

UNIVERSIDADE FEDERAL DO PARANÁ

DANILO JORGE DOS SANTOS NAKONECZNY

MÉTODO COMPUTACIONAL PARA FLUXO DE TRÁFEGO COM OBJETIVO DE
MINIMIZAR A RETENÇÃO DE VEÍCULOS EM UMA REDE URBANA

CURITIBA

2019

DANILO JORGE DOS SANTOS NAKONECZNY

MÉTODO COMPUTACIONAL PARA FLUXO DE TRÁFEGO COM OBJETIVO DE
MINIMIZAR A RETENÇÃO DE VEÍCULOS EM UMA REDE URBANA

Dissertação apresentada ao curso de Pós-Graduação em Métodos Numéricos em Engenharia, Setor de Ciências Exatas e Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Métodos Numéricos em Engenharia.

Orientadora: Profa. Dra. Liliana M. Gramani

Coorientador: Prof. Dr. Eloy Kaviski

CURITIBA

2019

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

N163m

Nakoneczny, Danilo Jorge dos Santos

Método computacional para fluxo de tráfego com objetivo de minimizar a retenção de veículos em uma rede urbana [recurso eletrônico] / Danilo Jorge dos Santos Nakoneczny. – Curitiba, 2019.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2019.

Orientador: Liliansa Madalena Gramani – Coorientador: Eloy Kaviski.

1. Trânsito urbano. 2. Trânsito – Controle eletrônico. 3. Trânsito – Sinais e sinalização. 4. Trânsito – Congestionamento. 5. Programação heurística. I. Universidade Federal do Paraná. II. Gramani, Liliansa Madalena. III. Kaviski, Eloy. IV Título.

CDD: 388.41

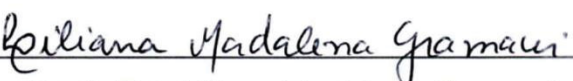
Bibliotecário: Elias Barbosa da Silva CRB-9/1894


TERMO DE APROVAÇÃO

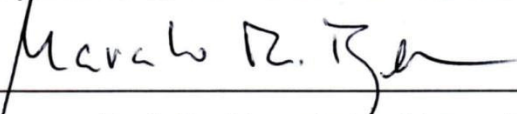
DANILO JORGE DOS SANTOS NAKONECZNY

MÉTODO COMPUTACIONAL PARA FLUXO DE TRÁFEGO COM OBJETIVO DE
MINIMIZAR A RETENÇÃO DE VEÍCULOS EM UMA REDE URBANA

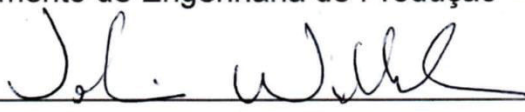
Dissertação aprovada como requisito parcial à obtenção do título de Mestre em
Métodos Numéricos em Engenharia – Programa de Pós-Graduação em Métodos
Numéricos em Engenharia, Universidade Federal do Paraná, pela seguinte banca
examinadora:


Profª. Drª. Liliansa Madalena Gramani
Orientadora – Departamento de Matemática – UFPR


Prof. Dr. Eloy Kaviski
Departamento de Hidráulica e Saneamento – UFPR


Prof. Dr. Marcelo Rodrigues Bessa
Departamento de Hidráulica e Saneamento – UFPR


Profª. Drª. Neida Maria Patias Volpi
Departamento de Engenharia de Produção – UFPR


Prof. Dr. Volmir Eugênio Wilhelm
Departamento de Engenharia de Produção – UFPR

Curitiba, 06 de março de 2019

AGRADECIMENTOS

Agradeço a Deus, por ter me fortalecido e dado saúde para poder realizar este trabalho.

À minha esposa Sahra, pela compreensão da minha ausência, por me motivar e ajudar a prosseguir.

Aos meus pais Joacir e Rosangela, pelo cuidado, carinho e investimento que realizaram em minha vida, e o apoio as minhas decisões.

À Liliana M. Gramani, minha Orientadora, por acreditar, confiar e me ajudar em muitos encontros que tivemos para obtenção deste trabalho.

Ao Eloy Kaviski, meu Coorientador, por sempre se dispor em ajudar, principalmente no âmbito da lógica e programação computacional presente neste trabalho.

À CAPES, por ter concedido a bolsa de estudo de maneira que fosse possível me dedicar com melhor qualidade de tempo aos estudos.

À SETRAN, pela contribuição no fornecimento dos dados.

Aos meus amigos e colegas de pós-graduação.

Agradeço também a todos a minha volta que de alguma maneira contribuíram para a concretização deste estudo.

*"Eu o instruirei e o ensinarei no caminho que você deve seguir;
eu o aconselharei e cuidarei de você."*

Salmos 32:8

RESUMO

O presente trabalho apresenta um modelo matemático de otimização baseando-se em equações de conservação de massa, que possibilita fornecer uma melhor programação de tempos de vermelho e verde em semáforos localizados em vias com alto fluxo veicular. De tal maneira que a função objetivo do modelo elaborado minimiza a quantidade de veículos em espera nos semáforos. Para isso, utiliza-se uma heurística de otimização denominada de busca em vizinhança variável (VNS). O modelo foi implementado e testado computacionalmente com dados reais de fluxo veicular em alguns trechos de uma rodovia brasileira localizada na capital do estado do Paraná, Curitiba. Com os resultados obtidos para a otimização de tráfego veicular, analisou-se o impacto na função objetivo do modelo ocasionado por algumas alterações dos parâmetros. Dentre eles, a velocidade média do fluxo veicular e o intervalo de tempo em segundos que os dados são discretizados e ocorre as simulações, sendo este denominado de tamanho do passo. Portanto, é observado que pequenas mudanças no tamanho do passo não geram grandes melhorias na função objetivo quando o passo é pequeno.

Palavras-chave: Fluxo veicular. Heurística Variable Neighborhood Search - VNS. Programação Semafórica. Otimização de Tráfego. Minimização de Esperas em Semáforos.

ABSTRACT

The present work introduces a mathematical shape of optimization based on the mass conservation equations, where it is possible to provide a better scheduling of red and green times at the traffic lights located in routes with a high vehicular flow in such a way that the function the objective of the elaborated model downplay the number of waiting vehicles at the traffic lights. For that, we use an heuristic optimization that is called Variable Neighborhood Search (VNS). The model was computationally implemented and tested computationally with some real data from the vehicle flow. in some shields of a Brazilian highway located in the capital of the state of Paraná Curitiba. With the obtinded results for the veicular traffic optimization it was analyzed the impact of the objective function caused by some paramenters modifications. Among them is the speed avarege of the veicular flow, the program's implementation time and the step size in the simulations showing up that small changes in pitch size do not generate great improvements in the objective function when the step is small.

Keywords: Vehicular flow. Heurístic Variable Neighborhood Search – VNS. Semaphore Programming. Traffic Optimization. Minimization of waiting at traffic lights.

SUMÁRIO

1	INTRODUÇÃO	9
1.1	HIPÓTESE	10
1.2	OBJETIVOS	10
1.3	JUSTIFICATIVAS	111
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO	11
2	FUNDAMENTAÇÃO TEÓRICA.....	13
2.1	REVISÃO BIBLIOGRÁFICA	13
2.2	MODOS DE OPERAÇÃO DOS CONTROLADORES SEMAFÓRICOS NO BRASIL	16
2.2.1	Modo intermitente	16
2.2.2	Modo centralizado	16
2.2.3	Modo manual.....	17
2.2.4	Modo fixo.....	17
2.2.5	Modo atuado.....	18
2.3	TIPOS DE ESCALAS	19
2.3.1	Escala microscópica	20
2.3.2	Escala macroscópica.....	21
2.3.3	Escala cinética.....	23
2.4	MÉTODOS DE COORDENAÇÃO DE SEMÁFOROS	233
2.4.1	TRANSYT.....	24
2.4.2	SCOOT.....	29
2.4.3	SIRI	311
2.5	CÓDIGO DE TRÂNSITO BRASILEIRO	344
3	MODELOS E MÉTODOS	355
3.1	MODELAGEM MATEMÁTICA	377
3.2	ESTUDO DOS PARÂMETROS	411
3.2.1	Fluxo máximo	411
3.2.2	Uniformização do fluxo	433
3.2.3	Tempo morto	455
3.2.4	Tempo de amarelo	47
3.2.5	Tempo de percepção e reação	48
3.2.6	Taxa de desaceleração	488
3.2.7	Tempo de vermelho geral	488
3.2.8	Tempo aproveitado após o final do sinal verde.....	49
3.2.9	Tempo perdido no início do verde.....	50
3.2.10	Cálculo do tempo morto.....	500
3.3	VALORES CONSIDERADOS	51
3.4	HEURÍSTICA <i>VARIABLE NEIGHBORHOOD SEARCH</i> - VNS	52
4	ESTUDO DO CASO E RESULTADOS	57

4.1	LOCAL DE ESTUDO	57
4.2	RESULTADOS	59
5	CONCLUSÕES E RECOMENDAÇÕES.....	877
	REFERÊNCIAS	888
	GLOSSÁRIO.....	93
	APÊNDICE 1 – CÓDIGO DO PROGRAMA ELABORADO EM PASCAL	94

1 INTRODUÇÃO

A chegada de grandes montadoras de automóveis no Brasil e a facilitação para financiamento de veículos fez com que nos últimos anos houvesse um notável aumento dos mesmos. Por um lado, tal crescimento contribuiu em parte para o crescimento da economia (PACHECO & TURETTI, 2003), mas em contrapartida, o aumento de veículos tem se tornado um problema recorrente nas cidades de grande e médio porte. O tráfego de automóveis tem se tornado mais intenso, causando muitos congestionamentos e, conseqüentemente, no aumento do tempo de viagem e no estresse dos motoristas, reduzindo assim a qualidade de vida e produtividade dos mesmos. Com isso, as autoridades responsáveis pelo setor vêm buscando formas de transformar o fluxo de veículos utilitários, ônibus e caminhões, a transitarem de forma harmoniosa.

Visando minimizar os problemas do trânsito, as autoridades têm criado linhas exclusivas para ônibus, períodos de tráfegos para transporte de mercadorias e limitação de velocidade. E vias de grande fluxo também é necessário adotar outras iniciativas como a sincronização de semáforos para minimizar os congestionamentos e o tempo de espera nos mesmos ou realizar a construção de viadutos. Entretanto, esta última iniciativa é pouco adotada por possuir custos elevados e um longo tempo de planejamento e execução.

Portanto, o desenvolvimento de modelos matemáticos que sejam capazes de fornecer informações para uma melhor programação de semáforos de maneira a minimizar a retenção de veículos em uma malha viária tem sido muito estudado e pode ser de grande ajuda para a melhoria no trânsito.

Entende-se por sincronização de semáforos o plano em que a diferença entre os tempos de sinalização em verde e vermelho, deve ser tal que um motorista, trafegando na velocidade máxima da via, encontre ao longo do seu trajeto a maior quantidade possível de sinais verdes.

Neste trabalho serão realizadas simulações computacionais com base em um modelo matemático de sincronização de semáforos, para um trecho de uma via denominada Linha Verde da cidade de Curitiba.

1.1 HIPÓTESE

Nesta dissertação será considerada uma abordagem macroscópica, onde o fluxo de veículos é considerado como um escoamento compressível. Assim, o problema poderá ser resolvido através de uma analogia com a hidrodinâmica, com base nas equações de conservação de massa e em modelos fenomenológicos onde, para minimizar o acúmulo de veículos nos semáforos, será utilizado alguns conceitos da propagação de vazões em reservatórios (PINTO; HOLTZ; GOMIDE, 1976).

1.2 OBJETIVOS

- Objetivo Geral

Desenvolver um modelo matemático que seja capaz de fornecer o tempo de sinalização de verde e vermelho em semáforos de uma via e suas transversais, para uma melhor programação de seus semáforos de maneira a minimizar a retenção de veículos. Portanto, o modelo não leva em consideração o tempo de defasagem entre os semáforos, ou seja, não maximiza necessariamente a largura da banda de passagem.

