

Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre

Ana Vieira de Souza, Gustavo Comiotto Schmitz,
Nelson Luiz Joppi Filho, João Pedro Perez Resmer,
Rafael de Santiago

Relatório Técnico INE 001/2023

a Identificação do Relatório

a.1 Título

Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre

a.2 Autores

Ana Vieira de Souza, Gustavo Comiotto Schmitz, Nelson Luiz Joppi Filho, João Pedro Perez Resmer, Rafael de Santiago

a.3 Palavras-chave

Network Design Problem. Otimização Discreta. Métodos Heurísticos.

a.4 Resumo

Congestionamento no tráfego de veículos terrestres afeta a população de grandes cidades. Network Design Problems (NDPs) aplicados ao trânsito de veículos são problemas de otimização que determinam o projeto de trânsito para uma cidade. Eles podem representar vários aspectos complexos das cidades, como tráfego multimodal e o transporte público. Um dos modelos presentes na literatura para lidar com NDPs é o Salman-Alaswad. Nele, alterações no projeto de trânsito de veículos de uma cidade é avaliado considerando inversões na direção de tráfego em pistas das estradas. Para isso, o modelo Salman-Alaswad emprega um método computacional que opera em dois níveis: *(i)* avaliação dos projetos pela metaheurística de algoritmo genético; e *(ii)* definição do equilíbrio do usuário dado por cadeias de Markov. O presente documento relata os resultados obtidos em projeto de pesquisa “Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre” (código no SIGPEX da UFSC 202004500). O projeto teve o objetivo de adaptar o modelo Salman-Alaswad alterando-o para incluir a adição ou remoção de novas pistas e estradas no projeto de trânsito de uma cidade. Para atender esse projeto, realizou-se: levantamento e seleção de modelos e heurísticas sobre NDPs, especificou-se adaptação do modelo e método computacional Salman-Alaswad, foram implementadas as adaptações através de métodos heurísticos de algoritmos genéticos e buscas locais, resultados foram analisados e divulgados em locais de livre acesso. Como resultados obtidos, destacam-se a investigação e métodos heurísticos que são capazes de melhorar o projeto de trânsito dado um modelo de equilíbrio de usuário da via.

b Introdução

Muitas cidades possuem sérios problemas de congestionamento relacionados a suas redes de tráfego. O relatório TomTom Traffic Congestion Ranking 2018 (TOMTOM, 2018) relata que mais de 45% das 403 cidades estudadas apresentaram 25% de nível de congestionamento¹. Em INRIX 2018 Global Traffic Scorecard foi apresentado que o custo por motorista pode variar entre 304,00 e 2.291,00 dólares em cidades norte americanas (INRIX, 2018). Nesse país, a velocidade de tráfego no centro das cidades pode chegar a menos de 14 km/h. No Brasil, em 2018, os níveis de congestionamento atingiram 35% em cidades como Recife, São Paulo e Rio de Janeiro (TOMTOM, 2018). Em Belo Horizonte e São Paulo, motoristas passam 202 e 154 horas em congestionamentos respectivamente (INRIX, 2018).

Neste contexto, alguns pesquisadores da área de otimização discreta lidam com várias famílias de problemas de otimização de tráfego em estradas. Seus trabalhos suportam o desenvolvimento de novas tecnologias para melhorar o trânsito nas cidades. A aplicação desses trabalhos pode criar planejadores e simuladores que auxiliam os tomadores de decisão na alocação de recursos de maneira mais eficiente. Novas formulações matemáticas e métodos podem ser vistos em Bhaskar, Ligett e Schulman (2014), Akhand et al. (2015), Labidi et al. (2016), Owais, Osman e Moussa (2016), Rodríguez-Martín, Salazar-González e Yaman (2016), Bagloee, Sarvi e Patriksson (2017), Nayeem, Islam e Yao (2018), Yıldız, Karahan e Yaman (2018), Islam et al. (2019). Kian e Kargar (2016), Paraskevopoulos, Gürel e Bektaş (2016), Salman e Alaswad (2018) lidam com problemas de congestionamento. Lin et al. (2017), Fontaine e Minner (2018), Kaya e Ozkok (2018), Kolak, Feyzioglu e Noyan (2018) incluíram sustentabilidade em seus modelos e métodos. Metadados para adicionar ou editar estradas são tratados em Chen et al. (2015), Wang, Liu e Szeto (2015), Fontaine e Minner (2017), Fukunaga (2017). Associação de tráfego em Jafari, Pandey e Boyles (2017), Wang, Peeta e He (2019).

Network Design Problems (NDP) são problemas de otimização discreta para apoiar a tomada de decisão no projeto de redes de tráfego considerando vários fatores. Formaliza-se um problema NDP usando os modelos de Leblanc (1975) e Poorzahedy e Turnquist (1982). Considere uma rede/gráfico $G = (V, A)$ que deve ter seu projeto alterado. V é o conjunto de nodos e $A = A_1 \cup A_2$ é o conjunto de arcos. A_1 é o conjunto de arcos existentes que não serão modificados. A_2 é o conjunto de arcos a serem considerados. Frequentemente, o NDP é resolvido em dois submodelos. O primeiro representa é concebido para definir os melhores planos de mudança na rede, chamado aqui de NDPm. O segundo é sobre o equilíbrio do usuário que mapeia o fluxo na rede, chamado aqui de NEPm.

O modelo NEPm é apresentado na Equação (1) usando a versão de Leblanc (1975)

¹ A medida de nível de congestionamento é obtida pela comparação entre condições com livre tráfego e o comportamento médio anual para cada segmento de pista.

no lugar de [Poorzahedy e Turnquist \(1982\)](#) por simplicidade. Assuma que x_a é o fluxo total que passa pelo arco, e x_{av} é o fluxo que passa pelo arco a para o nodo $v \in V$. $d_{u,v}$ denota a demanda do par origem-destino (u, v) , considerando o par de nodos $u, v \in V$. Os valores $y_a = 1$ se o projeto a é selecionado, $y_a = 0$ caso contrário. O valor $h_a(x_a, y_a)$ denota o custo médio de viagem sobre o arco a . Ele depende do fluxo, x_a e y_a .

O modelo NDPm é apresentado na Equação (2). Considere que c_a é o custo de implementar o projeto (arco) a , e B é o orçamento total disponível para o planejamento. x_a^* é a solução dos valores obtidos no modelo NEPm, ou seja, o fluxo que passa pelo arco a , previamente estabelecido pela solução obtida para um modelo NEPm.

$$\begin{aligned}
& \mathbf{min} \sum_{a \in A_2} x_a h_a(x_a, y_a) \\
& \mathbf{s.t.} \\
& \sum_{a \in \alpha(u)} x_{av} - \sum_{a \in \beta(u)} x_{av} = d_{u,v} \\
& \forall u, v \in V, u \neq v \\
& \sum_{u \in V} x_a u - x_a = 0 \quad \forall a \in A \\
& x_{av} \geq 0 \quad \forall v \in V, \forall a \in A \\
& x_a \geq 0 \quad \forall a \in A \\
& y_a \in \{0, 1\} \quad \forall a \in A_2.
\end{aligned} \tag{1}$$

$$\begin{aligned}
& \mathbf{min} \sum_{a \in A_2} x_a^* h_a(x_a^*, y_a) \\
& \mathbf{s.t.} \\
& \sum_{a \in A_2} c_a y_a \leq B \\
& y_a \in \{0, 1\} \quad \forall a \in A_2.
\end{aligned} \tag{2}$$

Problemas NDP aplicados ao trânsito de veículos terrestres são problemas de otimização discreta que determinam o trânsito de uma cidade ([NAYEEM; ISLAM; YAO, 2018](#)). Eles podem representar muitos aspectos da realidade complexa de uma cidade, incluindo tráfego multimodal, transporte público e poluição. Esses problemas são conhecidos pela alta demanda de recursos computacionais (principalmente tempo computacional) ([ISRAELI; CEDER, 1995](#)), então há diversos esforços na literatura científica que reportam modls não exatos para problemas NDP.

Alguns métodos apresentados na literatura consideram NDP como de programação em dois níveis (bilevel programming) ([GAO; WU; SUN, 2005](#); [ZHANG; GAO, 2009](#); [FARVARESH; SEPEHRI, 2011](#); [FONTAINE; MINNER, 2014](#); [YU et al., 2015](#); [SUN, 2016](#);

[KOLAK; FEYZIOĞLU; NOYAN, 2018](#)). [Yang e Bell \(1998\)](#) propuseram um modelo de dois níveis para o NDP considerando os seguintes componentes: atividade econômica, capacidade do sistema de transporte, fluxo de tráfego e sistema gerenciamento. Um modelo de programação em dois níveis para o NDP pode usar o modelo de jogo líder-seguidor, no qual o sistema de gerenciamento é o líder e os usuários da rede são os seguidores. O nível-superior representa as tomadas de decisão de investimento do transporte para maximizar os benefícios sociais. Nesse nível, é assumido que há um equilíbrio único de padrão de fluxo obtido pelo nível-inferior para qualquer variável do nível-superior.

O equilíbrio do usuário caracteriza o comportamento de escolha de rota na rede. É importante definir os padrões de fluxo, e quais partes da rede necessitam de maior atenção ([LUATHEP et al., 2011](#)). Geralmente, uma matriz de origem-destino é usada para mapear a demanda de uma origem até um local. O princípio de Wardrop ([WARDROP, 1952](#)) é um modelo de equilíbrio comumente utilizado. No Wardrop, o tempo de viagem de todos os caminhos usados conectando origens e destinos é igual ou inferior a qualquer caminho não usado ([WANG; LO, 2010](#)). [Salman e Alaswad \(2018\)](#) propuseram uma cadeia de Markov no lugar do tempo total de viagem para o equilíbrio do usuário devido ao tempo de viagem esperado entre dois pontos da rede não depende dos pontos iniciais ou finais. Então eles sugeriram o uso da densidade de veículos nas estradas (veículos por quilômetros por número de pistas)

Em [Salman e Alaswad \(2018\)](#), é apresentado um modelo para reduzir o congestionamento utilizando como método computacional uma composição entre a metaheurística de algoritmo genético e cadeias de Markov. O algoritmo genético encontra quais pistas de estradas existentes deve-se reverter o sentido para aliviar tráfego de veículos e, consequentemente, diminuir os congestionamentos na malha (nível-superior). Cada indivíduo do algoritmo genético é representado por um conjunto de pistas com a devida atribuição de quais pistas tiveram seu sentido alterado. O tráfego de veículos é atribuído por uma cadeia de Markov usando a densidade de veículos em cada pista (nível-inferior). Logo, cada indivíduo é avaliado de acordo com o valor resultante na cadeia de Markov para cada pista. A função de avaliação para cada indivíduo utiliza a densidade da via mais ocupada, ou seja, de maior densidade de veículos. Por utilizar cadeias de Markov como modelo de equilíbrio de usuário, o trabalho de [Salman e Alaswad \(2018\)](#) não foi avaliado utilizando matrizes de demanda origem e destino.

