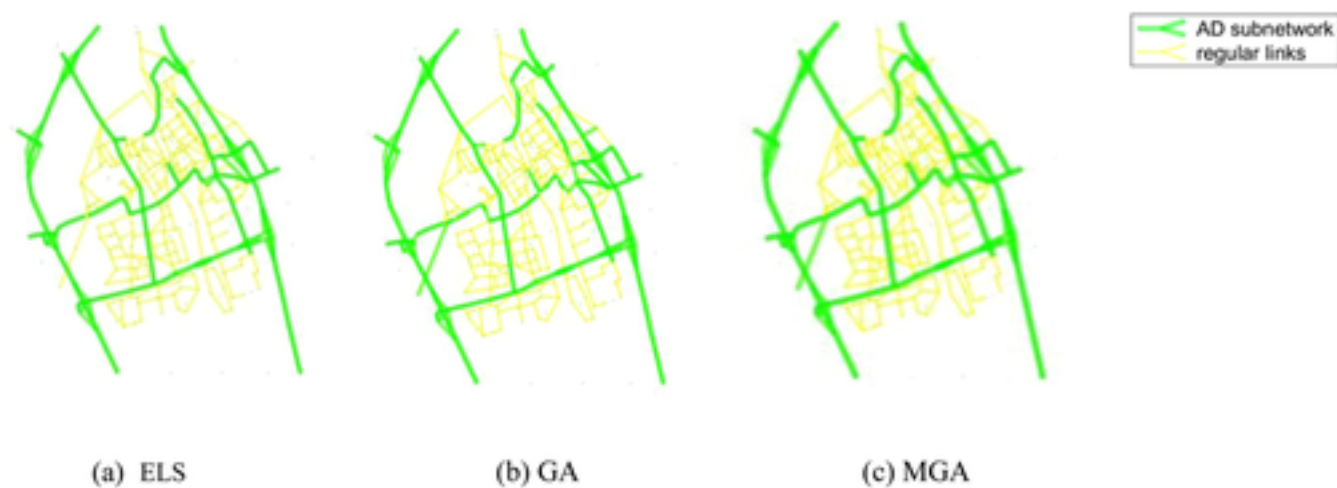


Figura 17 – Exemplo com mais nodos e 90% de penetração - Encontrado em Salman e Alaswad (2018)

que ela também pode ser utilizada aqui. Mesmo com os trabalhos abordando problemas significativamente diferentes esta métrica serve como um bom ajuste para o custo-benefício de uma solução, tendo em vista o retorno a longo prazo que um custo de viagem reduzido traz.

6.2.2 Optimisation of transportation service network using k-node large neighbourhood search

Este trabalho levanta pontos interessantes sobre a pesquisa de Service Network Design Problems, que podem ser expandidos para NDPs de forma geral. De acordo com Bai et al. (2018) mesmo existindo uma grande quantidade de pesquisa sobre NDPs e otimização de rotas dentro das redes, a maior parte dos trabalhos se limita a encontrar uma nova heurística para abordar uma mesma modelagem de vizinhança independente da abordagem. Os autores propõem então uma modificação vista como mais baixo nível, buscando otimizar a própria função de vizinhança. Embora o problema abordado de SNDPs seja significativamente diferente dos problemas RNDP abordados neste trabalho, é necessário observar a possível utilidade de fazer mudanças nas vizinhanças dentro de RNDPs também, pois potencialmente existem otimizações a serem feitas neste campo que poderiam afetar diversos problemas, não importando qual for o problema abordado ou implementação específica. Os autores por fim utilizam uma implementação híbrida, utilizando diferentes passos para encontrar a melhor solução, onde um dos passos é a otimização de vizinhanças para SNDPs proposta no trabalho, demonstrando a possibilidade de utilizar otimizações de vizinhança junto a estratégias meta-heurísticas. Os resultados para os testes com essa abordagem híbrida foram melhores que ambas as estratégias separadas, devido ao alto custo computacional do método apresentado, e da acurácia mais baixa da busca Tabu demonstrada no trabalho.

6.2.3 Discussão dos trabalhos

A partir dos trabalhos apresentados nas seções anteriores, percebe-se um padrão quando se trata de buscas locais que é o uso de mais de uma estratégia, organizando-as em etapas para melhor aproveitar as vantagens trazidas por cada uma. Os artigos também levantam pontos importantes, como diferentes funções de objetivo que não o fluxo médio da rede (como o método baseado em densidade de veículos em Salman e Alaswad (2018)) sendo então interessante para trabalhos que tragam novas abordagens sobre RNDPS observarem se os métodos que estão desenvolvendo podem ser generalizados para qualquer variável de objetivo e até mesmo se outras variáveis não são mais compatíveis com o modelo sendo desenvolvido.