- Objetivos Específicos

- a) Descrever os modos de funcionamento dos semáforos.
- b) Desenvolver um programa que resolva o modelo a ser proposto.
- c) Analisar a sensibilidade dos parâmetros utilizados no programa.
- d) Adaptar o algoritmo de Kaviski (OLIVEIRA, 2015), para resolução do modelo a ser aplicado em um trecho da Linha Verde.
- e) Implementar um algoritmo de otimização heurístico.
- f) Analisar a importância dos parâmetros relacionados ao programa através da análise feita no item c.

1.3 JUSTIFICATIVAS

Com a implementação de uma melhoria na programação de semáforos gera-se economia de tempo e de recursos, melhoria na mobilidade urbana e na qualidade de vida dos motoristas. A seguir, na figura (1.1), pode-se observar o transtorno que uma falha nos equipamentos semaforicos pode ocasionar.

FIGURA 1.1 – UMA FALHA NO SEMÁFORO.



FONTE: FOLHA DE S. PAULO (2017).

Segundo artigo publicado na Folha de S. Paulo (2017), uma falha no semáforo no dia 02 de fevereiro de 2017 entre as avenidas Faria Lima e Juscelino Kubitschek, zona oeste de São Paulo ilustrada na figura (1.1), travou a movimentação de carros e após a chegada de agentes da CET (Companhia de Engenharia de Tráfego) foram necessários cerca de 20 minutos para resolver o problema.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho está organizado da seguinte forma:

- No Capítulo 2 é apresentada uma revisão bibliográfica, os modos de atuação dos controladores semaforicos, as formas que os modelos de tráfego podem

ser representados e três modelos de coordenação semafórica. Ao fim deste capítulo encontram-se algumas definições de trânsito e semáforo segundo o código de trânsito brasileiro.

- No Capítulo 3 é feita a descrição do modelo matemático proposto e um estudo dos parâmetros a serem usados no modelo. Depois, apresenta-se a heurística para a otimização do modelo.
- No Capítulo 4 encontra-se a descrição do local de estudo e os resultados obtidos.
- No Capítulo 5 são apresentadas as conclusões e recomendações para as simulações de fluxo de tráfego.
- No Apêndice 1 encontra-se o algoritmo desenvolvido para este trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados uma revisão bibliográfica e os modos de operação dos controladores semafóricos utilizados no Brasil, entre eles o modo de tempo fixo no qual o presente trabalho se baseia, bem como as formas que os modelos de tráfego podem ser representados e três modelos de coordenação semafórica utilizados comercialmente em território nacional e internacional. Ao final deste capítulo apresentam-se algumas definições de trânsito e semáforo segundo o código de trânsito brasileiro.

2.1 REVISÃO BIBLIOGRÁFICA

Uma proposta para melhorar a programação semafórica trazida por CARMO (2016) foi a de utilizar um Sistema de Informações Geográficas - SIG para auxiliar na programação semafórica, onde propôs uma arquitetura que contemplasse a questão Geo-Temporal para analisar os principais eixos de congestionamento de uma malha viária. Tal arquitetura refere-se à união de informações do volume do tráfego veicular em um banco de dados, cálculo das medidas de desempenho e da integração com um Sistema de Informações Geográficas para analisar possíveis alternativas na reconfiguração do sistema de tráfego viário. O autor aplicou seu modelo utilizando dados cartográficos digitais do tráfego de uma cidade específica.

Para otimizar a programação semafórica de tempo real, os autores GONZAGA, ALMEIDA e CARRANO (2015) propuseram um método de otimização que utiliza um modelo matemático para avaliação das possíveis soluções, ao invés de utilizarem no algoritmo simulações do tráfego para avaliarem os cenários para diferentes programações. Neste modelo busca-se, obter melhorias no tempo de busca por uma solução, de maneira que a programação semafórica em tempo real seja mais eficiente. Portanto, a metodologia dos autores consiste em utilizar o simulador de tráfego apenas uma vez, para inicializar a execução do programa de maneira a construir um cenário para os fluxos de veículos em cada trecho. Com base nesses fluxos, um modelo matemático proposto por AKÇELIK (2000) é inicializado, onde tal modelo tem como função objetivo minimizar os tempos médios que os veículos gastam para percorrer os trechos da malha viária estudada. Já para se obter uma boa

solução foi utilizado o Algoritmo Genético – AG para otimizar a busca. Com isso, segundo os autores, a nova arquitetura adotada tem sido promissora para os testes preliminares, onde constaram uma melhoria de 100 vezes no tempo de execução do programa em relação as arquiteturas clássicas que usam simulações para avaliar cada solução candidata.

No artigo desenvolvido por LACORTT e KRIPKA'S (2013) são comparados dois modelos matemáticos de otimização semaforica, onde o 1º modelo busca minimizar atraso uniforme e o atraso aleatório utilizando as expressões propostas por Webster, e também minimizar o número de paradas sofridas por um veículo em uma malha semaforizada. Já o 2º modelo adotado, tem como objetivo minimizar o tempo total de deslocamento de um veículo em uma malha semaforizada de acordo com a expressão de GOBBO (2005). Em ambos os modelos a variável é o tempo de verde normal de cada cruzamento analisado, que segunda a restrição, deve ser maior ou igual ao tempo perdido e menor ou igual ao tempo de ciclo. Os modelos foram implementados em linguagem Fortran e para otimização foi utilizada uma meta-heurística denominada *Simulated Annealing* (KIRKPATRICK et al., 1983). Posteriormente os modelos foram testados com a utilização de dados reais da cidade de Passo Fundo – RS, onde os resultados obtidos demonstraram melhorias no fluxo veicular em relação a atual programação adotada na rede estudada. O desempenho dos modelos apresentados foram equivalentes, apresentando as mesmas soluções, segundo os autores.

Buscando encontrar uma melhor sincronização dos semáforos aplicada ao tráfego de Curitiba, JONER e VOLPI (2013) desenvolveram um modelo matemático baseado nos trabalhos de LITTLE (1966) e LITTLE, KELSON e GARTNER (1981), o qual possui como objetivo maximizar a soma ponderada das larguras de banda de todas as vias da malha considerada, onde as vias com maior importância recebem um peso maior. Tal modelo possui restrições de ciclo, velocidade, mudança de velocidade entre trechos, tempo de espera para virar a esquerda, tempo de espera do pelotão em semáforos e positividade, sendo o modelo um problema de programação linear inteira mista, tornado possível sua solução por um método exato como o *Branch and Bound* ou a utilização de alguma heurística para malhas grandes. Segundo os autores JONER e VOLPI (2013) a implementação da sincronização dos semáforos em Curitiba possibilitaria aumento da velocidade dos veículos, o que reduziria o tempo

de deslocamento dos motoristas, conforme já se constou em outras cidades que aderiram a sincronização dos seus semáforos.

No trabalho de PARAVISI, MUSSE e BICHO (2006), foi adotado um método computacional para simular o comportamento do tráfego de veículos utilizando uma escala microscópica, a qual analisa os veículos de forma individual, permitindo assim analisar alguns padrões de comportamento entre veículos consecutivos. Sendo considerado um método baseado na física que envolve a desaceleração para evitar colisões, utilização de redes de Petri para sincronização dos semáforos, o conceito de *blackboard* para tomadas de decisões entre os motoristas, onde as vias possibilitam aos mesmos uma ideia de posição e velocidade do veículo logo a frente, e uma política de acesso a cruzamentos baseado em reservas.

No artigo de HEINEN et al. (2013) utilizaram redes neurais artificiais (RNA) para fazer o controle dos tempos dos semáforos de uma malha viária, onde através da aplicação de um aprendizado por reforço a rede neural aprende o comportamento do tráfego veicular e assim, apresenta novos tempos semaforicos para evitar congestionamentos. Um simulador baseado em um modelo microscópico, denominado como ITSUMO, foi utilizado para descrever a topologia de dois modelos de malha, bem como simular e analisar o comportamento do tráfego com o controle da RNA em semáforos que atuam em tempo real. Com isso, os testes demonstraram redução de até 40% no número médio de veículos retidos nos semáforos quando se comparou com uma programação de tempo fixo nos semáforos da rede viária.

Em um trabalho desenvolvido por PAIVA, CASTRO e LOUREIRO (2003) foi realizado um estudo de caso, onde foi comparado o tamanho da banda de passagem, através do atraso veicular e a qualidade da progressão dos pelotões considerando a programação de semáforos com tempo fixo e com tempo real para períodos de baixo fluxo veicular. As análises foram realizadas com a utilização do sistema SCOOT, sendo implementado parte da malha viária da cidade de Fortaleza – CE com uso de dados reais do tráfego. Nos testes realizados pelos autores constou-se uma redução média de aproximadamente 40% no somatório dos atrasos veiculares e um aumento na qualidade da progressão de 20% em média, com o uso da programação de tempo fixo em relação a programação semaforica em tempo real para períodos de baixa demanda.

2.2 MODOS DE OPERAÇÃO DOS CONTROLADORES SEMAFÓRICOS NO BRASIL

Segundo ROMANI (2017), entende-se por Controlador Semafórico o equipamento eletroeletrônico responsável pela correta coordenação da sinalização semafórica em cruzamentos de vias urbanas de tráfego veicular e travessias sinalizadas para pedestres.

Portanto, as programações de tempo e sincronização dos semáforos são executadas pelos controladores semafóricos, onde estes, segundo os autores ARAÚJO (2006), PIETRANTONIO e BONETTI (2003), NETO (2012) e ROMANI (2017), podem funcionar nos seguintes modos de operação:

2.2.1 Modo intermitente

Este modo de operação coloca todos os grupos focais veiculares da interseção na cor amarela e os de pedestres são desligados ou ficam em vermelho intermitente, dependendo da programação adotada. Tal modo pode ser solicitado quando houver:

1. Requisição através da chave no painel de facilidades.
2. Falha do controle por hardware ou software.
3. Detecção de verdes conflitantes.
4. Requisição através um horário pré-programado.
5. Requisição externa através de comando da central.

A frequência de intermitência é de 1 Hz, sendo 0,5s de lâmpada acesa e 0,5s de lâmpada apagada.

2.2.2 Modo centralizado

Neste modo de funcionamento os controladores semafóricos estão interligados fisicamente por cabos a um sistema de controle, em um computador central, instalado na sala da Central de Controle Operacional (CCO). Portanto os

operadores da (CCO) podem receber e fornecer informações aos controladores, possibilitando assim alterar a programação dos controladores, adaptando os tempos semaforicos para situações inesperadas.

2.2.3 Modo manual

Neste modo as mudanças de estágio são condicionadas ao operador, e todas as condições de segurança, previamente programados no controlador são respeitadas. Após a finalização deste modo manual, os controladores voltam em sua programação normal automaticamente de acordo com sua tabela de entrada de planos ou em modo centralizado que depende do tipo de controle para o horário vigente.

2.2.4 Modo fixo

O controlador opera através de uma programação interna, mantendo tempos de verde fixos especificados em cada plano de tráfego. No caso de redes de controle a tempos fixos em modo local, os relógios da rede possuem uma referência única de tempo e todos os relógios dos controladores na rede são sincronizados através de uma rede de comunicação.

Portanto, o controle a tempos fixos permite a coordenação na rede de controladores semaforicos que podem operar em modo local ou centralizado.

Os controladores que operam neste modo possuem, o tempo de cada ciclo constante, assim como a duração e os instantes de mudança dos estágios das luzes dos semáforos. Isso significa que independentemente do volume de tráfego, os tempos de verde e vermelho serão sempre os mesmos e sua duração é calculada em função do volume de tráfego médio na interseção. O tempo de ciclo, a duração e a mudança dos estágios constituem um conjunto de parâmetros denominados de planos de tráfego. E os controladores atuais permitem armazenar vários planos de tráfegos, sendo cada plano implementado para diferentes períodos do dia, onde estes se caracterizam por um conjunto fixo de tempos.