O objetivo do projeto foi adaptar o método proposto em [Salman e Alaswad \(2018\)](#), considerando algum outro modelo de equilíbrio que utiliza matriz de demanda. Para atingir o objetivo: o método de [Salman e Alaswad \(2018\)](#) e outros da literatura foram levantados e estudados; foram mapeados modelos de equilíbrio presentes na literatura; o método proposto em [Salman e Alaswad \(2018\)](#) foi estudado e adaptado; instâncias de teste foram preparadas com base em regiões reais; experimentos foram realizados. Além

disso, outros métodos heurísticos de busca local foram especificados e desenvolvidos. Para isso, o projeto contou com duas bolsas de iniciação científica, dois trabalhos de conclusão de curso, e um estudante de mestrado (que foi desligado do programa). Com as pesquisas desenvolvidas, obteve-se um algoritmo genético e buscas locais.

c Resultados alcançados

Como principais resultados atingidos durante a execução do projeto destacam-se os métodos heurísticos desenvolvidos para o problema de pesquisa. A seguir, são apresentados estes resultados trazendo relato, especificações, experimentações e análises.

c.1 Algoritmo Genético

O algoritmo genético recebe como dados de entrada um conjunto de estradas e dados relacionados, tais como *travel time* (tempo de viagem) e velocidade máxima de cada via. Através disso, o algoritmo genético inicia sua execução em busca do projeto de trânsito com a melhor avaliação (*fitness* inspirado no modelo de [Salman e Alaswad \(2018\)](#)). Os resultados aqui relatados estão formalizados nos trabalhos de ([FILHO, 2022](#); [SCHMITZ, 2022](#)), nos trazem todos os conceitos relacionados e detalhes da pesquisa.

As subseções a seguir apresentam os aspectos considerados durante o projeto do algoritmo genético.

c.1.1 Representação de um indivíduo

Cada indivíduo (cromossomo) representa um projeto de trânsito da cidade, inspirado nas vias existentes da região a ser analisada. Cada segmento de uma via é considerado um gene, ou seja, uma parte do cromossomo. Cada gene possui os seguintes atributos: o sentido da via, a quantidade de faixas já existentes e dados de localização. Como o algoritmo genético considera a possibilidade de criação de novas vias, gene novos podem ser adicionados a um cromossomo. Durante a execução do algoritmo, esses atributos são manipulados pelos operadores genéticos afim de aumentar o *fitness* do modelo.

c.1.2 Operadores genéticos

O modelo proposto neste trabalho utiliza os operadores genéticos, sendo eles: Seleção, Crossover e Mutação.

Seleção

Neste trabalho, foram implementadas três técnicas de seleção diferentes: “Torneio”, “Roleta” e “Aleatória”. A técnica de “Torneio” consistiu em selecionar aleatoriamente 5

dentre a geração atual de indivíduos, após esse passo, é selecionado para a próxima etapa o indivíduo que apresentar o melhor *fitness* entre os 5 selecionados previamente.

A técnica de “Roleta” consistiu em, primeiramente, somar todos os valores de *fitness* de todos os indivíduos. Tendo esse valor em mãos, é escolhido um valor aleatório entre 0 e o total da soma feita anteriormente. Então é realizada uma iteração entre todos os indivíduos da geração atual, incrementando o valor de uma variável com seu valor de *fitness*, e verificando se o valor dessa variável é maior que o valor aleatório escolhido anteriormente. Caso a condição descrita anteriormente seja verdadeira, o indivíduo é escolhido, caso contrário, ignora-se o indivíduo e a iteração é continuada.

Por último, a técnica “Aleatória” escolhe um indivíduo dentre todos os presentes na atual geração.

Crossover

O método de *crossover* desenvolvido para este trabalho consiste em analisar cada alteração, em relação ao plano viário original, dos indivíduos pais. Cada alteração dos pais tem 50% de chance de integrar indivíduo filho.

Mutação

A técnica de Mutação apresentada nesse trabalho consiste em, com base na taxa de mutação definida no início da execução do algoritmo, modificar aleatoriamente apenas as estradas pré-existentes no plano viário original. As modificações variam entre: inserção de uma nova faixa num trecho de estrada ou inserção de uma nova faixa em todos os trechos de uma estrada.

Fitness

O *fitness* no modelo proposto por este trabalho utiliza, diferentemente do modelo de [Salman e Alaswad \(2018\)](#) que utiliza a densidade máxima de veículos numa estrada, o inverso da densidade média de veículos de todas as estradas do plano viário. A fórmula abaixo descreve de maneira detalhada a obtenção deste valor:

$$DM = \frac{DC \cdot 100}{DS},$$

onde *DM* representa a densidade média, *DC* representa o número de estradas do plano e *DS* o somatório das densidades de veículos de todas as estradas. O indivíduo que obtiver o maior resultado ao final deste cálculo será considerado o mais apto.

População inicial

A população inicial utilizada consiste num conjunto de *N* indivíduos, ou seja, *N* possíveis projetos de otimização da malha viária de uma determinada região.

Para definir a população inicial, os *N* indivíduos deverão conter mudanças aleatori-

amente selecionadas a partir do plano atual da região de entrada do algoritmo (bairro, cidade, estado, ...).

c.1.3 Experimentos e Análises

Para a execução dos experimentos foi utilizado um computador com as seguintes características: (i) 32 GB de memória RAM e processador “Intel core i7 11th”.

Os experimentos efetuados neste trabalho são basicamente execuções do modelo com variações dos parâmetros existentes no algoritmo genético, com objetivo de detectar em quais gerações acontece convergência entre eles, considerando o tamanho da instância de entrada (número de trechos da via).

Os dados de entrada que foram obtidos a partir da ferramenta OpenStreetMap, as informações capturadas foram: a malha viária da região, com a velocidade máxima de cada via, a quantidade de pistas por via, direção de cada pista. Com esses dados, representa-se o projeto de trânsito atual da cidade. A ideia é a de que o plano da cidade seja utilizado como base para os indivíduos gerados na população inicial e nas demais etapas do algoritmo genético. Para a execução dos testes, foi selecionado como plano viário base o município de Borá, considerado o menor município do estado de São Paulo.

Dois parâmetros foram fixados durante todo o experimento, são eles: 50 gerações e 10km de orçamento (*budget*) disponível para alteração na malha viária. A lista abaixo, ilustra os demais parâmetros que serão alterados durante a execução do algoritmo genético e seus valores, respectivamente:

- População inicial: 10, 15 e 20 Indivíduos.
- Taxa de mutação: 1% e 10%.
- Taxa de indivíduos elitistas por geração: 10% e 20%.
- Técnicas de seleção: Torneio, Roleta e Aleatória.
- Taxa de filhos gerados por geração: 25% e 50%.

Tomando tais parâmetros como base, o algoritmo genético foi executado 10 vezes com cada uma das 72 combinações existentes.

Análise de Resultados

Com base nos dados obtidos nos experimentos, foi efetuado um cálculo de média, onde soma-se os valores de média de *fitness* por geração em cada execução, conforme

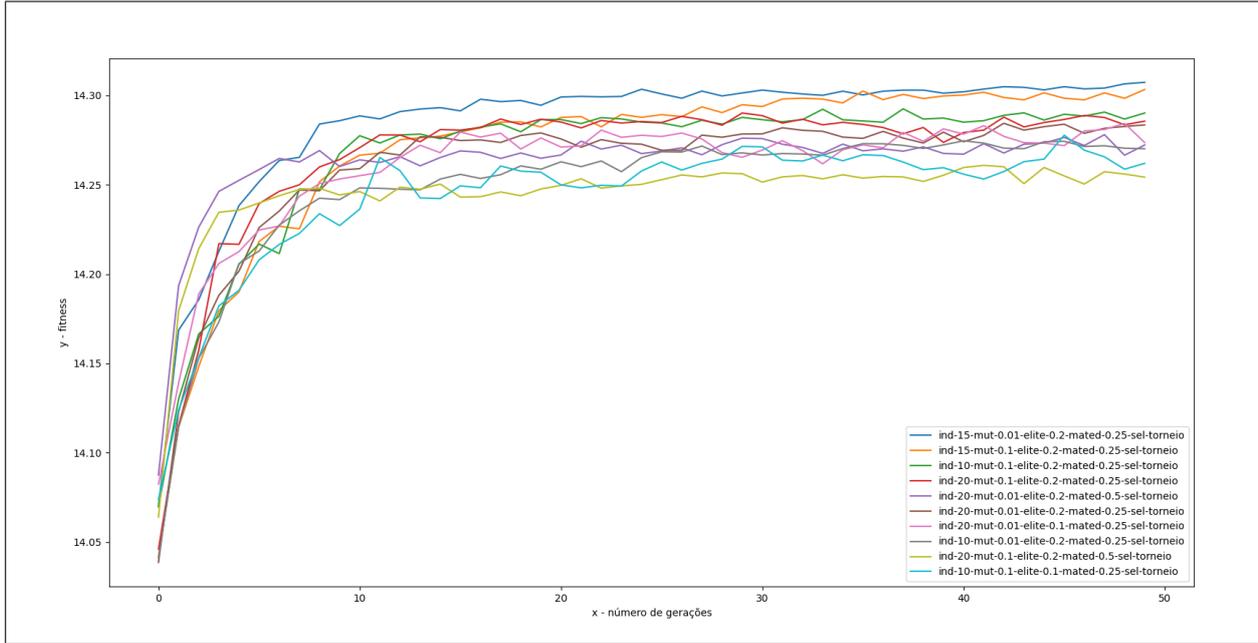


Figura 1 – Os 10 melhores conjuntos de parâmetros com base no *fitness* médio por geração.

ilustrado abaixo:

$$FMCP = \frac{\frac{SMFG_1}{NE} + \frac{SMFG_2}{NE} + \dots + \frac{SMFG_G}{NE}}{G} \quad (3)$$

no qual, $FMCP$ indica o *fitness* médio por conjunto de parâmetros. O $SMFG_i$ representa o somatório das médias de *fitness* da i -ésima geração em todas as execuções do algoritmo com determinado conjunto de parâmetros. NE representa o número de execuções do algoritmo, já G representa o número de gerações pré-definidas. Essa equação é utilizada nas Tabela [c.1.3](#).

Dado este cenário, obteve-se como resultado as 10 melhores médias de *fitness* por conjunto de parâmetros, conforme gráfico da Figura 1. Para o gráfico, é considerado “fitness” a equação abaixo:

$$\text{Fitness por geração} = \frac{SMFG_i}{NE}. \quad (4)$$

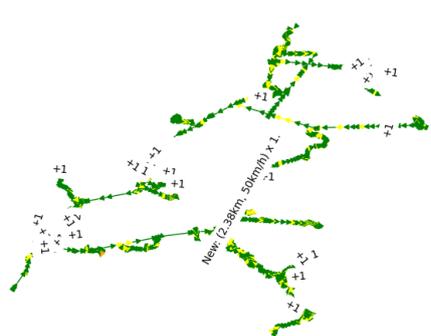
no qual i corresponde a i -ésima geração.

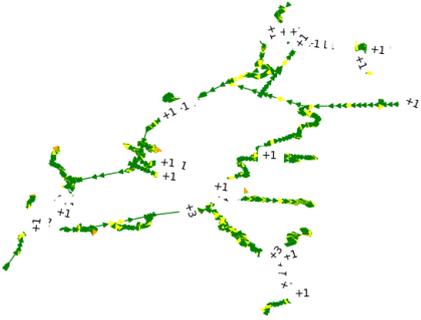
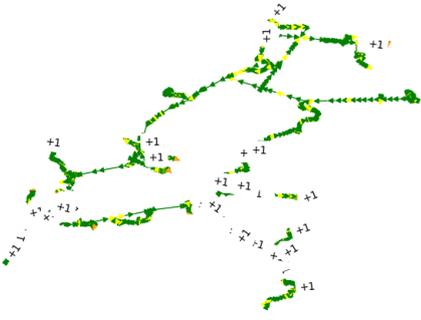
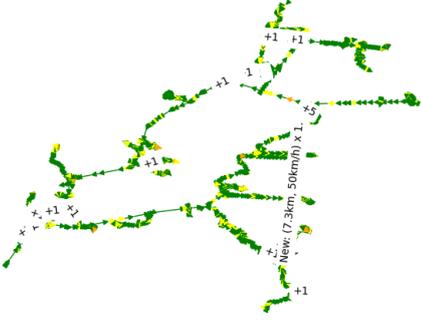
Analisando o gráfico da Figura 1, pode-se concluir que o algoritmo tende a apresentar os melhores indivíduos, em média, quando a técnica de seleção utilizada é a de “Torneio”, visto que todos os registros presentes no gráfico utilizaram tal técnica. Há outros dois pontos a serem observados, o primeiro refere-se as taxas de indivíduos elitistas e de filhos gerados através da etapa de *crossover*. Neste primeiro ponto, percebe-se que tais parâmetros são importantes na obtenção do resultados, mas de maneiras opostas. A

taxa de elitismo se comportou de maneira progressiva, ou seja, quanto maior, melhor o resultado. Com relação ao parâmetro taxa de filhos gerados através da etapa de *crossover*, observa-se que quanto maior o número de filhos, menor a qualidade esperada por geração. Já o segundo ponto, refere-se a taxa de mutação e o tamanho da população, tais fatores não se mostraram relevantes, pois seus resultados são equilibrados. Portanto, após os experimentos verificou-se que, em 50 gerações e com 10 execuções do algoritmo com cada conjunto de parâmetros, os melhores resultados obtidos possuíam os seguintes parâmetros:

- População Inicial: 15 indivíduos.
- Taxa de mutação: 1%.
- Taxa de indivíduos elitistas por geração: 20%.
- Taxa de filhos gerados por geração: 25%.
- Técnica de seleção: Torneio.