6.3 Especificação e Métodos

Neste capítulo, serão apresentados o modelo desenvolvido neste trabalho, bem como os métodos utilizados para realizar os testes.

6.3.1 Busca local em RNDP

A primeira e mais importante definição de busca local é quanto a monotonia do modelo que será utilizado. Ao analisar as vantagens e desvantagens dos modelos, estipula-se que heurísticas monótonas se encaixem melhor ao escopo do problema. Isso se deve a minimização de mudanças que estas trazem, já que os RNDPs não podem sofrer modificações muito drásticas em suas soluções.

Um algoritmo genérico de busca local monótona pode ser encontrado em Ritt (2020)

Algorithm 4: Algoritmo Genérico de Busca Local Monótona

Data: Solução inicial s , vizinhança N , uma distribuição P , uma função de otimização α

Result: Saída: Uma solução com valor de otimização no máximo $\alpha(s)$

```

1  $s^* \leftarrow s$ ;
2 while  $\neg(P(s) = 1)$  do
3   | seleciona  $s' \in N(s)$  de acordo com  $P(s)$ ;
4   |  $s \leftarrow s'$ ;
5   | if  $\alpha(s) < \alpha(s^*)$  then
6   |   |  $s^* \leftarrow s$ ;
7   |   end
8 end
9 return  $s^*$ 

```

Este algoritmo pode ser adequado a uma RNPD utilizando o seguinte mapeamento para as variáveis:

- $s \leftarrow$ solução inicial do caso de teste;
- $N \leftarrow$ conjunto de soluções a um movimento de distância, para todos os possíveis movimentos;
- $P \leftarrow$ uma função de distribuição a ser desenvolvida;
- $\alpha \leftarrow$ o tempo médio das viagens dentro da rede

6.3.2 Experimentos

Para os experimentos, foi desenvolvida uma aplicação em *python* implementando a heurística a ser testada. À priori foi construída uma abordagem com uma heurística de busca local simples, como o visto em 1, tendo como critério de parada o gasto máximo permitido para mudanças no projeto de malhas, também parando a execução caso após uma iteração não seja encontrada nenhuma melhoria.

Foram propostos casos de testes com diferentes tamanhos, orçamentos e regiões. Para todos os casos propostos, foi gerado um grafo do mapa a partir de dados coletados do

Algorithm 5: Algoritmo para calculo de média de tempo

```
Data: Grafo direcionado G, valor minimo V  
Result: Saída: média de tempo de viagem M  
1 nodes ← G.nodes;  
2 total_time ← 0;  
3 total_routes ← 0;  
4 for origin ∈ nodes do  
5   for dest ∈ nodes do  
6     if origin ≠ dest then  
7       route_time ← shortest_path(origin, dest);  
8       if route_time ≥ V then  
9         total_time ← total_time + route_time;  
10        total_routes ← total_routes + 1;  
11      end  
12    end  
13  end  
14 end  
15 M ← total_time/total_routes;  
16 return M;
```

OpenStreetMap (OSM). Sobre esse grafo é analisado o tempo médio dos trajetos, utilizando o algoritmo 16. Tendo este valor salvo, podemos fazer uma busca local exaustiva para todas as possíveis mudanças, no escopo deste trabalho foram inclusas a inversão de pista, a remoção de uma pista e a adição de uma pista. Para cada mudança feita este tempo é recalculado e, caso seja melhor que o mapa inicial e o candidato atual de mudança, é eleito como a próxima alteração a ser feita ao mapa. Ao final de um ciclo têm-se a melhor mudança viável para o mapa.

6.3.3 Otimização de tempo de viagem

Para a primeira bateria de testes, definimos três localidades (UFSC, Chapecó e Nashville), bem como dois possíveis raios (450m e 550m) e três níveis de orçamento (10, 50 e 150 mil reais). O custo de modificação de cada km de via foi utilizada uma tabela de custos do MT/DNER (2007). A partir deste custo e do comprimento da via a ser alterada é determinado o preço da obra para uma dada modificação.