De acordo com ROMANI (2017), os controladores de modo fixo recentes, oferecem as seguintes possibilidades:

a) Armazenamento, no mínimo, de 8 planos de tráfego independentes, e mais um plano no qual todos os grupos focais operam em modo intermitente.

b) Armazenamento independente de, no mínimo, 50 eventos de mudanças de planos através de uma tabela de horários, cada um podendo ser programado em dia da semana, hora, minuto e segundo (dependendo do processo de programação das defasagens).

E tais controladores podem ser programados com, pelo menos, os seguintes parâmetros:

- Tempo de verde (por fase e por plano): 1s a 120s, com passos de 1s.
- Tempo de verde mínimo de segurança (por fase): 1s a 25s, com passos de 1s.
- Tempo de amarelo (por fase): 3s a 10s em passos de 1s.
- Tempo de Vermelho Geral de Segurança (por fase): zero segundo a 10s, com passos de 1s.
- Tempo de defasagem (por plano e quando necessário): zero segundo a tempo de ciclo estipulado com passos de 1s.
- Tempo de vermelho piscante para pedestres (por fase de pedestre): 1s a 30s, com passos de 1s.

2.2.5 Modo atuado

Neste modo os tempos de controle para cada uma das fases é variável de acordo com a demanda de tráfego. O controlador analisa as condições de extensão de verde de acordo com o tempo entre os veículos trafegando na via. Este espaçamento entre veículos é medido por sensores sob o pavimento e a partir desta informação o controlador estende ou interrompe o tempo de verde de uma fase. O tempo de verificação de demanda nos detectores podem ser menores ou iguais a 1 segundo. O controlador a cada passo de 1 segundo verifica se há demanda, se houver, então uma nova extensão de verde entra em atuação até que os

espaçamentos entre os veículos sejam maiores que a distância máxima programada, então, neste caso o tempo de verde é interrompido.

O Modo Atuado pode ser, **Total** ou **Semi-Atuado**, onde o **Atuado Total** é quando todas as aproximações operam no modo atuado e o **Semi-Atuado** pode ser considerado quando todas as fases estão em modo atuado, mas o tempo de ciclo é constante. Portanto, a forma mais clássica da Semi-Atuação é quando se utiliza botoeiras de pedestres como atuação em uma rede de controle. E outra forma é quando pelo menos uma das fases está em modo atuado e as outras não, mas para este caso o tempo de ciclo permanece constante.

De acordo com ROMANI (2017), no Modo Atuado os controladores recentes podem ser programados com, pelo menos, os seguintes parâmetros:

- Tempo de verde (por fase e por plano): 1s a 120s, com passos de 1s.
- Tempo de verde mínimo de segurança (por fase): 1s a 25s, com passos de 1s.
- Tempo de verde máximo (por fase e por plano): 1s a 120 s, com passos de 1s.
- Tempo de Extensão de verde (por fase e por plano): 1s a 10s, com passos de 0,1s.
- Tempo de amarelo (por fase): 3s a 10s, com passos de 1s.
- Tempo de defasagem (por plano e quando necessário): zero segundo a tempo de ciclo, com passos de 1s.
- Tempo de Vermelho Geral de Segurança (por fase): zero segundo a 10s, com passos de 1s.
- Tempo de vermelho piscante para pedestres (por fase de pedestre): 1s a 30s, com passos de 1s.

2.3 TIPOS DE ESCALAS

O fluxo veicular pode ser abordado segundo GRAMANI (2010), CRISTIANI, PICCOLI e TOSIN (2011), em três escalas: microscópica, macroscópica e cinética, as quais são descritas na sequência.

2.3.1 Escala microscópica

A descrição microscópica refere-se à identificação individual dos veículos, conforme é ilustrado na figura (2.1). Neste caso, a posição e a velocidade de cada veículo definem o estado do sistema como variáveis dependentes do tempo.

FIGURA 2.1 – AVENIDA 23 DE MAIO NO ESTADO DE SÃO PAULO



FONTE: FOLHA DE S. PAULO (2018).

Um dos modelos matemáticos consiste na determinação de uma equação diferencial para a dinâmica de cada veículo, baseada na mecânica newtoniana, sob a ação dos veículos circundantes.

Pela equação horária do espaço, tem-se que a cinemática de um veículo para condições iniciais denotadas por x_0 , t_0 e v é:

$$x(t) = x_0 + \int_{t_0}^t v(t) dt + \iint_{t_0}^t a(t) dt dt, \quad (2.1)$$

onde:

$x(t)$: distância em função do tempo;

$v(t)$: velocidade como função do tempo;

$a(t)$: aceleração em função do tempo.

Assim a solução do sistema de equações diferenciais ordinárias fornece a descrição do escoamento de veículos na estrada. Mas, para descrever de forma satisfatória o fluxo real, o número total de veículos considerados deve ser muito grande e isso implica na complexidade computacional.

2.3.2 Escala macroscópica

Fisicamente, o tráfego é observado a uma longa distância, o comportamento individual dos veículos é ignorado e o fluxo pode ser associado a um escoamento de um fluido.

A descrição macroscópica refere-se à representação do estado do sistema por uma média de quantidades (quantidades macroscópicas), ou seja, a densidade, o momento linear e a energia, são consideradas como variáveis dependentes do tempo t e do espaço x .

A modelagem é análoga a da hidrodinâmica, com base nas equações de conservação e a modelos fenomenológicos. O fluxo de tráfego é modelado como um fluido compressível formado por veículos e descrito por uma estrutura macroscópica considerando que, a estrada poderá acumular um número máximo de veículos e os veículos têm uma velocidade média máxima. Portanto, tem-se as seguintes considerações:

a) Densidade de veículos

Segundo GRAMANI (2010) a densidade é igual a zero quando não se tem veículos na estrada e aumenta conforme o número de veículos aumenta. A densidade máxima (ρ_{max}) é alcançada quando os veículos não podem mais se mover, que é o caso de congestionamento ou engarrafamento. Portanto a densidade de veículos é denotada por:

$$\rho = \rho(t, x) \in [0, \rho_{max}], \quad (2.3)$$

b) Velocidade máxima média dos veículos

Segundo GRAMANI (2010) a velocidade máxima média dos veículos é denotada por v_{max} e atinge o seu máximo no caso de fluxo livre. Entretanto, um veículo veloz e isolado poderá atingir uma velocidade maior do que v_{max} . Experiências mostram que a velocidade limite v_L é afetada por vários fatores, entre eles, a qualidade da estrada e dos veículos. Em particular, a velocidade limite pode ser definida como:

$$v_L = (1 + \mu)v_{max}, \quad \mu > 0. \quad (2.4)$$

onde:

μ : parâmetro fenomenológico que depende do tipo de estrada e tipo de veículos que nela transitam, ou seja, se são veículos lentos ou rápidos.

c) Fluxo de veículos

O fluxo de veículos é denotado por $q = q(t, x) \in [0, q_{max}]$. O fluxo é igual a zero quando não se tem veículos na estrada ou quando a estrada comporta a sua capacidade máxima e os veículos não se movem. A densidade (ρ), a velocidade (v) e o fluxo (q) são as variáveis fundamentais do fluxo de tráfego cuja relação fundamental é:

$$q = f(v, \rho) = v\rho. \quad (2.5)$$

A modelagem é baseada nas equações de conservação e os modelos matemáticos são representados por equações diferenciais parciais não-lineares. As leis de conservação referentes as quantidades macroscópicas são:

Conservação de massa:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0, \quad (2.6)$$

e o equilíbrio do momento:

$$\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} = Q, \quad (2.7)$$

onde:

$$\rho = \rho(t, x) \in [0,1];$$

$$v = v(t, x) \in [0,1];$$

Q : a aceleração referida aos veículos dentro de um volume elementar.

Pode-se ter fechamento da equação (2.7), ou seja, validação da igualdade, por propriedades do fluxo local (equação analítica) ou fechamento por uma equação de evolução para a velocidade (equação funcional).

2.3.3 Escala cinética

A escala cinética, utilizada quando o estado do sistema ainda é identificado pela posição e pela velocidade dos veículos, no entanto esta identificação, não se refere a cada veículo, mas a distribuição de probabilidade adequada ao longo do estado microscópico considerada como uma variável aleatória.

Os modelos descrevem a evolução da função de distribuição por equações não-lineares integro-diferenciais com uma estrutura semelhante a equação de Boltzmann que descreve a dinâmica dos gases (KREMER, 2005).

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} + \frac{F \partial f}{\partial v} = C, \quad (2.8)$$

onde:

C : termo que representa o tratamento das colisões no processo considerado;

f : função de distribuição de uma partícula;

F : força;

t : tempo;

v : velocidade média das partículas.

2.4 MÉTODOS DE COORDENAÇÃO DE SEMÁFOROS

Visando minimizar os problemas de tráfego nas cidades, em 1953, Wardrop, Hillier, Whiting e Webster, deram início aos primeiros trabalhos de simulações digitais

para configurações de semáforos isolados na Inglaterra. E mais tarde diversos *softwares* e algoritmos foram desenvolvidos, tais como o sistema SIRI (VILANOVA, 2005), ATEFI (PIAI; CERVANTES, 2010), SCOOT e TRANSYT (ROBERTSON, 1986), sendo este último o mais citado na literatura. A seguir é apresentado em ordem cronológica os principais conceitos desses sistemas.

2.4.1 TRANSYT

O nome do sistema TRANSYT é uma abreviação de *Traffic Network Study Tool*. Este sistema trata de um algoritmo baseado em um modelo macroscópico determinístico que possui como objetivo minimizar a soma das filas médias de veículos em vias que possuem semáforos que operam em modo de tempo fixo. Esse objetivo é expresso como um índice de desempenho (*Performance Index - PI*), que pode ser traduzido da sua significância física dos veículos em termos financeiros. (ROBERTSON, 1986).

O programa TRANSYT foi desenvolvido em 1967 pelo Laboratório de Pesquisa de Transportes (*Transport Research Laboratory - TRL*) do Reino Unido, e programado usando o montador (assembler), sendo testado pela primeira vez nesse mesmo ano na Cromwell Road, em Londres. E deste então o TRANSYT tem passado por várias melhorias e aperfeiçoamentos, chegando na versão TRANSYT-7F em 1978 onde foi "americanizada" para a Administração Federal de Rodovias (FHWA) sob o projeto do Centro de Pesquisa de Transporte da Universidade da Flórida (DUTRA, 2005). Posteriormente novas versões foram surgindo, sendo as últimas versões descritas a seguir:

- TRANSYT 12 - Esta versão de 2003 introduziu a primeira representação gráfica da rede e permitiu que muitos recursos de rede fossem construídos graficamente usando o Network Construction Editor (TRLSoftware, 2018a).
- TRANSYT 13 - Esta versão de 2008 introduziu o modelo de transmissão celular como um modelo de tráfego alternativo capaz de modelar os efeitos de bloqueio dos links a jusante, bem como uma interface de usuário gráfica totalmente integrada e muitos outros novos recursos significativos. Em uma atualização desta versão foi introduzido um método revisado de inserir dados de fluxo de tráfego na forma de uma Matriz OD (Origem Destino), bem como vários outros

aprimoramentos, como a introdução de um Diagrama de Distância de Tempo (TRLSoftware, 2018b).