Para encontrar os melhores indivíduos por conjunto de parâmetros individualmente, foi feita uma análise pelo *fitness* do melhor indivíduo de cada execução. Dado esse cenário, gerou-se uma classificação com os melhores indivíduos, onde também é ilustrado o layout do plano de trânsito de cada um dos classificados.

Colocação	Conj. parâmetros	Fitness	Layout
1º	Indivíduos - 15 Tx. Mutação - 10% Tx. Elitismo - 20% Tx. Filhos gerados - 50% Téc. Seleção - Torneio	14.4661	

Colocação	Conj. parâmetros	Fitness	Layout
2º	<p>Indivíduos - 20 Tx. Mutação - 10% Tx. Elitismo - 20% Tx. Filhos gerados - 50% Téc. Seleção - Torneio</p>	14.4102	 <p>A network layout visualization showing a complex, interconnected structure of nodes and edges. The nodes are represented by small green and yellow markers, and the edges are thin lines connecting them. The overall shape is somewhat irregular and spread out.</p>
3º	<p>Indivíduos - 15 Tx. Mutação - 10% Tx. Elitismo - 10% Tx. Filhos gerados - 25% Téc. Seleção - Torneio</p>	14.4015	 <p>A network layout visualization showing a complex, interconnected structure of nodes and edges. The nodes are represented by small green and yellow markers, and the edges are thin lines connecting them. The overall shape is somewhat irregular and spread out.</p>
4º	<p>Indivíduos - 10 Tx. Mutação - 10% Tx. Elitismo - 10% Tx. Filhos gerados - 50% Téc. Seleção - Torneio</p>	14.4005	 <p>A network layout visualization showing a complex, interconnected structure of nodes and edges. The nodes are represented by small green and yellow markers, and the edges are thin lines connecting them. The overall shape is somewhat irregular and spread out. A label 'New (7.3km, 50km/h) X 1.' is visible near the bottom right of the network.</p>

Colocação	Conj. parâmetros	Fitness	Layout
5º	<p>Indivíduos - 15</p> <p>Tx. Mutação - 1%</p> <p>Tx. Elitismo - 10%</p> <p>Tx. Filhos gerados - 50%</p> <p>Téc. Seleção - Torneio</p>	14.3926	
6º	<p>Indivíduos - 20</p> <p>Tx. Mutação - 10%</p> <p>Tx. Elitismo - 10%</p> <p>Tx. Filhos gerados - 25%</p> <p>Téc. Seleção - Torneio</p>	14.3863	
7º	<p>Indivíduos - 15</p> <p>Tx. Mutação - 1%</p> <p>Tx. Elitismo - 20%</p> <p>Tx. Filhos gerados - 50%</p> <p>Téc. Seleção - Torneio</p>	14.3862	

Colocação	Conj. parâmetros	Fitness	Layout
8º	<p>Indivíduos - 20</p> <p>Tx. Mutação - 10%</p> <p>Tx. Elitismo - 20%</p> <p>Tx. Filhos gerados - 25%</p> <p>Téc. Seleção - Torneio</p>	14.3794	
9º	<p>Indivíduos - 20</p> <p>Tx. Mutação - 1%</p> <p>Tx. Elitismo - 10%</p> <p>Tx. Filhos gerados - 25%</p> <p>Téc. Seleção - Torneio</p>	14.3745	
10º	<p>Indivíduos - 15</p> <p>Tx. Mutação - 10%</p> <p>Tx. Elitismo - 10%</p> <p>Tx. Filhos gerados - 50%</p> <p>Téc. Seleção - Torneio</p>	14.3745	

Tabela 1 – Os 10 melhores indivíduos encontrados por conjunto de parâmetros

Ao observarmos a tabela [c.1.3](#), é possível verificar as diferentes alterações efetuadas pelo algoritmo genético no plano viário da cidade de Borá-SP. As indicações que contenham um “+” acompanhado de um determinado valor N , indicam que houve um acréscimo

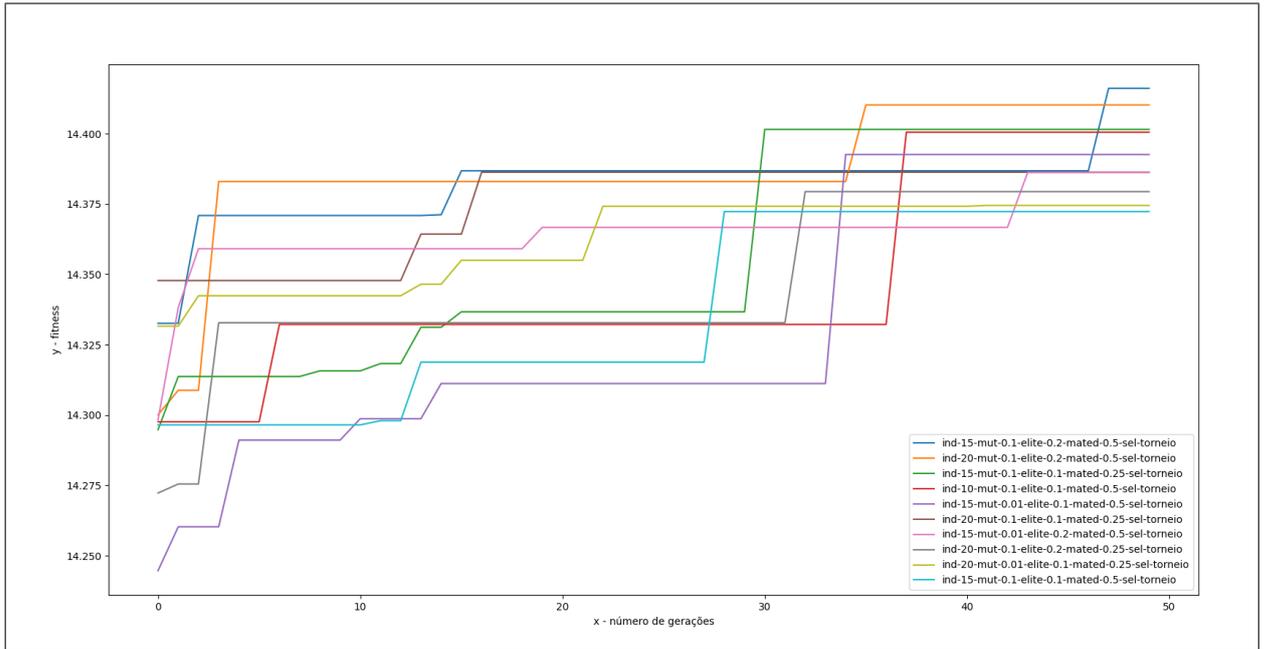


Figura 2 – Os 10 melhores indivíduos com base no *fitness* por conjunto de parâmetros.

de N faixas de trânsito naquele trecho de via. Já a outra indicação, sugere que houve a inserção de uma nova estrada no plano viário preexistente, de tamanho, velocidade máxima permitida e número de faixas também explicitados. As cores nos mapas indicam a intensidade do trânsito nas vias, quanto mais quente for a cor, mais intenso o trânsito. A Figura 2 mostra a evolução do *fitness* dos indivíduos classificados no decorrer das 50 gerações.

Com base nas informações do gráfico da Figura 2, percebe-se que novamente a técnica de seleção que apresentou os melhores resultados é a de “Torneiro”. Portanto, pode-se concluir que o parâmetro mais influente nos resultados foi a técnica de seleção empregada, já os demais parâmetros demonstram-se equilibrados durante os experimentos. Outro ponto a ser destacado é o fato de nenhum dos registros do gráfico apresentar declínio durante o decorrer das gerações, isso se deu pelo fato do algoritmo possuir um caráter elitista. Isso garante que ao final da execução, teremos sempre o melhor indivíduo dentre todos os gerados.

Após a execução dos experimentos com base no plano viário apresentado na Figura 3, o indivíduo que apresentou o melhor *fitness* com base nos parâmetros, fixos e variáveis, foi o da Figura 4 ilustrada abaixo

Comparado com o plano viário inicial do experimento, o indivíduo selecionado apresentou cerca de 27 faixas novas em trechos de estradas preexistentes e a inserção de uma nova pista de 2,38KM no plano viário da cidade analisada. Além disso, possui um *fitness* no valor de 14.4161, sendo levemente superior aos demais indivíduos. O conjunto

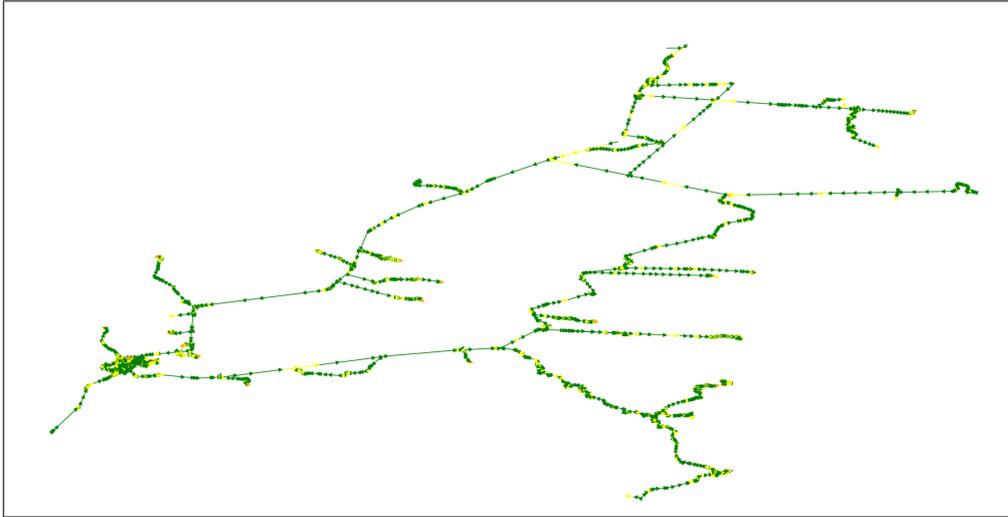


Figura 3 – Plano inicial.