Os resultados destes testes podem ser observados nas Tabelas 13, 14, 15, 16. Pode-se observar a média inicial do tempo das viagens, a média otimizada, o número de modificações feitas e o tempo de execução. Rapidamente percebe-se que essa otimização não parece muito grande, sendo em média uma redução de apenas 0.05% no tempo médio de viagem na malha. Isso é natural, visto que qualquer mudança feita em um único arco não vai afetar a maior parte das viagens. Também percebe-se um alto tempo de execução, sendo em média 52.8s por passo dado nas malhas de pequeno porte (450m), e de 883s (14m43s) por passo nas malhas de porte maior (600m). Este crescimento rápido com o aumento dos nodos é também natural das buscas locais exaustivas, porém a heurística se mostra inviável para malhas de grandes portes.

As Figuras 7, 8, 9, 10, 11 12 mostram os mapas antes de serem otimizados (com um *colormap* somente ilustrativo).

As Figuras 13, 14, e 15 nos mostram os mapas de otimização da UFSC: 550m em diferentes níveis de orçamento, com as pistas modificadas em vermelho, e as não modificadas em azul. Especificamente na figura 15 podemos observar uma pista onde uma de suas faixas foi modificada e a outra não, gerando uma estrada em roxo. Podemos observar que são mudanças pequenas em relação ao tamanho total da malha.

Tabela 13 – UFSC: 450m (48 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	98.20553420445817	98.10107692307687	R\$ 3.913,15	2	127,6629s
R\$ 50.000	98.10107692307687	98.10107692307706	R\$ 45.315,20	2	137,3100s
R\$ 150.000	98.10107692307687	98.10107692307680	R\$ 146,452.20	1	92,3190s

Tabela 14 – UFSC: 550m (84 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	116.62899875363496	116.54137931034467	R\$ 1.417,15	2	1379,3913s
R\$ 50.000	116.62899875363496	116.54137931034452	R\$ 33,375.20	3	1900,8265s
R\$ 150.000	116.62899875363496	116.54137931034467	R\$ 37,624.60	3	1850,0693s

Tabela 15 – Chapecó: 450m (119 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	75.0378384426048	75.0378384426048	R\$ 10.000	0	30,3279
R\$ 50.000	75.0378384426048	75.0378384426048	R\$ 50.000	0	31,6178
R\$ 150.000	75.0378384426048	75.0378384426048	R\$ 150.000	0	29,7678

Tabela 16 – Chapecó: 550m (171 nodos)

Orçamento	Média Inicial	Melhor Média	Dinheiro Restante	Número de Mudanças	Tempo de Execução
R\$ 10.000	81.17417908178834	81.15966246008828	R\$ 8,611.75	1	1654,0852s
R\$ 50.000	81.17417908178834	81.15966246008819	R\$ 39,802.70	1	1142,8250s
R\$ 150.000	81.17417908178834	81.15966246008806	R\$ 98,554.09	2	1122,2729s

Pode-se observar na diferença dos tempos de execução, por exemplo entre as tabelas 14, e 16, que o custo computacional desta abordagem é bastante alta para mapas com uma quantidade elevada de nodos, este comportamento é esperado da heurística de busca local, porém ele limita o escopo do trabalho para pequenas regiões. Para lidar com essa limitação, o mapa pode ser redesenhado como uma série de regiões pequenas (com menos de 150 nodos por exemplo) interconectadas por um mapa maior que inclua apenas estradas centrais da cidade, como as SCs na grande Florianópolis. Ao reimaginar o mapa desta forma, podem ser feitas múltiplas execuções mais curtas do algoritmo, ao invés de uma longa execução.

O tempo de execução cresce muito rapidamente ao adicionar mais nodos ao grafo, isso fica claro no algoritmo 16, onde a função *shortest_path* implementa a busca de menor rota através do algoritmo de Dijkstra, tendo sozinha uma complexidade de $O(A+V\log V)$, onde A é o número de arcos e V o número de Nodos. Além disso, o algoritmo busca a rota mais curta para todos os pares de nodos, aumentando essa complexidade por V^2 vezes. Finalmente, esse cálculo de média de tempo deve ser executado cada vez que uma mudança é feita nas vias, para verificar se houve melhoria. No caso dos experimentos atuais, são feitas três mudanças por arco em cada ciclo, aumentando então a complexidade em $3A$ vezes trazendo a nossa complexidade final para $O(V^2A^2 + AV^3\log V)$. Sabendo da complexidade do algoritmo, pode-se afirmar que as execuções de diferentes regiões

menores, e interconectadas seriam sim mais rápidas do que uma execução de um único grande mapa, devido a natureza cúbica do problema.