- TRANSYT 14 - Esta versão de 2010 possui funcionalidades específicas para avaliação fácil de junções controladas por sinais isolados e redes sinalizadoras e sinalizadas, além de suas funções tradicionais. Como sua atualização em 2011 foi introduzido um modelo de atribuição de equilíbrio jornada-tempo (TRLSoftware, 2018c).
- TRANSYT 15 - O lançamento atual do TRANSYT 15 é a versão 15.5.1, que foi lançado em setembro de 2017, amplia a capacidade do TRANSYT em muitas direções; incluindo a sua utilização para o projeto de junções com a adição do diagrama de fluxo da nova linha, sob a forma de um conjunto abrangente de sobreposições de redes de fluxo de novas pistas. E mais uma série de melhorias, reparos e aperfeiçoamentos (TRLSoftware, 2018d).

Segundo Vincent *et al.* (1980) o programa TRANSYT é composto de um modelo de simulação de tráfego e de um algoritmo de minimização, cuja função objetivo considera o peso combinado do número de paradas e o atraso sofrido pelos veículos em todas as aproximações da rede semaforizada. Este valor representa o custo do congestionamento, sendo denominado de Índice de Desempenho (PI), onde este é definido por:

$$PI = \sum_{i=1}^n (W \cdot w_i \cdot d_i + \frac{K}{100} k_i \cdot S_i) q_i, \quad (2.9)$$

onde:

W : custo médio por unidade de atraso;

w_i : coeficiente de ponderação de atraso no trecho i ;

d_i : atraso médio no trecho i ;

K : custo médio para 100 unidades de parada;

k_i : coeficiente de ponderação do número de paradas no trecho i ;

S_i : número médio de paradas no trecho i ;

q_i : fluxo médio no trecho i ;

n : número de trechos na rede.

A base do modelo de simulação é a representação dos “Perfis Cíclicos de Fluxo” (PCF), que corresponde a quantidade média de veículos, ou unidades de carros de passeio (ucp) que chegam em cada linha de retenção da interseção da malha em pequenos intervalos de tempo do ciclo. Os intervalos de tempo do TRANSYT podem ser fixados de 1 a 6 segundos, onde esses intervalos do ciclo são denominados de passos (VINCENT *et al.*, 1980).

Para os cálculos de simulação do tráfego são considerados três tipos de perfis de fluxo para cada trecho da malha, sendo definidos em: *IN*, *GO* e *OUT*.

IN - Representa o perfil de fluxo que chegaria à linha de retenção da interseção, caso estes veículos não fossem detidos na linha de retenção pela indicação semafórica.

GO - Representa o perfil do fluxo de escoamento do tráfego de um trecho para seu subsequente em condições de saturação no período de verde. Este perfil de fluxo é o valor máximo de fluxo, ou seja, igual ao fluxo de saturação.

OUT - Representa o perfil do fluxo que deixa um trecho.

O perfil do fluxo que chega (*IN*) em cada trecho é definido pela combinação dos fluxos de saída (*OUT*) atual e chegada (*IN*) do passo anterior, e por um fator de suavização que leva em conta o “Tempo de Cruzeiro”, sendo este o tempo médio que o veículo leva para percorrer, sem atraso, um trecho. Portanto, obtém-se a seguinte equação (2.10):

$$q_k^1 = f \cdot q_k \cdot p + (1 - f) \cdot q_{(k-1)+t}^1, \quad (2.10)$$

onde:

k : um determinado passo de tempo;

$(k - 1)$: um passo de tempo anterior ao passo k ;

q_k^1 : perfil do fluxo de chegada (*IN*) no passo k ;

q_k : perfil do fluxo de saída (*OUT*) no passo k ;

p : proporção do fluxo de saída (*OUT*) que alimenta o trecho;

t : corresponde a 0,8 vezes o “Tempo de Cruzeiro” médio para o trecho, onde este tempo é dado em passos;

f : fator de suavização.

O fator de suavização (f) é obtido pela equação (2.11), mas na versão 8 do TRANSYT, onde o modelo tem o potencial de realizar a priorização de ônibus, o fator (f) passou a ser calculado pela equação (2.12).

$$f = \frac{1}{1 + 0,35 \cdot t}, \quad (2.11)$$

$$f = \frac{1}{1 + 0,7 \cdot b + 0,3 \cdot t}, \quad (2.12)$$

onde:

b : tempo médio parado num ponto de ônibus [em passos].

O número de veículos que ficam retidos na linha de retenção devido o surgimento da sinalização vermelha é obtido pela seguinte equação (2.13):

$$m_k = m_{k-1} + q_k^1 - s_k \quad \text{ou} \quad m_k = 0, \text{ se } m_k < 0, \quad (2.13)$$

onde:

m_{k-1} : número de veículos retidos no passo anterior;

s_k : número máximo de veículos deixando a interseção no passo k , sendo este o perfil de descarga.

No modelo, o atraso total da rede é calculado pela soma das parcelas de atraso sofridas pelas aproximações da rede semaforizada em questão. O TRANSYT calcula o atraso uniforme, aleatório e ainda o atraso devido à supersaturação, que corresponde aos casos nos quais a demanda maior que a capacidade do trecho. Já o “atraso aleatório” corresponde as variações do fluxo de chegada a cada ciclo. Esses atrasos são calculados como uma taxa média e pode ser considerado como a fila média no início do sinal vermelho. Na equação (2.14) a seguir observa-se o cálculo dos atrasos aleatórios e supersaturados.

$$A_{aleatório} + A_{supersaturado} = \frac{T}{4} \left(\sqrt{(q - s)^2 + \frac{4q}{T}} + (q - s) \right), \quad (2.14)$$

onde:

q : fluxo médio de chegada (I/N) [ucp/hora];

s : fluxo máximo de descarga (GO) [ucp/hora];

T : duração do período para a qual os tempos semafóricos foram considerados [horas];

A quantidade média de paradas é dada pela razão entre a soma dos atrasos e o valor médio de (ucp's) que são descarregados do trecho considerado durante cada intervalo de verde.

O TRANSYT realiza a otimização do seu modelo por meio de um procedimento *hill climbing* que imita a maneira como os engenheiros de tráfego buscam bons tempos. Neste processo, o TRANSYT calcula o PI para a rede em um plano semafórico inicial. Posteriormente, o programa altera a defasagem de uma das interseções um certo número de tempo em passos e recalcula o PI da rede, caso este novo valor seja menor, então a defasagem é sucessivamente alterada na mesma direção e com o mesmo número de passos até obter um valor mínimo para o PI, mas caso o valor do PI obtido após a primeira alteração de defasagem seja maior, então a alteração caminha na direção oposta até que um valor menor seja encontrado. Além disso, o TRANSYT também realiza modificações individuais nas repartições de verde de cada uma das interseções para buscar melhores valores para o PI (VINCENT *et al.*, 1980).

A administração da Rodovia Federal onde o TRANSYT foi testado salientou que o uso desse sistema para otimizar a coordenação é altamente econômico apenas em termos de economia de energia. Portanto, com o passar dos anos os planos semafóricos podem ficar mal adaptados ao tráfego presente, caso não haja atualizações nos planos semafóricos de tempo fixo em que o TRANSYT e todos os métodos "off-line" são baseados. Devido às limitações do "tempo fixo" surgiu o sistema SCOOT para programação semafórica em tempo real, mas se basendo nos princípios do sistema TRANSYT.

2.4.2 SCOOT

O nome do sistema SCOOT é uma abreviação de *Split, Cycle and Offset Optimisation Technique*. Assim como no TRANSYT, este algoritmo possui como objetivo minimizar a soma das filas médias de veículos em vias que possuem semáforos que operam em modo atuado na área da malha. Esse objetivo é expresso como um índice de desempenho (PI). A metodologia adotada é realizar ajustes nos tempos dos sinais em incrementos pequenos e frequentes para corresponder à situação de tráfego mais recente.

O SCOOT foi desenvolvido em 1973 pelo TRL para superar as dificuldades encontradas em Glasgow e em outros lugares. O sistema SCOOT começou a ser aplicado como versão de teste em 1975 na cidade de Glasgow e foi passando por melhorias até 1979, onde nesta data passou a ser aplicado também na cidade de Coventry (ROBERTSON, 1986). Desde então o SCOOT tem conquistado confiança, e vem sendo aplicado em várias outras cidades do mundo. O sistema tem passado por várias versões ao longo dos anos, sendo algumas delas descritas a seguir:

- Versão 3.1- Incluiu prioridade de ônibus, instalações de banco de dados e detecção de incidentes (BRETHERTON; BOWEN, 1996).
- Versão 4.2 - Adicionou estimativas de emissão de poluentes (BRETHERTON; WOOD; BOWEN, 1998).
- Versão 4.5 - Permitiu sete níveis de prioridade diferentes para ônibus (SIEMENS, 2018).

Depois da versão 4.5 outras versões surgiram e atualmente encontra-se na versão 6.1 (MMX SP1), onde é possível especificar um nível máximo de saturação para um trecho ou uma série de trechos e SCOOT tentará manter o grau de saturação igual ou abaixo do nível especificado (SIEMENS, 2018).

De acordo com ROBERTSON (1986), o desenvolvedor do método, SCOOT se baseia em três princípios:

- **Medir os perfis de fluxo cíclico (PFC):** O PFC é uma medida do fluxo médio de um trecho da via durante cada intervalo do tempo de ciclo do sinal a montante. O tempo de ciclo é dividido em subintervalos de tempo de 4

segundos. Os padrões de perfil tendem a ser repetidos e acoplados a novos dados em uma sequência cíclica para evitar grandes flutuações aleatórias no perfil.

- **Modelo de tráfego on-line:** É um modelo de tráfego que funciona em tempo real. O SCOOT consegue isso levando em conta os dados do PFCS e calculando as filas atuais em todas as vias da malha, onde tais estimativas de fila são necessárias para o otimizador.
- **Otimização incremental:** É um plano de coordenação capaz de responder as novas situações de tráfego em uma série de incrementos frequentes, mas pequenos. SCOOT, portanto, usa um plano de coordenação flexível que pode ser prolongado ou abreviado para combinar com a situação mais recente registrada pelo PFCS. Tais informações alimentam o sistema SCOOT que possui três otimizadores, sendo eles responsáveis para se determinar: a repartição de verde (*Split*), o tempo de ciclo (*Cycle length*) e o tempo entre os semáforos adjacentes (*Offset*).
 - **Otimizador da repartição de verde:** Nesta otimização, alguns segundos antes de todas as mudanças de fase, o otimizador SCOOT calcula se é melhor avançar ou retardar a mudança agendada em até 4 segundos, ou para deixá-lo inalterado, assim obtém-se um melhor balanceamento das saturações nas aproximações da interseção, mas cada interseção é tratada de forma individual. A decisão de alterar ou não o tempo da fase verde tem o propósito de minimizar o maior grau de saturação nos trechos. Essa modificação no tempo de verde, apesar de implementada no estágio do ciclo atual, não é levada integralmente ao próximo ciclo. Dessa forma, espera-se que o SCOOT reaja às variações aleatórias que ocorrem de ciclo para ciclo, e uma vez que acompanha a tendência de variação da demanda ao longo do tempo. (LOUREIRO; LEANDRO; GOMES, 2018).
 - **Otimizador do tempo de ciclo:** Da mesma forma, o tempo de ciclo de um grupo de junções podem ser incrementadas para alguns segundos a menos ou mais. Então alterações favoráveis de divisão e compensação são implementadas imediatamente. Para se coordenar a operação de interseções em uma subárea, é preciso que estas

estejam operando sob um mesmo tempo de ciclo, ou múltiplo deste, de modo que se garanta a manutenção das defasagens. A cada 5 minutos o sistema SCOOT analisa se é necessário realizar mudança no tempo de ciclo, para cada interseção da subárea, de maneira que o grau de saturação especificado seja atendido, onde normalmente o grau de saturação adotado é de 90%. Usando o maior dos ciclos encontrados, o SCOOT verifica se a implementação desse novo tempo de ciclo trará uma redução igual ou maior que 2% nos atrasos experimentados na subárea controlada. Caso afirmativo, a modificação é implementada e então, esta otimização passa a acontecer a cada 2,5 minutos, onde os tempos de ciclos podem ser alterados em valores múltiplos de 4 segundos. (LOUREIRO; LEANDRO; GOMES, 2018).