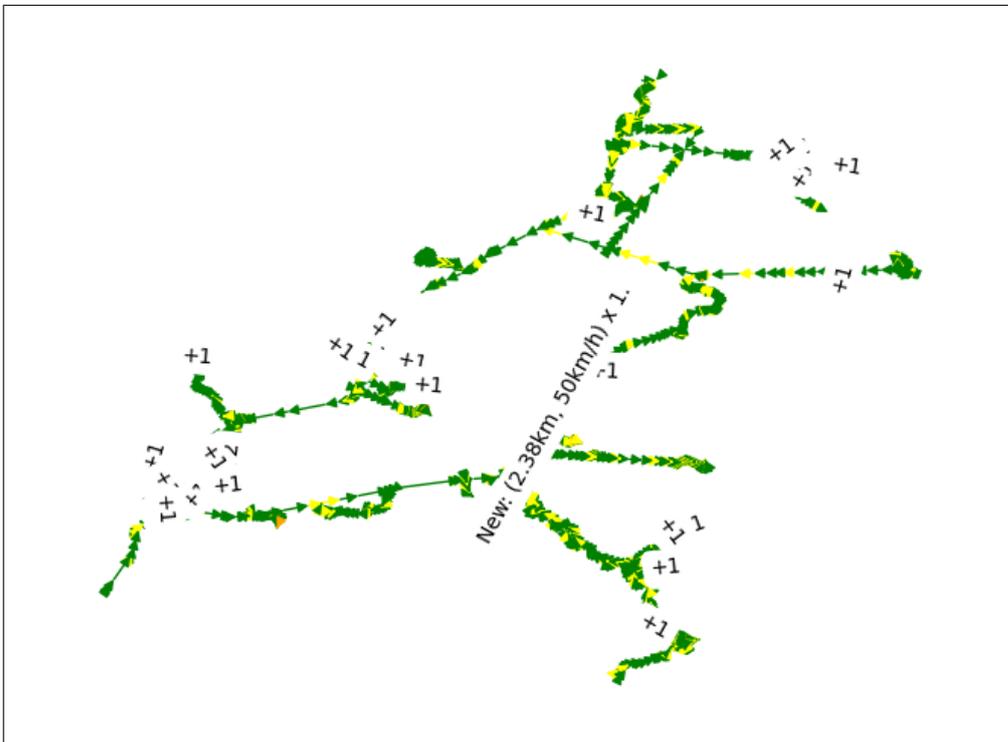


Figura 4 – O melhor indivíduo encontrado durante os experimentos.

de parâmetros utilizado para encontrar este indivíduo possui:

- Budget: 10 quilômetros.
- Número de gerações: 50 gerações.
- População Inicial: 15 indivíduos.
- Taxa de mutação: 10%.
- Taxa de indivíduos elitistas por geração: 20%.
- Taxa de filhos gerados por geração: 50%.
- Técnica de seleção: Torneio.

Algoritmo Genético Adaptado

Com base no algoritmo genético explicado na seção anterior, o trabalho de iniciação científica (FILHO, 2022) amadureceu os experimentos incluindo um novo modelo de equilíbrio de usuário.

O modelo de equilíbrio refere-se ao modo como se redistribui os carros na malha viária a partir da distribuição inicial de carros nas vias. Na prática, se itera sobre todas as vias, transferindo uma porcentagem de carros de cada uma para vias posteriores, simulando o fluxo de carros. Esse cálculo de transferência é feito exclusivamente com o tempo de viagem da via iterada, quanto maior este for, mais carros permanecem na pista; consequentemente, quanto menor este for, menos carros permanecem. O problema então é escolher a melhor função para fazer este mapeamento de um intervalo que vai de 0 a infinito, para um intervalo de 0 a 1, e os métodos cogitados foram os seguintes:

1. Divisão pelo menor valor de tempo de viagem; A ideia aqui é normalizar nosso tempo de viagem a partir do menor tempo de viagem dentre todas as vias da malha viária analisada, com a seguinte fórmula:

$$permacencia = 1 - menor_tempo_de_viagem/tempo_de_viagem \quad (5)$$

O problema desta fórmula é uma taxa de permanência excessiva nas vias, fazendo com que haja pouco fluxo de carros. A imagem pode ser observada em 5

2. Divisão pelo maior valor de tempo de viagem; A ideia é parecida com a anterior, mas agora ao invés de normalizarmos a partir do menor valor, faremos a partir do maior, com a seguinte fórmula:

$$permacencia = tempo_de_viagem/maior_tempo_de_viagem \quad (6)$$

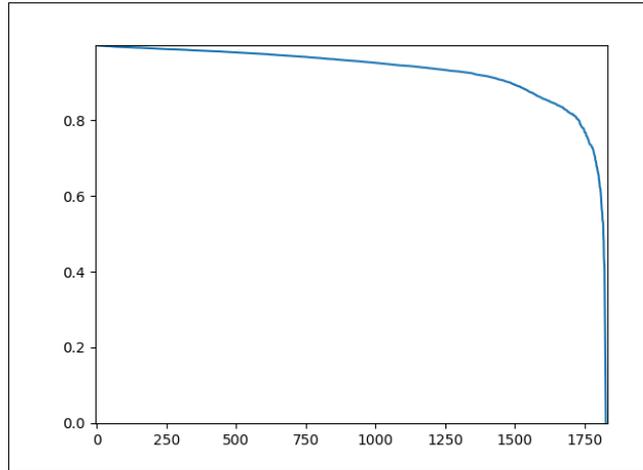


Figura 5 – Modelo de equilíbrio normalizando com o menor valor de tempo de viagem.

Agora, temos uma taxa de permanência média baixa, o que é melhor que uma muita alta para simular o fluxo de carros, porém assim como no método anterior, há uma dependência por valores extremos de tempo de viagem, o que gera uma imprevisibilidade do resultado final. A imagem pode ser observada em 6

3. Usando a fórmula que define o ponto que possui 50% de permanência de carros:

$$permanencia = 1 - 2^{-tempo_de_viagem/valor_50_porcento} \quad (7)$$

Aqui não utilizamos valores extremos como o maior ou menor tempo de viagem, e sim pegamos como um valor intermediário, que é muito mais constante, para possuir uma permanência de 50%, caso este estiver muito próximo ao menor tempo de viagem, teremos um gráfico muito parecido com o da primeira Figura 5, caso estiver muito próximo ao maior tempo de viagem, ficará parecido com gráfico seguinte 6. Ao invés disso, decidimos utilizar como ponto de permanência igual a 50%, o ponto que divide a área do gráfico de tempo de viagem que pode ser visto na Figura 8 em dois, o seu centro de massa, e o gráfico com esta fórmula aplicada pode ser visto na Figura 7.

Optou-se usar a terceira opção pois não apresenta problemas com *outliers* como as duas primeiras apresentam, e pelo fato de podermos customizar o valor de tempo de viagem que queremos que possua vazão de 50%.

Experimentos e Análise dos resultados

Foi utilizado o projeto no bairro de São Pedro, em Juiz de Fora, Minas Gerais, Brasil. E as alterações feitas podem ser visualizadas em 9. O indivíduo é o melhor encontrado em 10 execuções do algoritmo, e a semente para replicar o resultado é 0,941343013515275. A aptidão inicial (sem novas pistas ou estradas) para o local foi de 141,2514 e a aptidão final

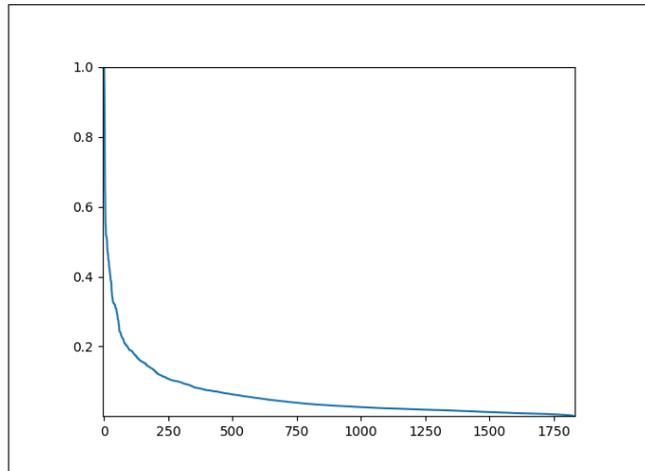


Figura 6 – Modelo de equilíbrio normalizando com o maior valor de tempo de viagem.

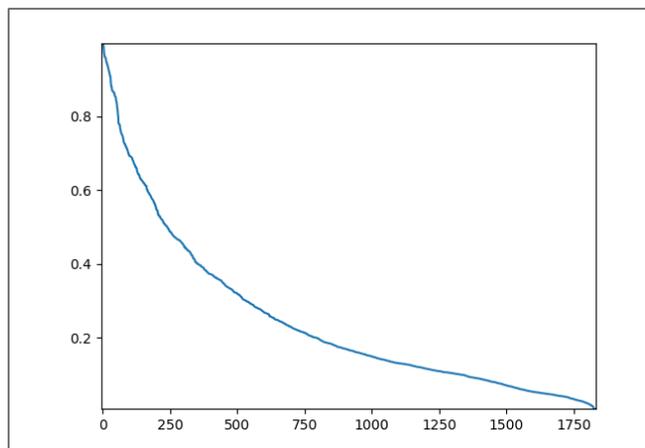


Figura 7 – Modelo de equilíbrio normalizando com a fórmula de ponto de 50% customizável.

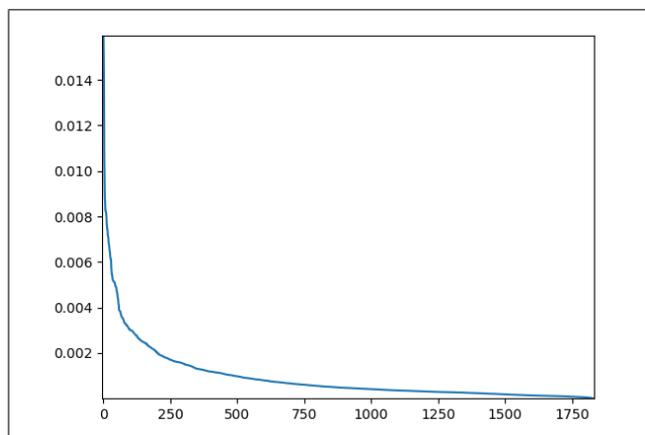


Figura 8 – Distribuição em ordem, dos tempos de viagem das vias.

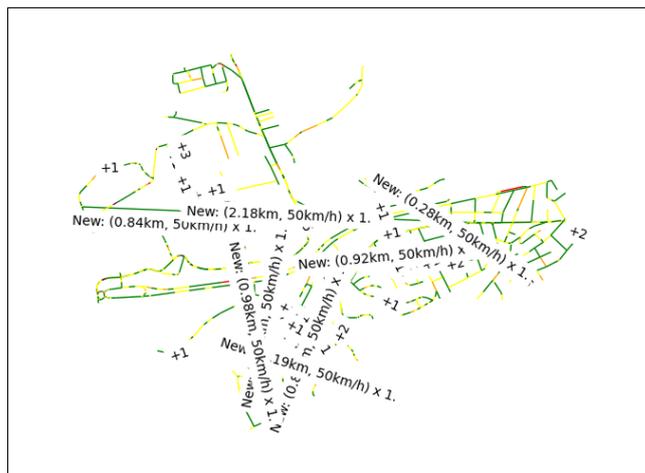


Figura 9 – Algoritmo genético aplicado no bairro de São Pedro

foi de 143,1987, uma melhora de 1,37%. Entre as mudanças ocorridas, que foram a adição de faixas em 119 localidades, e a criação de 8 novas vias, podemos citar: Adição de 1 faixa na rua 'Avenida Presidente C...ta e Silva', entre nós '1273' e '1145'; Adição de 2 pistas na 'Rua José Lourenço Kelmer', entre os nós '1461' e '78'; Adição de 1 faixa na rua 'Rua Octavio Malvaccini', entre os nós '36' e '943'; Criação de um novo caminho entre os nós '23' e '445'; E mais mudanças, que podem ser vistas na saída do programa.

Grande parte dos esforços deste trabalho envolveu achar o melhor conjunto de parâmetros para nosso trabalho. Por exemplo, na etapa de seleção, existem diversos métodos diferentes, como roleta, torneio, e até uma seleção aleatória. Como não tínhamos conhecimento de quais parâmetros funcionariam melhor para nosso problema, dedicamos um tempo exclusivamente a isto, com o objetivo de achar o conjunto de parâmetros que geraria os melhores indivíduos constantemente.