Também pode-se perceber que o número de mudanças relevantes encontradas pelo método exerce uma influência no tempo de execução. Isso fica bastante claro ao comparar as Tabelas 14 e 15 onde mesmo com mais nodos o mapa da segunda tabela tem um tempo de execução curtíssimo. Isso se deve ao fato de que não existiam mudanças relevantes para o tempo médio de viagem na malha, causando o algoritmo a realizar apenas uma passagem inicial sobre a malha.

6.3.4 Otimização de densidade de veículos

Propõe-se uma segunda próxima bateria de testes, onde a variável a ser otimizada é alterada, ao invés de trabalhar sobre o *traveltime* médio da malha, o foco será a densidade média de veículos da malha (como visto em Salman e Alaswad (2018)), esta decisão de mudança é devido ao fato de que a otimização da média do tempo de viagem é, de forma geral, bastante baixa e por vezes pode não realmente demonstrar uma melhoria na malha de forma geral (e.g. um cenário onde uma melhoria grande suficiente de um único *traveltime* A->B melhora a média porém piora o *traveltime* de diversas outras rotas), já o método de densidade de veículos tem como motivação a dispersão do congestionamento, trazendo uma melhoria mais ampla para a malha viária observada.

Para o cálculo dessas densidades foi utilizado o método também encontrado em Salman e Alaswad (2018) utilizando cadeias de Markov, onde a probabilidade de um veículo depende do comprimento da via onde ele está, bem como das vias para onde ele pode ir, onde uma maior quilometragem aumenta a probabilidade de ser uma via destino. Pode-se pensar nessa definição como a seguinte função de probabilidade

$$\forall e, f \in A$$

$$P(e, f) = \begin{cases} 0 & \text{quando } v(e) \neq u(f) \\ \frac{(length(f)*lanes(f))}{\sum_{i=0}^n P(e,x(i))} & \text{quando } v(e) = u(f) \end{cases}$$

- A o conjunto de arcos do grafo
- x uma função que retorna um arco de acordo com seu índice
- *length* uma função que retorna o comprimento do arco
- *lanes* uma função que retorna o numero de pistas do arco
- u e v retornam os nodos de origem e destino de um arco, respectivamente

Para a definição de classes de ocupação de uma pista (visualizada nas imagens) podemos observar a tabela 18:

Tabela 17 – Highway level of service(LOS) and congestion conditions

LOS	Traffic Conditions	Volume/Capacity Ratio	Maximum Density (vehichle/km/lane)
A	Free	0.35	7
B	Stable	0.54	11
C	Stable	0.77	16
D	High Density	0.93	22
E	Near Capacity	1.00	28
F	Breakdown	Unstable	>28

Tabela 18 – Tabela encontrada em Salman e Alaswad (2018)

A partir desta tabela e do método da cadeia de Markov acima, foram geradas as densidades de veículos para cada arco dos mapas tanto antes quanto depois das modificações serem realizadas. A variável a ser otimizada é a soma dos dez maiores valores dessa lista, assim evita-se que o algoritmo simplesmente alterne a via onde se encontra o congestionamento. Além disso, foi removido o calculo de custos para essa segunda bateria de testes, essa mudança foi motivada tanto para redução de operações em máquina sendo feitas (trazendo mais velocidade para a execução de uma análise) e também tendo em vista de que o custo estimado poderia ser longe da realidade dependendo do local sendo analisado e portanto ter o custo como um critério de parada não auxiliava realmente a solucionar o problema em questão. Foi utilizado então somente quantidades fornecidas de metros que o algoritmo poderia modificar, para os testes foram utilizados 200, 500 e 1000 metros de orçamento.

Este método obteve resultados de maior qualidade, como podemos ver nas Tabelas 19, 20, 21, 22, 23, 24. Obtendo reduções da métrica observada magnitudes acima do que as reduções dos testes realizados na primeira bateria, com uma média de redução de 10.60% nos mapas de 450 metros e de 3.85% nos mapas de 600 metros. Pode-se observar na Tabela 20 um comportamento inesperado onde foi obtido um tempo total de execução de duas horas para um único mapa, mas ao realizar uma análise mais profunda observando o tempo de execução por passo, encontramos uma média de 361s por passo noas mapas de 600m, uma redução de 59% em relação ao método testado na primeira bateria.