- **Otimizador de defasagens:** Quanto ao otimizador de defasagens, este atua a cada ciclo, verificando se modificações da ordem de 4 segundos na programação trarão benefícios à progressão do fluxo veicular modelados pelo SCOOT. A decisão de realizar alguma alteração no tempo de defasagem é feita comparando-se a soma dos PIs nos trechos de entrada e saída de cada interseção sob a programação semafórica atual, com aquelas obtidas com defasagens de 4 segundos a mais ou a menos e aquela que fornece a menor soma de PI passa a ser aplicada (LOUREIRO; LEANDRO; GOMES, 2007).

Concluiu-se que o SCOOT reduziu os comprimentos médios da fila em cerca de 12% e foi mais eficaz onde o tráfego estava congestionado e os fluxos eram variáveis. Portanto, os benefícios podem ser ainda maiores ao se comparar com planos semafóricos de tempo fixo quando estes se encontram desatualizados por alguns anos, podendo assim o SCOOT reduzir os congestionamentos em pelo menos 20%. As conclusões gerais sobre a eficácia do SCOOT nasceram por tentativas subsequentes em Southampton e Londres (ROBERTSON, 1986).

2.4.3 SIRI

O sistema SIRI trata-se de um algoritmo baseado em um modelo mesoscópico cuja função é calcular o tempo de ciclo, tempos de verde e defasagens de uma rede

de semáforos. E possui como objetivo minimizar o atraso e o número de paradas de veículos em vias que possuem semáforos que operam em modo de tempo fixo.

O sistema SIRI começou a ser desenvolvido em 2000 pela CET de São Paulo, e sua primeira versão foi concluída no início de 2004. Atualmente o software encontra-se na versão 2.2, que incorpora vários recursos apontados pelos técnicos que acompanharam a sua implementação desde sua primeira versão. As melhorias realizadas foram na interface com o usuário e em alterações significativas no próprio algoritmo. Atualmente, existem cerca de 250 semáforos, em São Paulo, que operam segundo as programações semafóricas calculadas por este sistema (VILANOVA, 2008).

Segundo VILANOVA (2008) o programa SIRI é composto de um modelo de simulação que respeita as seguintes condições impostas pelo usuário: o tempo de ciclo máximo, os tempos de verde de segurança dos grupos semafóricos e os tempos de entreverdes de tráfego. Os veículos são agrupados em pelotões, assim como no TRANSYT, sendo a velocidade, a aceleração e a posição enxergadas não de forma individual, mas de maneira coletiva do pelotão.

O sistema busca calcular os melhores tempos de verde, defasagens e tempo de ciclo, de forma que minimize o tempo de espera e o número de paradas dos veículos que trafegam na rede, onde esta função objetivo é denominada como Índice de Desempenho (ID), a qual é apresentada a seguir pela equação (2.15).

$$ID = \sum_{i=1}^n (A_i + K_p \cdot NP_i), \quad (2.15)$$

onde:

ID: Índice de Desempenho da rede;

n: número total de trechos da rede;

A_i: Atraso no trecho *i*;

K_p: peso por parada;

NP_i: Número de Paradas no trecho *i*.

O parâmetro *K_p* traduz a relação de importância entre o prejuízo causado por uma parada e o prejuízo causado por um segundo de atraso. O valor adotado para este parâmetro é geralmente de 30 segundos (VILANOVA, 2008).

Segundo VILANOVA (2008) a estratégia fundamental da simulação do SIRI é utilizar passos de um segundo. A respeito da distância entre semáforos é considerado finito, assim, leva-se em conta as situações de congestionamento. A simulação ocorre em duas etapas:

- A primeira tem duração igual a oito vezes o tempo de ciclo calculado e tem a função de permitir que todos os veículos que estejam chegando na rede possam preencher todos os trechos internos de forma que o processo alcance uma distribuição de fluxos estável.
- Na segunda etapa são calculados os atrasos e as paradas para duas situações. Uma é quando a via não está congestionada, e outra quando está.

Nos trechos em que os veículos entram na rede, o programa adota um perfil de chegada uniforme. E nos trechos internos existem duas seções consideradas: uma seção no início do trecho e outra ao final deste. O perfil do fluxo no início da seção de um determinado trecho deriva do fluxo fornecido pelos trechos que contribuem para o trecho em questão. O perfil do fluxo no final da seção é igual ao perfil inicial da correspondente seção, mas com uma diferença no tempo, devido ao tempo de deslocamento para percorrer o trecho.

O sistema SIRI calcula a fila dos veículos em cada trecho e analisa a possibilidade de que os veículos de um trecho não possam realizar a travessia para o trecho seguinte devido à existência de um congestionamento. Permitindo assim calcular o número de paradas e o atraso em cada trecho de maneira mais realista.

Com o fornecimento dos tempos de amarelo e vermelho de limpeza para toda rede, o programa calcula três diferentes conjuntos de valores. O primeiro é o tempo de ciclo, o qual é adotado para todas as interseções da rede. O segundo conjunto são os tempos de verde e vermelho. E o terceiro conjunto é a defasagem entre os semáforos.

O tempo de ciclo da rede é calculado através do método “Grau de saturação”, de modo que as interseções mais críticas operem com um grau de saturação de aproximadamente 88%, semelhante ao sistema SCOOT. Em casos de congestionamento, o tempo de ciclo é dimensionado no seu valor máximo, que é de 120 segundos na programação padrão do SIRI, mas este valor máximo pode ser alterado pelo usuário.

Os tempos de verde são inicialmente calculados para os semáforos nos trechos mais críticos. Depois os tempos são calculados segundo um critério de equalização dos graus de saturação entre os trechos críticos.

O cálculo das defasagens é realizado inicialmente para uma rota analisada pelo programa como sendo a principal e posteriormente define das defasagens de cada uma das rotas secundárias.

Para minimizar a função objetivo foi utilizado um processo denominado de *hill-climbing* na primeira versão do sistema, mas a partir da segunda versão este processo foi substituído por um procedimento de varredura mais detalhada.

2.5 CÓDIGO DE TRÂNSITO BRASILEIRO

Segundo o Código de Trânsito Brasileiro em seu artigo 1º, parágrafo 1, o trânsito é definido por: “Considera-se trânsito a utilização das vias por pessoas, veículos e animais, isolados ou em grupos, conduzidos ou não, para fins de circulação, parada, estacionamento e operação de carga ou descarga” (BRASIL. Código de Trânsito Brasileiro, 1997).

Com relação ao semáforo, o Manual Brasileiro de Trânsito descreve a finalidade deste como sendo a de transmitir aos usuários a informação sobre o direito de passagem em intersecções e/ou seções de via onde o espaço viário é disputado por dois ou mais movimentos conflitantes. Os cruzamentos que possuem semáforos têm três estágios: aberto(verde), alerta (amarelo) e fechado (vermelho).

“Mesmo que a indicação luminosa do semáforo lhe seja favorável, nenhum condutor pode entrar em uma interseção se houver possibilidade de ser obrigado a imobilizar o veículo na área do cruzamento, obstruindo ou impedindo a passagem do trânsito transversal”, segundo o Código de Trânsito Brasileiro em seu artigo 45º (BRASIL. Código de Trânsito Brasileiro, 1997).

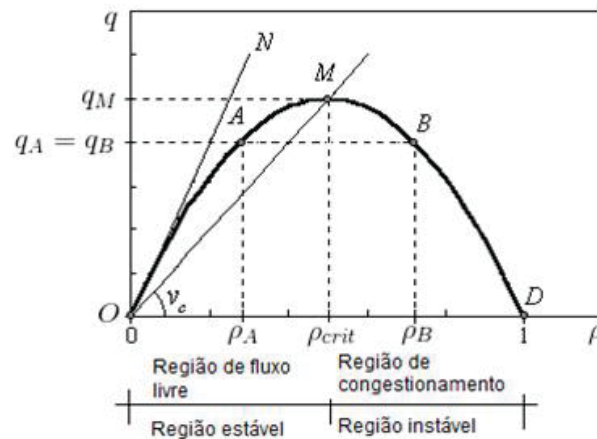
3 MODELOS E MÉTODOS

Para elaboração deste trabalho foi utilizado um modelo matemático baseado na escala macroscópica e os dados da SETRAN (Secretaria Municipal de Trânsito - PR) com a quantidade de veículos acumulados por hora nos trechos da Linha Verde que foram estudados, mas para melhorar a visualização dos dados e otimizar as simulações, utilizou-se uma distribuição uniforme para discretização dos dados com intervalos de tempo em segundos, os quais denominamos de “passo”. O tempo de ciclo dos semáforos no modelo é considerado como um dado. Após a definição do modelo matemático, foi elaborado um algoritmo para o cálculo da função a ser minimizada. Esta se refere à quantidade de veículos que ficam acumulados, o que depende das variáveis de decisão sendo, a quantidade de tempo que cada semáforo fica aberto ou fechado, em função de 4 períodos distintos do dia.

Para encontrar a solução que minimize a função objetivo do problema, foi usado um método heurístico de Busca em Vizinhança Variável (MLADENOVIC; HANSEN, 1997).

O fluxo do tráfego de veículos em uma rua pode ser considerado um fluido (GRAMANI, 2010). A analogia entre o fluxo de tráfego e a dinâmica de fluidos é possível através da representação macroscópica, fluidodinâmica, na qual o tráfego é modelado como um fluido compressível de veículos. Assim a quantidade de veículos na pista determina o fluxo, criando assim a relação fluxo versus densidade. Quando se tem o fluxo máximo há densidade crítica, se ainda assim houver aumento na densidade tem-se a diminuição do fluxo e a redução da velocidade que pode ocasionar o engarrafamento (ou congestionamento) como demonstrado na figura (3.1).

FIGURA 3.1 – CURVA PARABÓLICA DENSIDADE-FLUXO



FONTE: Adaptada de BONZANI e CUMIN (2009).

O ponto O como origem refere-se ao caso onde densidade e fluxo são nulos, o ponto D refere-se à densidade máxima cujo correspondente fluxo é nulo (neste caso temos o trânsito completamente parado). Por fim, o ponto M refere-se ao fluxo máximo correspondendo a densidade crítica (ou fluxo crítico).

Compara-se o fluxo de uma rua ao de um rio e aplicam-se os fundamentos de mecânica dos fluidos para solucionar o problema de densidade, que neste caso, seria o engarrafamento de veículos na rua.

Para minimizar esse engarrafamento usam-se os conceitos de propagações de vazões em reservatórios (PINTO et al. 1976).

Considere-se a vazão afluyente Q_a de um certo trecho do rio e uma vazão efluente dada por Q_e . Assim, a variação da vazão de uma seção do rio pode ser descrita pela equação da conservação de massa:

$$Q_a = Q_e + \frac{dV}{dt}, \quad (3.1)$$

sendo que dV representa a variação do volume acumulado no rio e dt a variação do tempo.