Como inicialmente não sabíamos quais técnicas eram melhores, rodamos nosso algoritmo em todas as 486 combinações de parâmetros possíveis, 10 vezes para cada uma, totalizando 4860 execuções. Foram 6 parâmetros diferentes, cada um com suas técnicas próprias, sendo eles: número total de indivíduos, que pode ser 10, 20 ou 40; taxa de mutação, que pode ser 0%, 5% ou 20%; percentual de indivíduos elitistas, que pode ser 0%, 10% ou 30%; percentual de indivíduos provenientes da seleção (também chamados de indivíduos acasalados), que pode ser 20%, 50% ou 70%; técnica de seleção, que pode ser "torneio", "roleta" ou "aleatório"; por fim, a técnica de inserção de indivíduos aleatórios, que pode ser "nova população" ou "população atual". Outra coisa importante é que usamos o mapa da cidade de Borá, em São Paulo, Brasil, para nossos testes, usando a ferramenta OpenStreetMap. Por fim, ao final de cada uma das 4860 execuções, salvamos o melhor fitness encontrado na população, para cada geração. Dessa forma, salvando esse valor por geração, conseguimos visualizar a evolução de cada execução ao longo do tempo.

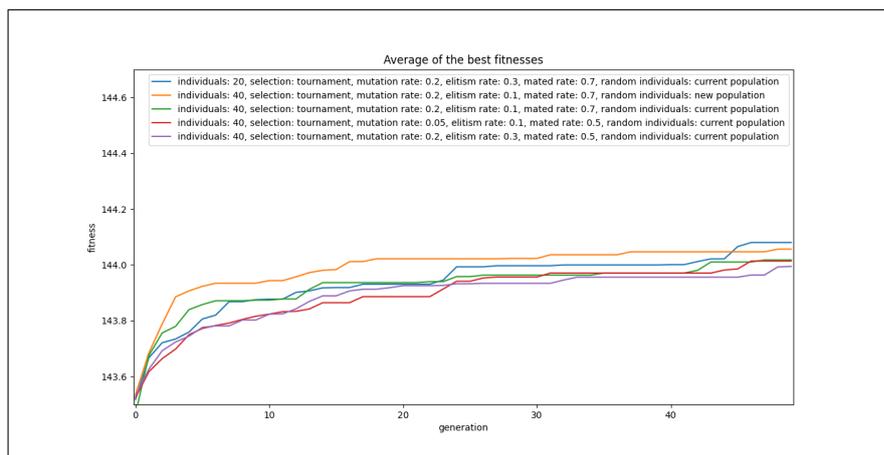


Figura 10 – Conjunto de parâmetros que apresentaram melhor desempenho, dentre 10 execuções

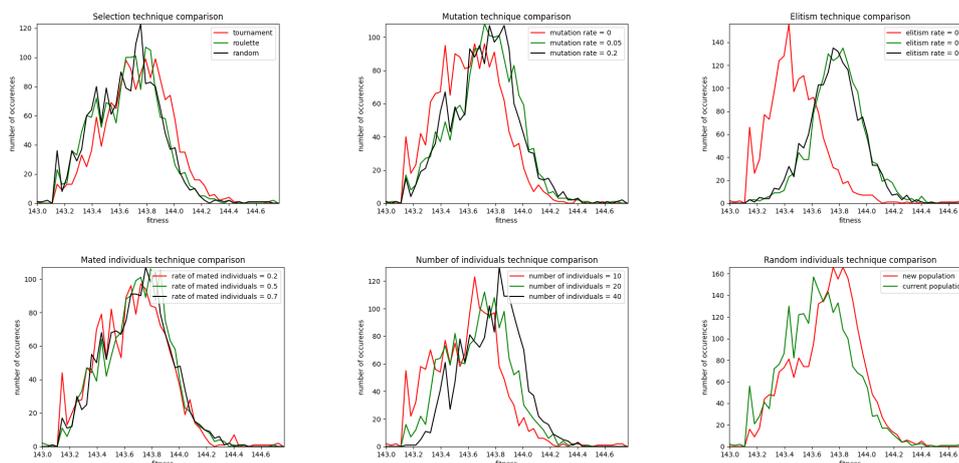


Figura 11 – Comparação entre as técnicas de cada parâmetro

Para discutir os resultados obtidos, usamos duas figuras diferentes. A Figura 10 mostra as melhores combinações de parâmetros encontrados entre os 486 testados, para isso executamos cada combinação 10 vezes, e calculamos a média do melhor fitness encontrado em cada geração das 10 execuções, depois repetimos esse processo para cada combinação de parâmetros. A Figura 11 compara as técnicas de cada um dos parâmetros utilizados, onde no eixo x temos o valor de fitness, e no eixo y temos o número de ocorrências, entre o total de 4860, que estão em torno do respectivo valor x. Diferentes curvas possuem cores que representam as diferentes técnicas comparadas, indicadas pela legenda do gráfico. Na próxima seção analisamos os gráficos apresentados.

Uma análise para encontrar o melhor conjunto de parâmetros para o algoritmo foi realizada. Mas imediatamente encontramos um problema, como sabemos o que é melhor? Pensamos em duas abordagens diferentes para responder a isso: A primeira, mostrada na Figura 10, visa comparar combinações específicas de **parâmetros** entre si, na qual

Técnica de seleção	torneio
Taxa de mutação	0.05 ou 0.2
Taxa de elitismo	0.1 ou 0.3
Indivíduos acasalados	0.2 ou 0.5 ou 0.7
Número de indivíduos	40
Indivíduos aleatórios	nova população

Tabela 2 – Melhores técnicas para cada parâmetro

vemos que no final do 50 gerações, a combinação representada em azul teve o melhor desempenho, seguida de laranja. No entanto, um problema com essa abordagem é que ela pode variar muito a cada nova simulação que realizamos, pode ser que outros conjuntos de parâmetros fiquem entre os 5 melhores na próxima simulação. Observamos então que precisamos de uma análise que tenha resultados mais estáveis, como uma análise individual das **técnicas** utilizadas em cada um dos parâmetros. Mostramos essa comparação na Figura 11 e comentamos o que foi observado. A partir das técnicas de seleção, número de indivíduos e números de indivíduos aleatórios, vemos um leve deslocamento para a direita em relação aos demais em “torneio”, “40 indivíduos” e “nova população”, o que coloca como ligeiramente superiores às outras técnicas desses parâmetros. Para taxa de elitismo e taxa de mutação, observamos um cenário um pouco diferente, as duas melhores técnicas são muito semelhantes, mas a terceira técnica em cada uma é significativamente pior, principalmente no parâmetro elitismo, onde uma taxa de 0% é claramente a pior.

Por fim, no parâmetro “taxa de indivíduos acasalados”, não há diferença significativa entre as técnicas. Com isso vimos que as melhores técnicas para cada parâmetro são aquelas encontradas na Tabela 2, e escolheremos como melhor conjunto de parâmetros aquele que se encaixa nas técnicas apresentadas na Tabela 2 e tem a melhor posição na Figura 10. Este é o conjunto representado em laranja: “indivíduos: 40, seleção: torneio, taxa de mutação: 0,2, taxa de elitismo: 0,1, taxa de acasalamento: 0,7, indivíduos aleatórios: nova população”. Acreditamos que este conjunto foi o mais próximo do que poderíamos afirmar como o sendo o melhor conjunto.

c.2 Busca Local

O trabalho de Souza (2022) definiu um novo método de busca local para o problema NDP para malha viária de cidades. A mais importante definição para o contexto da específica deste projeto 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)

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

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

$s^* \leftarrow s$;

while *not*($P(s) = 1$) **do**

 seleciona $s' \in N(s)$ de acordo com $P(s)$;

$s \leftarrow s'$;

if $\alpha(s) < \alpha(s^*)$ **then**

$s^* \leftarrow s$;

end

end

return s^*

Algoritmo 1: Algoritmo Genérico de Busca Local Monótona

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

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 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 OpenStreetMap (OSM). Sobre esse grafo é analisado o tempo médio dos trajetos, utilizando o algoritmo 2. 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.

Otimização de tempo de viagem

Entrada: Grafo direcionado G , valor mínimo V **Saída:** média de tempo de viagem

$M \leftarrow G.nodes;$

$total_time \leftarrow 0;$

$total_routes \leftarrow 0;$

for $origin \in nodes$ **do**

for $dest \in nodes$ **do**

if $origin \neq dest$ **then**

$route_time \leftarrow shortest_path(origin, dest);$

if $route_time \geq V$ **then**

$total_time \leftarrow total_time + route_time;$

$total_routes \leftarrow total_routes + 1;$

end

end

end

end

$M \leftarrow total_time/total_routes;$

return $M;$

Algoritmo 2: Algoritmo para calculo de média de tempo

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 [3](#), [4](#), [5](#), [6](#). Pode-se observar a média inicial do tempo das viagens, a média otimizada, o numero 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 [12](#), [13](#), [14](#), [15](#), [16](#) [17](#) mostram os mapas antes de serem otimizados (com um *colormap* somente ilustrativo).

As Figuras [18](#), [19](#), e [20](#) 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 [20](#) 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.

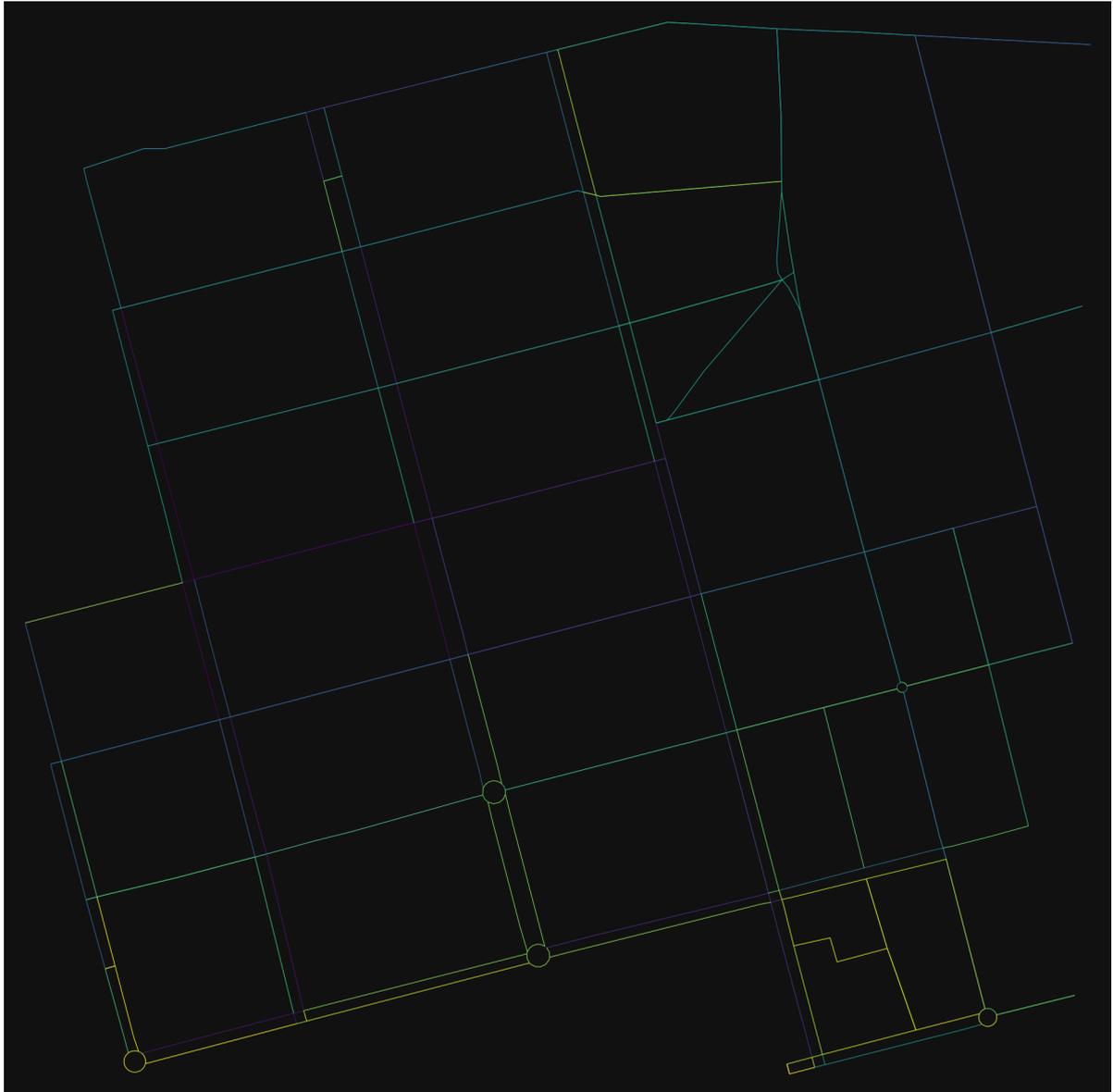


Figura 12 – Mapa de Chapecó com 450m de raio

Tabela 3 – 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

Pode-se observar na diferença dos tempos de execução, por exemplo entre as tabelas 4, e 6, 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