Analisando mais a fundo o algoritmo utilizado, é perceptível o motivo de uma redução tão significativa no tempo de execução. O algoritmo utilizado para a segunda bateria de testes não depende do numero de nodos, dependendo apenas do numero de arcos de um dado mapa. Além disso sua complexidade é bastante menor, tendo uma complexidade total de $O(A^3 + A^2)$ ou seja, seu crescimento é cúbico em relação as arestas, enquanto na primeira bateria tinha-se um de grau muito maior ($A^2V^2 + AV^3 \log V$).

Tabela 19 – Chapecó - 450m (119 nodos, 257 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	784.9107000000004	764.9800000000006	2	447,502033s
500	784.9107000000004	731.3361750000001	10	1482,678092s
1000	784.9107000000004	711.4766749999999	11	1543,163062s

Tabela 20 – Chapecó - 600m (219 nodos, 482 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	1447.8596000000014	1427.8835000000013	2	1934,328336s
500	1447.8596000000014	1393.3651000000011	5	3901,393258s
1000	1447.8596000000014	1356.712125000001	13	9660,576185s

Tabela 21 – Nashville - 450m (36 nodos, 89 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	450.4251749999999	428.9938749999999	4	32,995950s
500	450.4251749999999	396.6446499999999	4	30,193408
1000	450.4251749999999	372.6997249999994	5	33,621320

Tabela 22 – Nashville - 600m (75 nodos, 184 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	1012.3540999999999	1010.4519750000001	2	215,784380s
500	1012.3540999999999	1004.4859499999999	3	289,140887s
1000	1012.3540999999999	980.3352499999999	5	421,830091

Tabela 23 – UFSC - 450m (48 nodos, 89 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	322.1485750000001	307.3986250000001	3	28,588393s
500	322.1485750000001	287.534475	3	29,138282s
1000	322.1485750000001	233.7517750000004	3	25,793708s

Tabela 24 – UFSC - 600m (115 nodos, 229 arcos)

Metros Disponíveis	Densidade Inicial	Melhor Densidade	Número de Mudanças	Tempo de Execução
200	795.5606749999998	789.6622999999997	3	484,99126s
500	795.5606749999998	753.9375249999996	3	475,906789s
1000	795.5606749999998	691.0352249999995	3	469,407217s

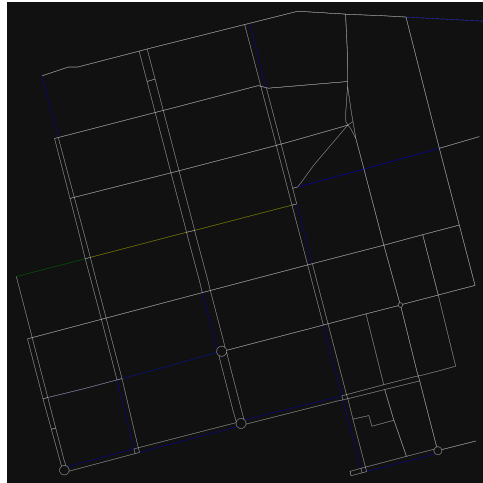


Figura 18 – Densidades padrão



Figura 19 – Densidades otimizadas

Figura 20 – Mapa de chapecó 450m, com 1000m de pista disponíveis para otimização

Podemos ver na figura 20 um exemplo de otimização. Onde as cores do *heatmap* foram mapeadas para as classes definidas em 18. Como podemos ver, as estradas em amarelo (classe D) foram reduzidas para azul (classe B). É necessário ressaltar que certas vias brancas (classe A) também se tornam azuis. Isso se deve a distribuição do congestionamento que está sendo diluído para vias alternativas e é um bom sinal de que não há veículos sendo apagados durante o processo de otimização.

6.4 Considerações Finais

Tendo em mente o objetivo de otimizar o fluxo de uma malha viária de forma geral, e não apenas um único trajeto, este trabalho utilizou de métodos de busca local, e a implementação de um algoritmo para o uso dessa heurística.

O algoritmo apresentado é capaz de fazer alterações de uma malha, para um dado local, de tamanho pequeno melhorando o fluxo desta localidade. Para que o mesmo pudesse ser implementado, foi realizado um levantamento de trabalhos com temas similares utilizando outras heurísticas, bem como a definição do que deveria ser otimizado e o desenvolvimento em si.

Após realizar os experimentos, foi obtido um plano viário ótimo para cada conjunto de localidade, orçamento e tamanho de mapa. Com estes dados, se tornou possível observar o comportamento do algoritmo em diferentes situações.

Os objetivos deste trabalho foram alcançados, demonstrando uma implementação funcional da heurística, bem como uma análise sobre sua eficiência para a resolução de RNDPs.