Trabalhou-se uma rua de mão única e considerou-se os veículos que entram na pista como uma vazão afluyente e os que saem dela sendo uma vazão efluente, o fluxo acumulado é o volume. E assim tem-se um sistema semelhante ao da propagação das vazões e pode-se utilizar a equação da conservação de massa para

medir o volume do trânsito em um determinado espaço de tempo. Faz-se uma adaptação da equação (3.1) para

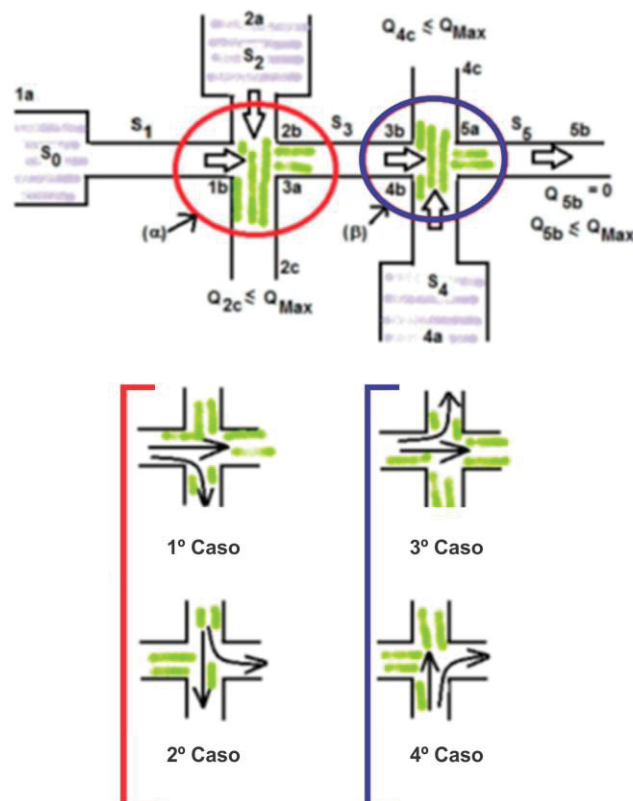
$$\frac{dv}{dt} = Q_a - Q_e, \quad (3.2)$$

que será usada no modelo.

3.1 MODELAGEM MATEMÁTICA

A modelagem foi realizada sobre uma malha viária, considerando esta com sentido único e com duas vias transversais, conforme as figuras (3.2).

FIGURA 3.2 – MODELAGEM DO TRECHO



FONTE: Adaptada de KAVISKI (2017).

As variáveis e parâmetros apresentados na figura (3.2) são:

S_i : volume de veículos no trecho $i = 0, \dots, 5$;

a : início do trecho;

b : final do trecho;

α_i : porcentagem de veículos que continuarão na via principal passando pelo semáforo i ;

β_i : porcentagem de veículos que vira na via principal passando pelo semáforo i ;

Q_{ia} : quantidade de veículos no início do trecho i ;

Q_{ib} : quantidade de veículos no final do trecho i .

Como tem-se duas transversais estuda-se quatro casos:

1° - Via principal aberta e 1ª via secundária fechada (S_1), onde uma parte de carros dobra à direita na via secundária e uma parte de carros continua na via principal, esta quantidade que continua na via principal é dada pela equação (3.3).

$$Q_{3a} = \alpha_1 Q_{1b}, \quad (3.3)$$

e a quantidade de carros da via secundária que entra a direita na via principal é nula, ou seja,

$$Q_{2b} = 0, \quad (3.4)$$

pois trata-se de uma conversão imprópria.

2° - Via principal fechada e 1ª via secundária aberta (S_2), onde a quantidade que carros da via principal que entra à esquerda na via secundária é nula, portanto:

$$Q_{1b} = 0, \quad (3.5)$$

pois o semáforo encontra-se fechado e neste caso os carros da via secundária podem seguir reto ou entrar à esquerda na via principal, onde a quantidade de veículos que acessão a via principal é dada por,

$$Q_{3a} = \beta_2 Q_{2b}. \quad (3.6)$$

3° - Via principal aberta e 2ª via secundária fechada (S_3), onde tem-se uma parte dos carros dobrando à esquerda na via secundária e outra parte que continua na via principal. A quantidade que continua na via principal é dada por,

$$Q_{5a} = \alpha_3 Q_{3b}. \quad (3.7)$$

E os veículos da via secundária que entram à esquerda na via principal é nulo, ou seja,

$$Q_{4b} = 0, \quad (3.8)$$

pois é contramão.

4° - Via principal fechada e 2ª via secundária aberta (S₄), onde a conversão dos veículos da via secundária que entram à esquerda na via principal é contramão, então a quantidade de veículos que realizam este movimento é nulo, portanto,

$$Q_{3b} = 0. \quad (3.9)$$

Sendo assim, os veículos que trafegam na via secundária podem seguir reto ou entrar à direita na via principal, onde a quantidade de automóveis que acessa a via principal é dada por,

$$Q_{5a} = \beta_4 Q_{4b}. \quad (3.10)$$

Em S₀ são considerados os veículos que entram na via principal e S₅ os que saem ao final da mesma.

Portanto, a quantidade de carros que entram na via principal é dada pelas equações:

$$Q_{3a} = \alpha_1 Q_{1b} + \alpha_2 Q_{2b} \quad (3.11)$$

e

$$Q_{5a} = \alpha_3 Q_{3b} + \beta_4 Q_{4b}. \quad (3.12)$$

Assim, o modelo matemático passa a ser descrito como:

Função objetivo: minimizar a quantidade de carros que ficam retidos no ciclo mais crítico dos semáforos para todos os trechos $S_i(t)$, ou seja,

$$\text{Min } Z = \sum_{i=0}^5 S_i(t),$$

sujeita as condições de conservação de massa e capacidade:

$$\frac{d(S_0(t) + S_1(t))}{dt} = Q_{1a}(t) - Q_{1b}(t). \quad (3.13)$$

$$\frac{d(S_i(t))}{dt} = Q_{ia}(t) - Q_{ib}(t), \text{ onde } i = 2, 3, 4 \text{ e } 5. \quad (3.14)$$

$$Q_{3a}(t) = \alpha_1 Q_{1b}(t) + \beta_2 Q_{2b}(t). \quad (3.15)$$

$$Q_{5a}(t) = \alpha_3 Q_{3b}(t) + \beta_4 Q_{4b}(t). \quad (3.16)$$

$$S_i(t) \leq S_{i,max}, \text{ onde } i = 1,3 \text{ e } 5. \quad (3.17)$$

As **variáveis** são $t \in N_*^{4 \times 3 \times 2}$, tal que, t é uma matriz tridimensional sendo, as linhas da matriz os 4 períodos de programação semafórica no dia, as colunas da matriz os 3 semáforos considerados e as 2 profundidades da matriz trata-se dos tempos de verde na primeira camada e os tempos de vermelho na segunda camada.

As restrições (3.13) e (3.14) são resultados da aplicação da equação (3.2) em cada trecho. $S_{i,max}$, representa a capacidade máxima que o trecho i pode comportar. Nota-se que neste modelo as restrições de capacidade são consideradas apenas para os trechos da via principal.

Nota-se que o modelo não leva em consideração o tempo de defasagem entre os semáforos, ou seja, não maximiza necessariamente a largura da banda de passagem.

Adotando-se o modelo matemático apresentado, elaborou-se um programa utilizando a linguagem de programação Pascal, para o cálculo da função a ser minimizada, que depende das variáveis de decisão do tempo que cada semáforo fica aberto e fechado para os 4 períodos do dia.

No algoritmo as restrições (3.13) e (3.14) referentes a conservação de massa foram calculadas da seguinte forma,

$$V_{i,p} = q_{i,p} - qs_{i,p} \quad (3.18)$$

onde:

$V_{i,p}$: volume de veículos no trecho i no tempo em passo p ;

$q_{i,p}$: quantidade de veículos no trecho i no tempo em passo p , a qual é obtida

a equação (3.19) a seguir:

$$q_{i,p} = \frac{qe_{i,(p-1)} + qe_{i,p} - qs_{i,(p-1)}}{2} + V_{i,(p-1)}, \quad (3.19)$$

onde:

$V_{i,(p-1)}$: volume no trecho i no tempo em passo $(p - 1)$;

$qe_{i,p}$: quantidade de veículos que entra no trecho i no tempo em passo p ;

$qe_{i,(p-1)}$: quantidade de veículos que entrou no trecho i no tempo em passo $(p - 1)$;

$qs_{i,(p-1)}$: quantidade de veículos que saiu no trecho i no tempo em passo $(p - 1)$;

$qs_{i,p}$: quantidade de veículos que saem no trecho i no tempo em passo p , a qual é determinada pela equação (3.20).

$$qs_{i,p} = \begin{cases} q_{i,p}, & \text{se } q_{i,p} < qf \\ qf, & \text{caso contrário} \end{cases} \quad (3.20)$$

onde:

qf : quantidade média de veículos que atravessa um cruzamento em um intervalo de tempo igual ao tamanho do passo, ou seja, trata-se do escoamento de veículos nos cruzamentos. Tal valor é obtido pela equação (3.25) definida na sequência, onde qf é dado por Qft .

3.2 ESTUDO DOS PARÂMETROS

A seguir é apresentado um estudo para se determinar os valores de alguns parâmetros utilizados.

3.2.1 Fluxo máximo

O fluxo máximo, ou seja, a quantidade de veículos que atravessa um cruzamento por passo de sinal verde aproveitado, pode ser calculado da seguinte forma:

$$F_{max,i} = n_i \cdot passo \cdot \frac{v_m}{3,6 \cdot C_i} \quad (3.21)$$

onde:

$F_{max,i}$: fluxo máximo saindo do trecho i [veíc./s];

v_m : velocidade média do fluxo de veículos nos trechos [km/h];

C_i : comprimento do cruzamento saindo do trecho i [m];

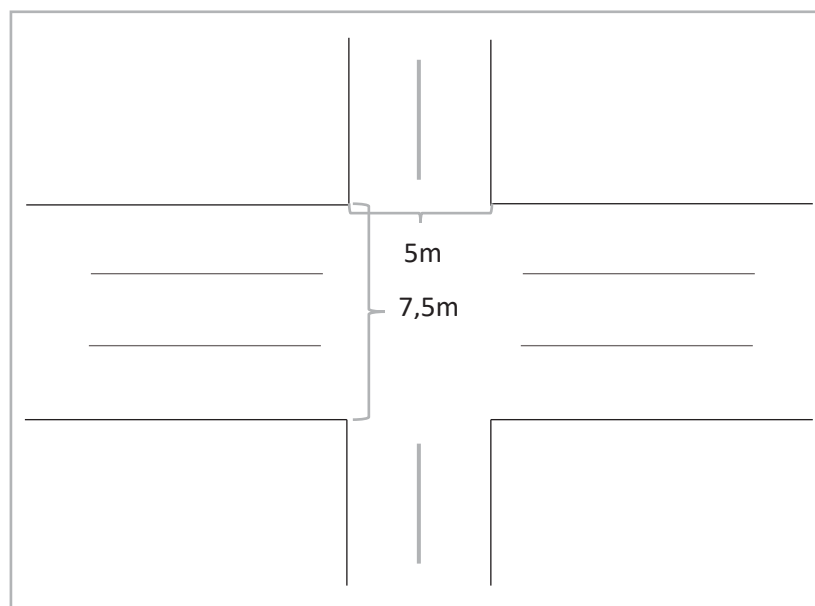
$passo$: intervalo de tempo o qual se realiza as simulações de tráfego [s];

n_i : número de pistas no trecho i .

O fluxo máximo é denominado por muitos autores como fluxo de saturação.