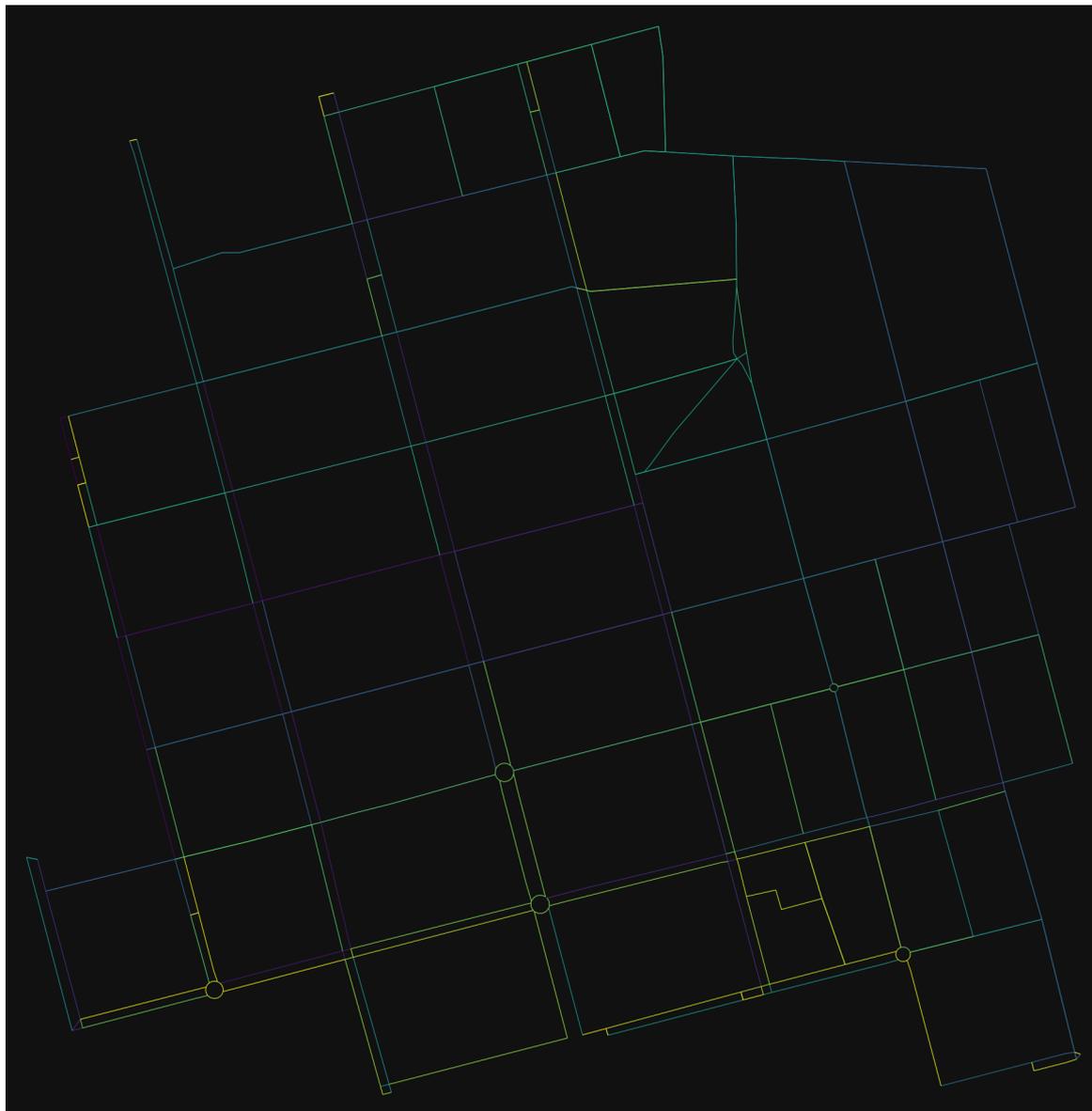


Figura 13 – Mapa de Chapecó com 550m de raio

Tabela 4 – 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 5 – 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 6 – 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



Figura 14 – Mapa da região da UFSC com 450m de raio

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 2, 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 numero de arcos e V o numero 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 calculo 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 $O(V^2 A^2 + AV^3 \log V)$. 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



Figura 15 – Mapa da região da UFSC com 550m de raio

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 4 e 5 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.

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,

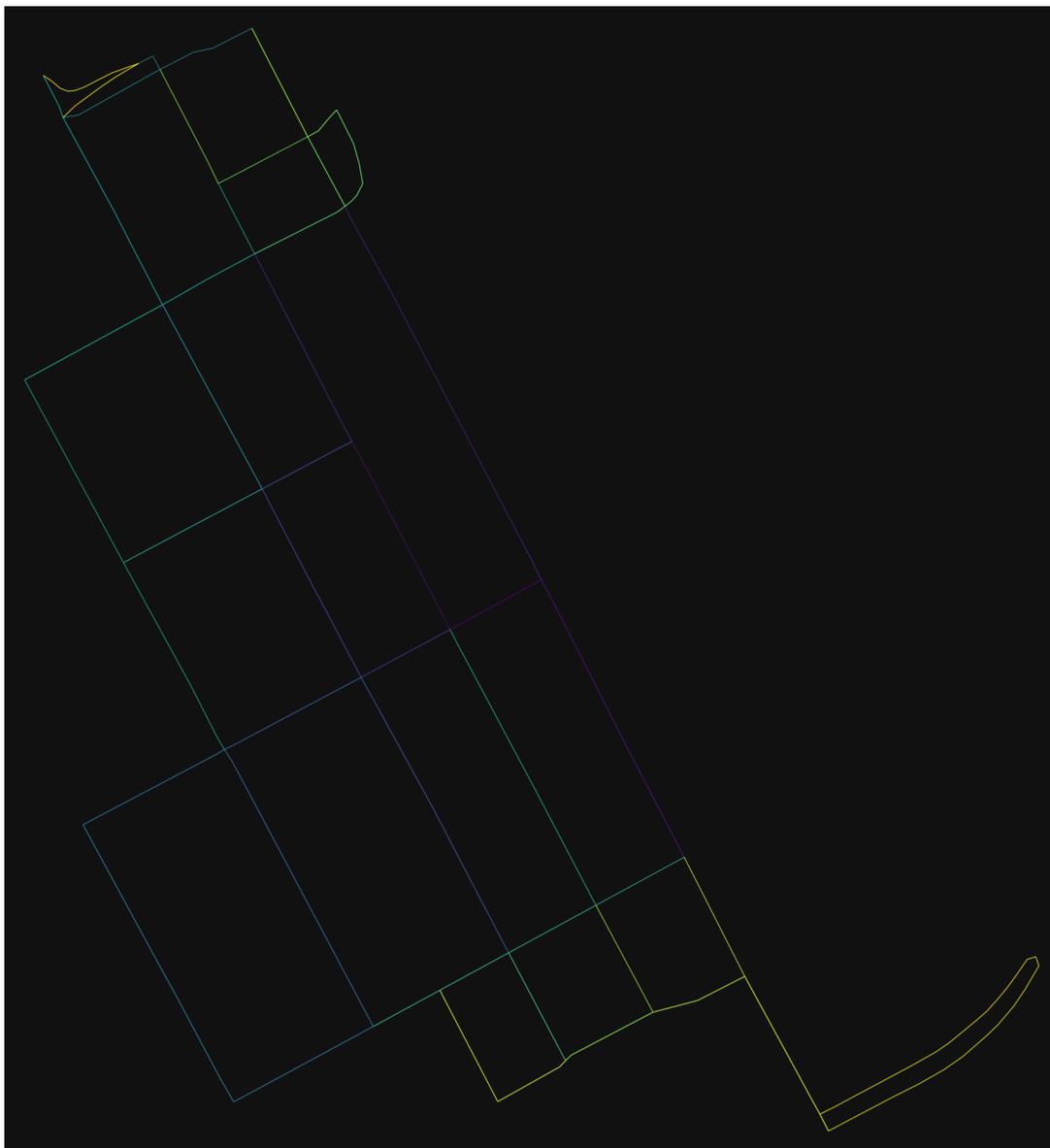


Figura 16 – mapa de Nashville com 450m de raio

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

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

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

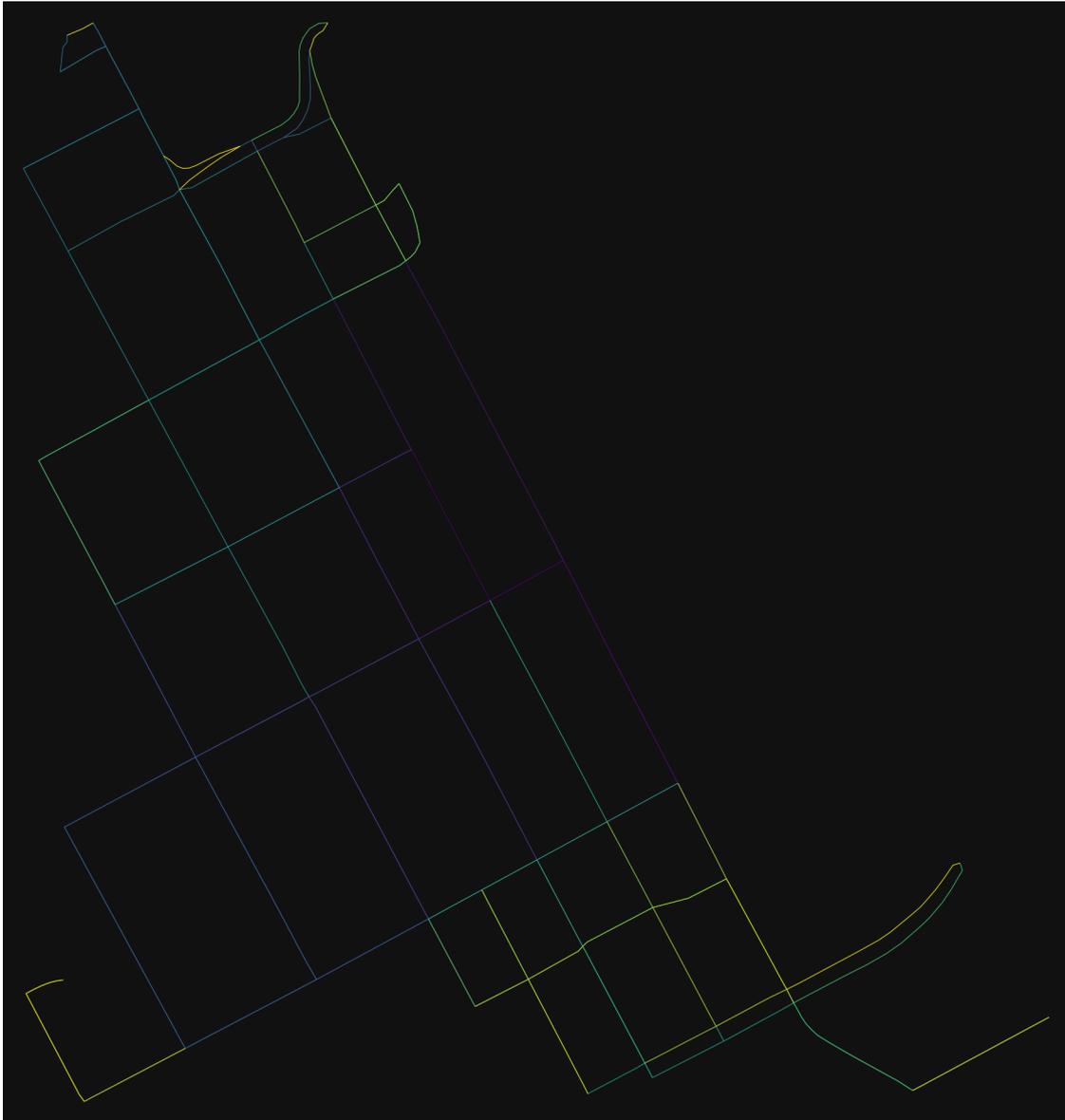


Figura 17 – mapa de Nashville com 550m de raio

- 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 8:

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

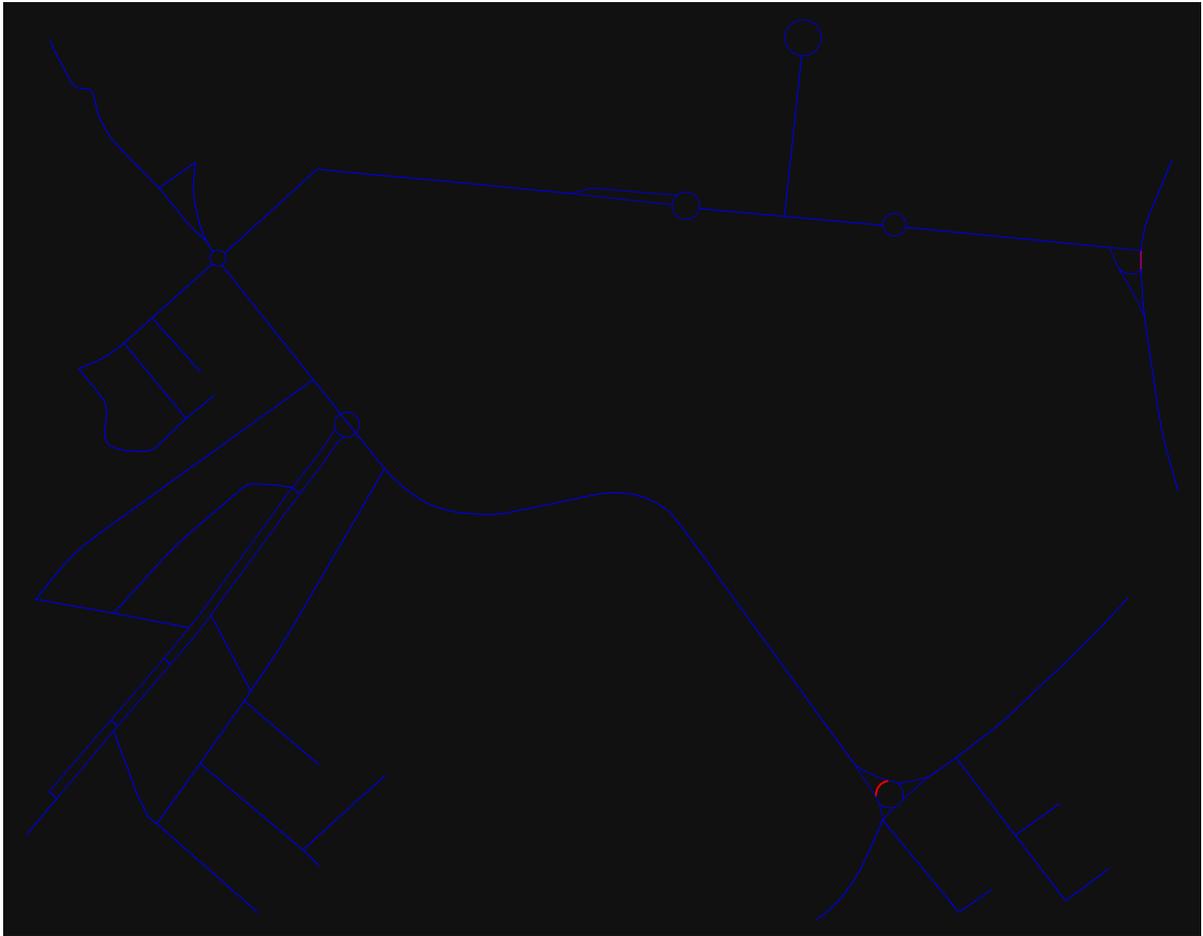


Figura 18 – Mapa da UFSC otimizado com R\$ 10.000

Tabela 7 – Highway level of service(LOS) and congestion conditions

LOS	Traffic Conditions	Volume/Capacity Ratio	Maximum Density (vehicule/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 8 – Tabela encontrada em [Salman e Alaswad \(2018\)](#)



Figura 19 – Mapa da UFSC otimizado com R\$ 50.000

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 cálculo 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 9, 10, 11, 12, 13, 14. 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 10 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

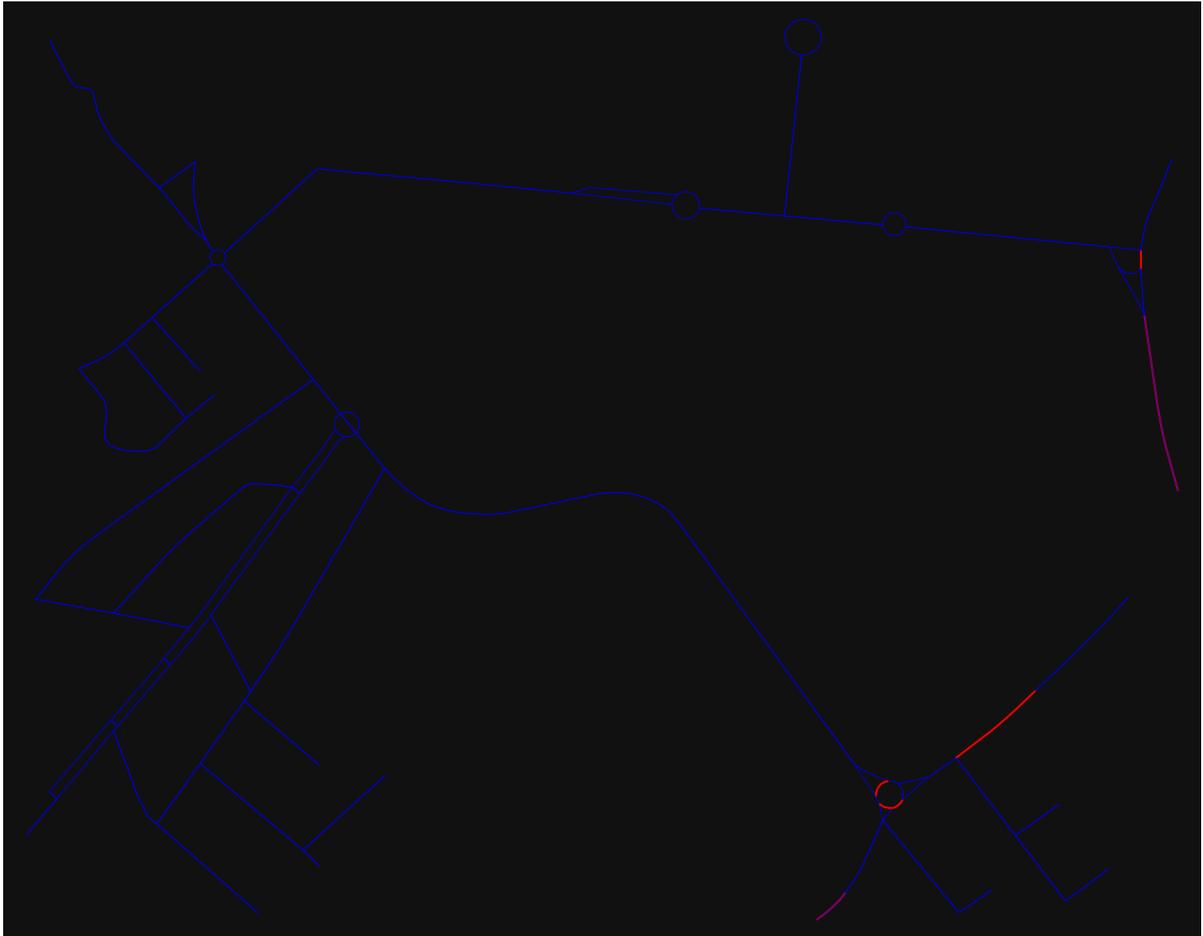


Figura 20 – Mapa da UFSC otimizado com R\$ 150.000

tempo de execução por passo, encontramos uma média de 361s por passo nos 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 número de nós, dependendo apenas do número 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 9 – 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 10 – 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	1356.7121250000001	13	9660,576185s

Tabela 11 – 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

Podemos ver nas Figuras 21 e 22 um exemplo de otimização para o mapa de chapecó 450m, com 1000m de pista disponíveis para otimização respectivamente. Onde as cores do *heatmap* foram mapeadas para as classes definidas em 8. 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.

d Contribuição técnico-científica

Considera-se que como contribuição técnico-científicas, além das análises relatadas, há a disposição da comunidade, os repositórios com código-fonte do que foi ou do que está sendo produzido:

- Algoritmo genético de Filho (2022) e Schmitz (2022): Schmitz, Filho e Santiago (2022);
- Busca local de (SOUZA, 2022): Souza e Santiago (2022);
- Busca local (busca Tabu) de João Pedro Perez Resmer: Resmer e Santiago (2023) (a ser disponibilizado em breve).

Tabela 12 – 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 13 – 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	233.75177500000004	3	25,793708s

Tabela 14 – 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

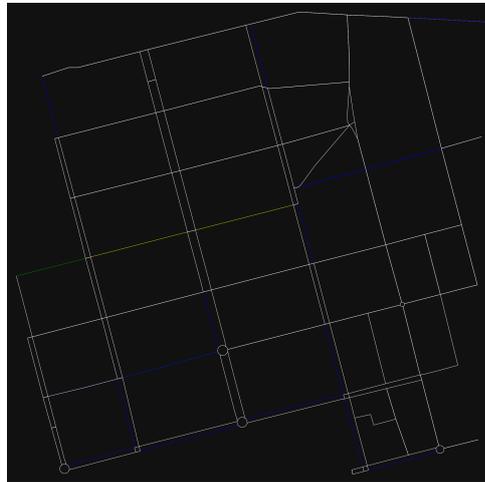


Figura 21 – Densidades padrão

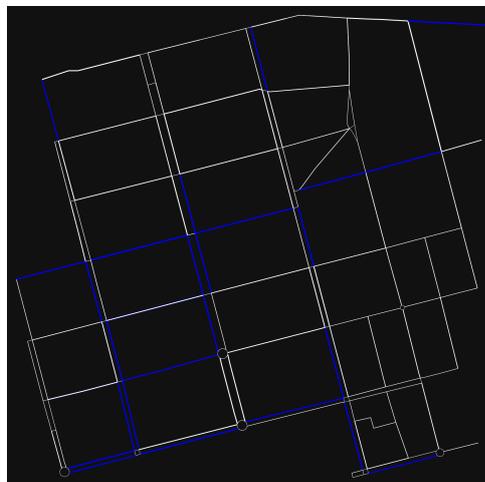


Figura 22 – Densidades otimizadas

Um relatório técnico foi preparado e disponibilizado no repositório da biblioteca universitária da UFSC contendo o relato dos resultados apresentados neste relatório final.