Observa-se que na equação (3.21) o parâmetro v_m é dividido por 3,6, o que transforma a velocidade de km/h em m/s, pois nas simulações o valor da velocidade média do fluxo é fornecido em km/h. E para o cálculo da equação são considerados que a via principal possui 3 pistas, 5m de cruzamento e suas transversais possuem 2 pistas, 7,5m de cruzamento, conforme a figura (3.3) a seguir:

FIGURA 3.3 - CARACTERÍSTICAS DO CRUZAMENTO



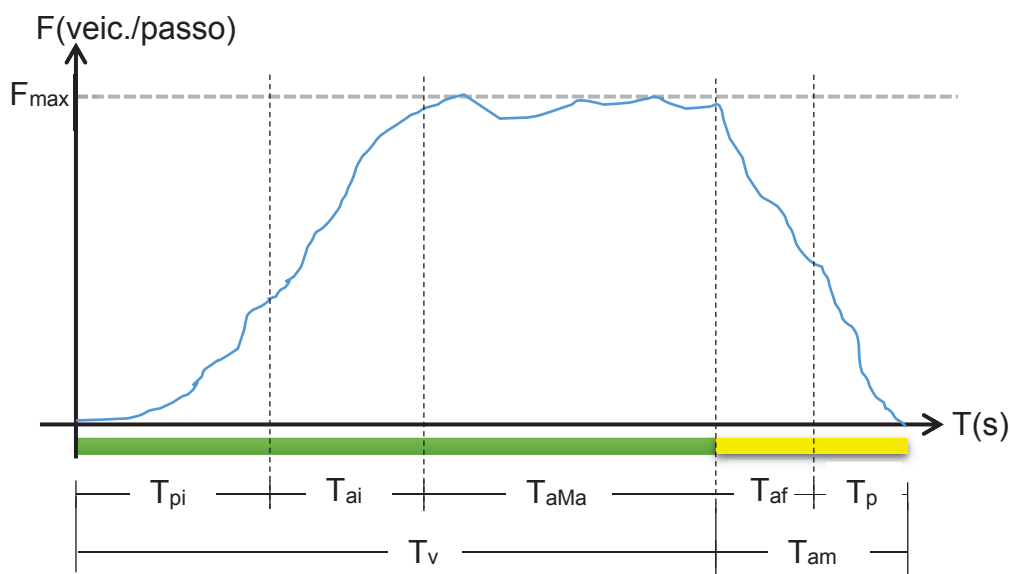
FONTE: O autor (2018).

Os parâmetros v_m e o $passo$ serão modificados para cada execução do programa para realizar uma futura análise de sensibilidades desses parâmetros.

3.2.2 Uniformização do fluxo

Para simplificar as simulações a serem realizadas será utilizada uma uniformização do fluxo veicular em cada estágio do ciclo, pois ao longo do tempo de verde e amarelo ocorre uma oscilação do fluxo de maneira similar com o gráfico da figura (3.4) a seguir:

FIGURA 3.4 – FUNÇÃO DE FLUXO VEICULAR REAL EM UMA VIA LIVRE



FONTE: O autor (2018).

Na figura (3.4) observa-se que nos primeiros segundos do intervalo T_{pi} o fluxo é zero, pois em cada início de verde ocorre um tempo perdido pelo fato que o motorista que se encontra parado nesta faixa de retenção demora alguns segundos para perceber que o semáforo ficou verde e realizar a aceleração do veículo. A fim de transformar a função da figura (3.4) em uma constante, nos intervalos T_{ai} e T_{af} o fluxo será considerado igual ao fluxo máximo, e para compensar o arredondamento, nos intervalos de tempo T_{pi} e T_{pf} o fluxo será considerado nulo. Portanto o tempo total de verde com máximo fluxo podem ser obtidos das seguintes formas:

$$T_{ta} = T_{ai} + T_{aMax} + T_{af} \quad (3.22)$$

ou

$$T_{ta} = T_v - T_{pi} + T_{af}, \quad (3.23)$$

onde:

T_{ta} : tempo total de aproveitamento;

T_{pi} : tempo perdido no início;

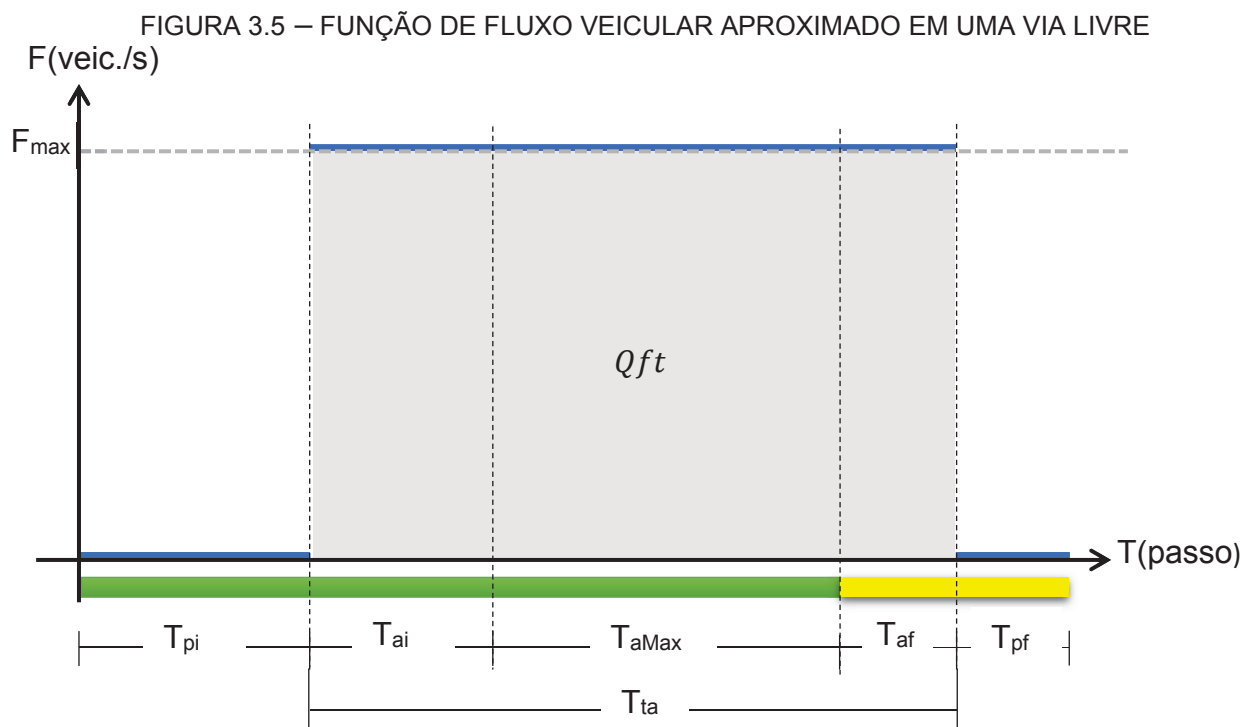
T_{ai} : tempo aproveitado no início;

T_{aMax} : tempo de máximo aproveitamento;

T_{af} : tempo aproveitado após o fim do verde;

T_v : tempo de verde.

Sendo assim, a função de fluxo torna-se constante com fluxo máximo no intervalo de tempo $T \in T_{ta}$ conforme apresentado na figura (3.5) a seguir:



FONTE: O autor (2018).

A função apresentada na figura (3.5) possibilita determinar mais facilmente a vazão de veículos, ou seja, a quantidade de veículos que podem trafegar na via para cada intervalo de verde e amarelo aproveitado, pois basta calcular a área do retângulo

em cinza da figura (3.5) que é equivalente a área abaixo da curva na figura (3.4), portanto

$$Qft = T_{ta} \cdot F_{max,i}, \quad (3.24)$$

onde:

Qft : quantidade de veículos que flui em um intervalo totalmente aproveitado;

T_{ta} : tempo total de aproveitamento [número de passos];

$F_{max,i}$: fluxo máximo saindo do trecho i [passo.veíc./s].

Substituindo a equação (3.21) na equação (3.24), tem-se:

$$Qft = T_{ta} \cdot n_i \cdot passo \cdot \frac{v_m}{3,6 \cdot c_i}, \quad (3.25)$$

onde:

Qft : quantidade de veículos que flui em um intervalo totalmente aproveitado;

n_i : número de pistas no trecho i ;

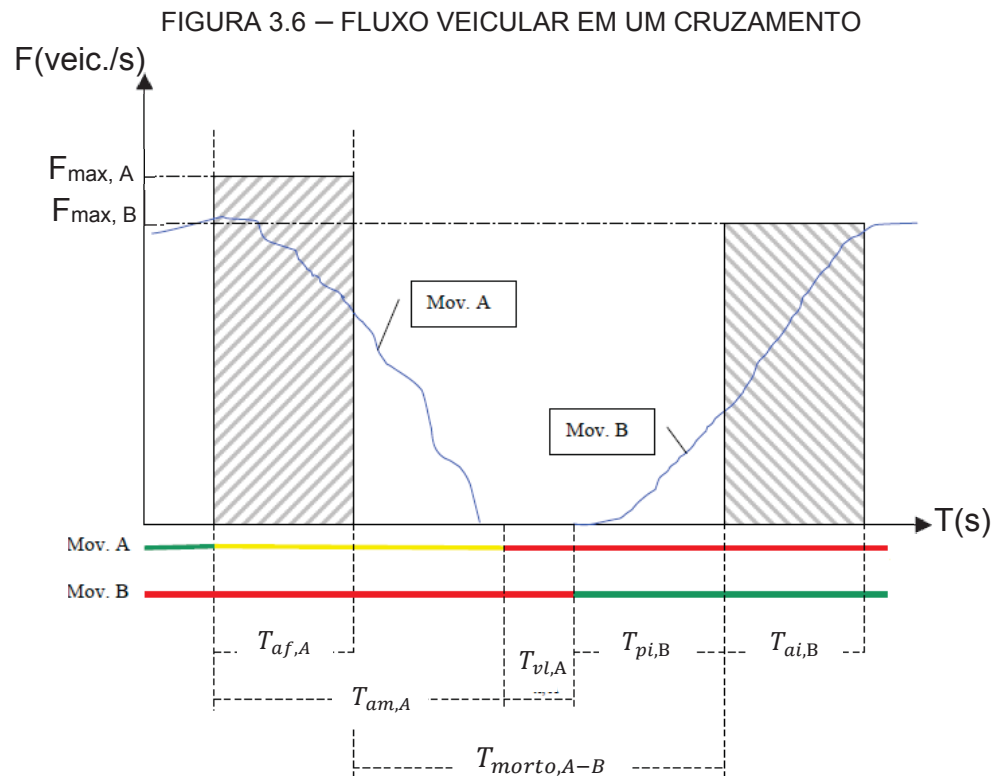
T_{ta} : tempo total de aproveitamento[s/passos];

v_m : velocidade média do fluxo de veículos nos trechos [km/h];

c_i : comprimento do cruzamento saindo do trecho i [m].

3.2.3 Tempo morto

Fazendo-se uso da teoria apresentada para a uniformização do fluxo em uma via livre, pode-se estender o conceito para um cruzamento e assim determinar o tempo morto, ou seja, o tempo perdido no ciclo, pois nesse intervalo de tempo o fluxo é considerado zero. Vide figura (3.6).



FONTE: Adaptado de VILANOVA (2018).

Segundo VILANOVA (2018), o fluxo real na via principal e na via transversal são representados respectivamente pelos movimentos *A* e *B*, onde estes são substituídos pelos retângulos equivalentes de alturas $F_{max,A}$ e $F_{max,B}$. Sendo assim, o fluxo de veículos do movimento *A* é aproximado ao fluxo máximo $F_{max,A}$ até o final do intervalo $T_{af,A}$ e o fluxo de veículos do movimento *B* é aproximado ao fluxo máximo $F_{max,B}$ a partir do início do intervalo $T_{ai,B}$. Com isso, todo fluxo entre o fim do intervalo $T_{af,A}$ e o início do $T_{ai,B}$ passa a ser considerado zero.

Portanto, na transição dos estágios de fechamento de um semáforo e abertura de outro do mesmo cruzamento pode haver um intervalo de vermelho geral, chamado de vermelho de limpeza, onde este período também é considerado para obtenção do tempo morto $T_{morto,A-B}$, como segue na equação (3.26).

$$T_{morto,A-B} = T_{am,A} + T_{vl,A} + T_{pi,B} - T_{af,A} \quad (3.26)$$

onde:

$T_{morto,A-B}$: tempo morto entre a transição do movimento *A* para o *B*;

$T_{am,A}$: tempo de amarelo do movimento A ;

$T_{vl,A}$: tempo de vermelho de limpeza do movimento A ;

$T_{pi,B}$: tempo perdido no início do movimento B ;

$T_{af,A}$: tempo aproveitado no final do movimento A .