Agradecimentos

Agradecemos o apoio do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina (UFSC) e ao CNPq (Centro Nacional de Desenvolvimento Científico e Tecnológico) pelas bolsas disponibilizadas através do programa de PIBIC.

Referências

AKHAND, M. A. et al. A PSO based transportation network design optimization of the mega city Dhaka. *2nd International Conference on Electrical Engineering and Information and Communication Technology, iCEEiCT 2015*, IEEE, n. May, p. 1–7, 2015. Citado na página 2.

BAGLOEE, S. A.; SARVI, M.; PATRIKSSON, M. A Hybrid Branch-and-Bound and Benders Decomposition Algorithm for the Network Design Problem. *Computer-Aided Civil and Infrastructure Engineering*, v. 32, n. 4, p. 319–343, 2017. ISSN 14678667. Citado na página 2.

BHASKAR, U.; LIGETT, K.; SCHULMAN, L. J. Network improvement for equilibrium routing. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 8494 LNCS, n. 2, p. 138–149, 2014. ISSN 16113349. Citado na página 2.

CHEN, X. M. et al. Surrogate-based optimization for solving a mixed integer network design problem. *Transportation Research Record*, v. 2497, p. 124–134, 2015. ISSN 21694052. Citado na página 2.

FARVARESH, H.; SEPEHRI, M. M. A single-level mixed integer linear formulation for a bi-level discrete network design problem. *Transportation Research Part E: Logistics and Transportation Review*, Elsevier Ltd, v. 47, n. 5, p. 623–640, 2011. ISSN 13665545. Citado 2 vezes nas páginas 3 e 4.

FILHO, N. L. J. *Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre - Algoritmo Genético*. Florianópolis, SC: [s.n.], 2022. Citado 3 vezes nas páginas 5, 15 e 32.

FONTAINE, P.; MINNER, S. Benders decomposition for discrete-continuous linear bilevel problems with application to traffic network design. *Transportation Research Part B: Methodological*, Elsevier Ltd, v. 70, p. 163–172, 2014. ISSN 01912615. Citado 2 vezes nas páginas 3 e 4.

FONTAINE, P.; MINNER, S. A dynamic discrete network design problem for maintenance planning in traffic networks. *Annals of Operations Research*, Springer US, v. 253, n. 2, p. 757–772, 2017. ISSN 1572-9338. Citado na página 2.

FONTAINE, P.; MINNER, S. Benders decomposition for the Hazmat Transport Network Design Problem. *European Journal of Operational Research*, Elsevier B.V., v. 267, n. 3, p. 996–1002, 2018. ISSN 03772217. Citado na página 2.

FUKUNAGA, T. Spider covers for prize-collecting network activation problem. *ACM Transactions on Algorithms*, v. 13, n. 4, 2017. ISSN 15496333. Citado na página 2.

GAO, Z.; WU, J.; SUN, H. Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, v. 39, n. 6, p. 479–495, 2005. ISSN 01912615. Citado 2 vezes nas páginas 3 e 4.

- INRIX. *INRIX 2018 Global Traffic Scorecard*. 2018. Citado na página 2.
- ISLAM, K. A. et al. A heuristic aided Stochastic Beam Search algorithm for solving the transit network design problem. *Swarm and Evolutionary Computation*, Elsevier B.V., v. 46, n. February, p. 154–170, 2019. ISSN 22106502. Citado na página 2.
- ISRAELI, Y.; CEDER, A. Transit route design using scheduling and multiobjective programming techniques. In: DADUNA, J. R.; BRANCO, I.; PAIXÃO, J. M. P. (Ed.). *Computer-Aided Transit Scheduling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. p. 56–75. ISBN 978-3-642-57762-8. Citado na página 3.
- JAFARI, E.; PANDEY, V.; BOYLES, S. D. A decomposition approach to the static traffic assignment problem. *Transportation Research Part B: Methodological*, Elsevier Ltd, v. 105, p. 270–296, 2017. ISSN 01912615. Citado na página 2.
- KAYA, O.; OZKOK, D. A network design problem with location, inventory and routing decisions. *GECCO 2018 Companion - Proceedings of the 2018 Genetic and Evolutionary Computation Conference Companion*, p. 139–140, 2018. Citado na página 2.
- KIAN, R.; KARGAR, K. Comparison of the formulations for a hub-and-spoke network design problem under congestion. *Computers and Industrial Engineering*, Elsevier Ltd, v. 101, p. 504–512, 2016. ISSN 03608352. Citado na página 2.
- KOLAK, O. I.; FEYZIOĞLU, O.; NOYAN, N. Bi-level multi-objective traffic network optimisation with sustainability perspective. *Expert Systems with Applications*, v. 104, p. 294–306, 2018. ISSN 09574174. Citado 3 vezes nas páginas 2, 3 e 4.
- LABIDI, M. K. et al. A parallel hybrid genetic algorithm for the K-edge-connected hop-constrained network design problem. *GECCO 2016 - Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, p. 685–692, 2016. Citado na página 2.
- LEBLANC, L. J. An algorithm for the discrete network design problem. *Transportation Science*, INFORMS, v. 9, n. 3, p. 183–199, 1975. ISSN 00411655, 15265447. Citado na página 2.
- LIN, B. et al. Modeling the railway network design problem: A novel approach to considering carbon emissions reduction. *Transportation Research Part D: Transport and Environment*, Elsevier Ltd, v. 56, p. 95–109, 2017. ISSN 13619209. Citado na página 2.
- LUATHEP, P. et al. Global optimization method for mixed transportation network design problem: A mixed-integer linear programming approach. *Transportation Research Part B: Methodological*, Elsevier Ltd, v. 45, n. 5, p. 808–827, 2011. ISSN 01912615. Citado na página 4.
- 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/>>. Citado na página 22.
- NAYEEM, M. A.; ISLAM, M. M.; YAO, X. Solving Transit Network Design Problem Using Many-Objective Evolutionary Approach. *IEEE Transactions on Intelligent Transportation Systems*, p. 1–12, 2018. ISSN 15249050. Citado 2 vezes nas páginas 2 e 3.

OWAIS, M.; OSMAN, M. K.; MOUSSA, G. Multi-objective transit route network design as set covering problem. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 17, n. 3, p. 670–679, 2016. ISSN 15249050. Citado na página 2.

PARASKEVOPOULOS, D. C.; GÜREL, S.; BEKTAŞ, T. The congested multicommodity network design problem. *Transportation Research Part E: Logistics and Transportation Review*, v. 85, p. 166–187, 2016. ISSN 13665545. Citado na página 2.

POORZAHEDY, H.; TURNQUIST, M. A. Approximate algorithms for the discrete network design problem. *Transportation Research Part B*, v. 16, n. 1, p. 45–55, 1982. ISSN 01912615. Citado 2 vezes nas páginas 2 e 3.

RESMER, J. P. P.; SANTIAGO, R. de. *Código-fonte: Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre - busca local monótona*. Florianópolis: [s.n.], 2023. Disponível em: <<https://github.com/jresmer/busca-tabu>>. Citado na página 32.

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>>. Citado na página 20.

RODRÍGUEZ-MARTÍN, I.; SALAZAR-GONZÁLEZ, J. J.; YAMAN, H. A branch-and-cut algorithm for two-level survivable network design problems. *Computers and Operations Research*, Elsevier, v. 67, p. 102–112, 2016. ISSN 03050548. Citado na página 2.

SALMAN, S.; ALASWAD, S. Alleviating road network congestion: Traffic pattern optimization using Markov chain traffic assignment. *Computers and Operations Research*, Elsevier Ltd, v. 99, p. 191–205, 2018. ISSN 03050548. Citado 7 vezes nas páginas 2, 4, 5, 6, 26, 27 e 29.

SCHMITZ, G. C. *Otimização de malha viária terrestre através de algoritmo genético*. Monografia (Bacharelado em Sistemas de Informação) — Sistemas de Informação, Universidade Federal de Santa Catarina, Florianópolis, 2022. Citado 2 vezes nas páginas 5 e 32.

SCHMITZ, G. C.; FILHO, N. L. J.; SANTIAGO, R. de. *Código-fonte: Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre - algoritmo genético*. Florianópolis: [s.n.], 2022. Disponível em: <<https://github.com/njoppi2/OTIMIZACAO-DE-MALHA-VIARIA-TERRESTRE-ATRAVES-DE-ALGORITMO-GENETICO>>. Citado na página 32.

SOUZA, A. V. d.; SANTIAGO, R. de. *Código-fonte: Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre - busca local monótona*. Florianópolis: [s.n.], 2022. Disponível em: <<https://github.com/anaemdsz/malha-viaria-busca-local>>. Citado na página 32.

SOUZA, A. V. de. *Metaheurística para Alteração, Adição e Remoção de Estradas em Projetos de Malha Viária Terrestre*. Monografia (Bacharelado em Ciências da Computação) — Ciências da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2022. Citado 2 vezes nas páginas 20 e 32.

SUN, Z. Continuous Transportation Network Design Problem Based on Bi-level Programming Model. *Procedia Engineering*, Elsevier B.V., v. 137, p. 277–282, 2016. ISSN 18777058. Citado 2 vezes nas páginas 3 e 4.

TOMTOM. *Traffic Congestion Ranking*. 2018. Citado na página 2.

WANG, D. Z.; LO, H. K. Global optimum of the linearized network design problem with equilibrium flows. *Transportation Research Part B: Methodological*, Elsevier Ltd, v. 44, n. 4, p. 482–492, 2010. ISSN 01912615. Citado na página 4.

WANG, D. Z. W.; LIU, H.; SZETO, W. Y. A novel discrete network design problem formulation and its global optimization solution algorithm. *Transportation Research Part E*, Elsevier Ltd, v. 79, p. 213–230, 2015. ISSN 1366-5545. Citado na página 2.

WANG, J.; PEETA, S.; HE, X. Multiclass traffic assignment model for mixed traffic flow of human-driven vehicles and connected and autonomous vehicles. *Transportation Research Part B: Methodological*, Elsevier Ltd, v. 126, p. 139–168, 2019. ISSN 01912615. Citado na página 2.

WARDROP, J. G. SOME THEORETICAL ASPECTS OF ROAD TRAFFIC RESEARCH. *Proceedings of the Institution of Civil Engineers*, v. 1, n. 3, p. 325–362, may 1952. ISSN 1753-7789. Citado na página 4.

YANG, H.; BELL, M. G. Models and algorithms for road network design: A review and some new developments. *Transport Reviews*, v. 18, n. 3, p. 257–278, 1998. ISSN 14645327. Citado na página 4.

YU, B. et al. A bi-level programming for bus lane network design. *Transportation Research Part C: Emerging Technologies*, Elsevier Ltd, v. 55, p. 310–327, 2015. ISSN 0968090X. Citado 2 vezes nas páginas 3 e 4.

YILDIZ, B.; KARASAN, O. E.; YAMAN, H. Branch-and-price approaches for the network design problem with relays. *Computers and Operations Research*, v. 92, p. 155–169, 2018. ISSN 03050548. Citado na página 2.

ZHANG, H.; GAO, Z. Bilevel programming model and solution method for mixed transportation network design problem. *Journal of Systems Science and Complexity*, v. 22, n. 3, p. 446–459, 2009. ISSN 10096124. Citado 2 vezes nas páginas 3 e 4.