A seguir apresenta-se um estudo para obtenção dos valores dos parâmetros $T_{am,A}$, $T_{vl,A}$, $T_{pi,B}$ e $T_{af,A}$ para possibilitar a determinação do tempo morto $T_{morto,A-B}$.

3.2.4 Tempo de amarelo

O tempo de amarelo é considerado o intervalo de tempo entre o final da luz verde do semáforo e o início da luz vermelha do mesmo. Tal intervalo deve possuir um tempo mínimo para permitir que os motoristas desta via que se aproximam do semáforo em amarelo possam realizar a frenagem do veículo com segurança, mesmo que este esteja trafegando com a velocidade máxima permitida da via, e o tempo de amarelo também deve possibilitar ao motorista que esteja a uma distância próxima ao semáforo realizar a travessia do cruzamento antes da abertura do semáforo da via conflitante. Portanto essa definição considera que o tempo de entreverdes é formado apenas pelo tempo de amarelo.

Segundo GAZIS, HERMAN e MARADUDINT (1959), o tempo de amarelo pode ser obtido pela seguinte equação:

$$T_{am} = T_{pr} + \frac{v_m}{2d} + \frac{C_i + C_{mv}}{v_m}, \quad (3.27)$$

onde:

T_{am} : tempo de amarelo [s];

T_{pr} : tempo de percepção e reação [s];

v_m : velocidade média do fluxo de veículos nos trechos [m/s];

d : desaceleração do veículo durante a frenagem [m/s²];

C_i : comprimento do cruzamento saindo do trecho i [m];

C_{mv} : comprimento médio dos veículos [m].

Atualmente o tempo de amarelo vem sendo calculado apenas para permitir ao motorista parar com segurança na faixa de retenção após a percepção da luz amarela, o que possibilita excluir a última parcela da equação (3.27). Sendo assim, o mínimo tempo de amarelo passa a ser calculado pela seguinte equação:

$$T_{am} = T_{pr} + \frac{v_m}{2d} \quad (3.28)$$

Portanto, o amarelo passa a estar associado apenas a ideia de retenção. A parcela $(C_i + C_{mv})/v_m$, fica restrita a outro cálculo.

3.2.5 Tempo de percepção e reação

O tempo de percepção e reação é definido pelo tempo que o motorista leva para perceber o aparecimento da luz amarela do semáforo e reagir acionando os freios. Segundo pesquisas realizadas por Sousa (2011), que analisou uma amostra de 146 veículos utilitários, 36 motos e 32 ônibus em quatro vias brasileiras, mostrou que o tempo de percepção e reação possui uma média de 1,1, 1,5 e 1,3 segundos respectivamente. E 85% da amostra apresentam um tempo de percepção e reação de no máximo 1,3 segundos.

3.2.6 Taxa de desaceleração

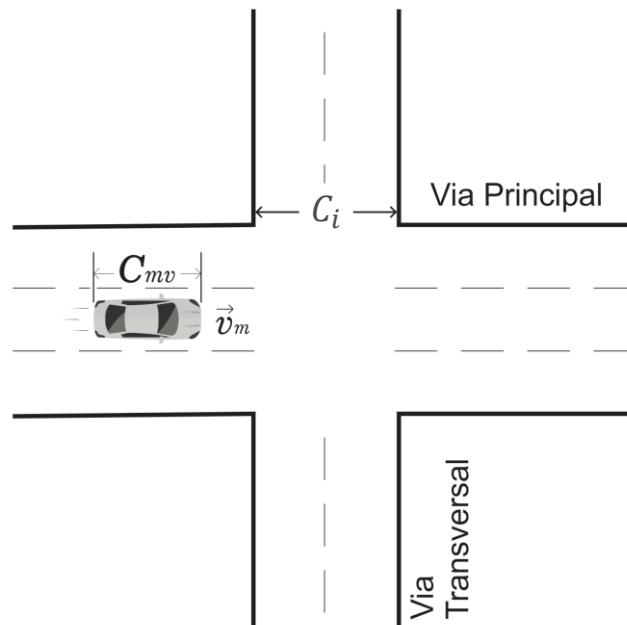
Segundo Sousa (2011), a taxa de desaceleração d dos veículos com o surgimento do sinal amarelo é de $3,3 \text{ m/s}^2$ em 85% da amostra analisada em quatro vias brasileiras, onde três destas possuem velocidade máxima permitida de 60 km/h e a outra de 70 km/h. Tal taxa mostra-se consistente com as encontradas na literatura internacional, onde os valores de maior ocorrência ficam entre $2,4$ e $3,6 \text{ m/s}^2$.

3.2.7 Tempo de vermelho geral

O tempo de vermelho geral também conhecido como vermelho de limpeza ou vermelho total, possibilita ao motorista que acaba de entrar na zona de cruzamento no final do tempo amarelo realizar a travessia antes da abertura do semáforo da via

conflitante. Para este cálculo leva-se em consideração um veículo de comprimento C_{mv} , o comprimento C_i do cruzamento saindo do trecho i e a velocidade média no início da travessia v_m , como ilustra a figura (3.7).

FIGURA 3.7 – CRUZAMENTO COM COMPRIMENTO C_i E VEICULO DE COMPRIMENTO C_{mv}



FONTE: O autor (2018).

Portanto, o tempo de vermelho de limpeza T_{vl} é determinado pela última parcela da equação (3.27), ou seja,

$$T_{vl} = \frac{C_i + C_{mv}}{v_m}. \quad (3.29)$$

3.2.8 Tempo aproveitado após o final do sinal verde

O tempo aproveitado após o final do sinal verde T_{af} é o intervalo de tempo do início do sinal amarelo, ou seja, é quando os veículos realizam a travessia do cruzamento por não possuírem uma distância segura para pararem antes da faixa de retenção. Esse tempo foi convencionado pelo autor como sendo a média do tempo de

percepção e reação T_{pr} e o tempo de amarelo T_{am} , conforme a seguinte equação (3.30):

$$T_{af} = \frac{T_{am} + T_{pr}}{2}. \quad (3.30)$$

3.2.9 Tempo perdido no início do verde

O tempo perdido no início do verde considerado neste trabalho é o tempo que o motorista leva para perceber o início do sinal verde e realizar a aceleração do veículo até que este atinja uma velocidade equivalente a 50% da velocidade média do fluxo de veículos v_m , considerando que a via se encontra desobstruída, foi definido pelo autor e segue com uma estrutura semelhante a equação (3.28). Sendo assim, o tempo perdido no início do verde é dado pela seguinte equação:

$$T_{pi} = T_{pri} + \frac{v_m}{2a_i}, \quad (3.31)$$

onde:

T_{pi} : tempo perdido no início do verde [s];

T_{pri} : tempo de percepção e reação inicial [s];

v_m : velocidade média do fluxo de veículos nos trechos [m/s];

a_i : aceleração média inicial [m/s²].

O tempo de percepção e reação inicial T_{pri} pode ser aproximado ao tempo de percepção e reação T_{pr} .

3.2.10 Cálculo do tempo morto

Considerando os seguintes dados de um cruzamento, onde:

$T_{pr} = 1,3$ s; $T_{pri} = 1,3$ s; $v_m = 19$ m/s; $d = 3,3$ m/s²; $a_i = 5$ m/s²; $C_i = 5$ m; $C_{mv} = 22,4$ m.

Fazendo-se uso das equações (3.28), (3.29), (3.30) e (3.31) chega-se nos seguintes resultados:

$$T_{am} \approx 4,17\text{s}; T_{vl} \approx 1,44\text{ s}; T_{af} \approx 2,74\text{ s}; T_{pi} \approx 3,2\text{ s}.$$

Logo, substituindo os valores acima na equação (3.26), encontram-se o tempo morto do ciclo semafórico no cruzamento em questão. Sendo este,

$$T_{morto,A-B} \approx 6,08\text{ s}.$$

3.3 VALORES CONSIDERADOS

O modelo apresentado foi programado utilizando algumas adaptações no algoritmo de KAVISKI (2017) e inserido a heurística *Variable Neighborhood Search* (VNS) para otimizar a busca do mínimo da função, sendo a linguagem de programação utilizada a “FreePascal”.

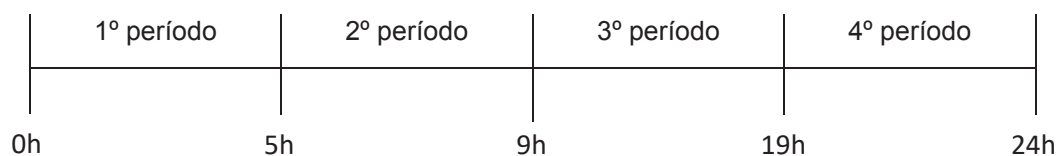
O programa foi testado levando-se em consideração uma via principal com 3 de suas vias transversais, mas os veículos retidos na 3ª transversal não são levados em conta na função objetivo do modelo apresentado por se tratar do fim da malha. E os dados considerados nas simulações foram:

$$\alpha_i = 80\%, c_i = 5\text{m}, n_i = 3\text{ pistas, onde } i = 1,3\text{ e }5.$$

$$\beta_i = 50\%, c_i = 7,5\text{m}, n_i = 2\text{ pistas, onde } i = 2\text{ e }4.$$

Em cada ciclo foi levado em conta nas simulações um tempo morto $T_{morto,A-B} = 6\text{ s}$. E os 4 períodos utilizados na programação semafórica são exibidos na figura (3.8) abaixo:

FIGURA 3.8 – PERÍODOS CONSIDERADOS DO DIA



FONTE: O autor (2019).

No início da simulação as ruas são consideradas vazias e todos os semáforos fechados. Portanto, o carregamento da malha viária é dado por, $qe_{i,0} = 0$ para $i = 3$ e 5 . Para $qe_{i,0}$ tem-se valores diferentes de zero para os trechos 1,2 e 4, a partir dos quais

foram calculados de acordo com o tamanho do passo escolhido sobre os dados fornecidos pela SETRAN. Já os parâmetros, $passo$ e v_m foram modificados entre as simulações.

3.4 HEURÍSTICA *VARIABLE NEIGHBORHOOD SEARCH* - VNS

Para minimizar a função objetivo do modelo elaborado será utilizada uma heurística de busca em variáveis vizinhas, conhecida como VNS, proposta por BOCTOR (1993). Define-se uma vizinhança de um problema discreto de otimização, a vizinhança $N(s)$ de uma solução é representado pelo conjunto $\{s' \mid d(s',s) \leq \epsilon\}$, onde d representa uma distância dada que está relacionada ao operador de movimento.

A VNS surgiu da heurística *Variable Neighbourhood Descent* - VND que consiste basicamente em gerar um conjunto de vizinhança $N(s)$ a partir de uma solução inicial ao analisar a melhor solução, onde a partir desta gera-se um novo conjunto vizinhança e o processo se repete até que algum critério de parada estabelecido seja satisfeito conforme o algoritmo contido na figura (3.9):

FIGURA 3.9 – ALGORITMO DA HEURÍSTICA VND

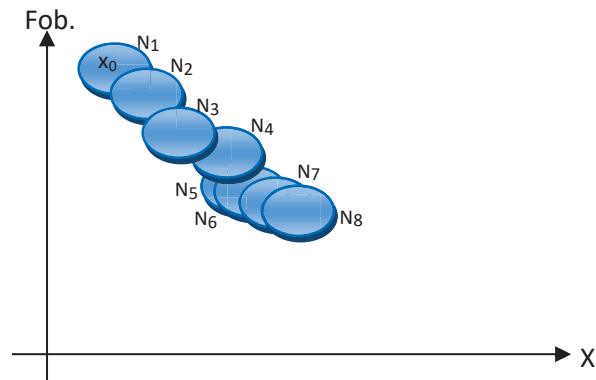
Algoritmo: *Variable Neighbourhood Descent*- VND

Input: um conjunto de estruturas de vizinhança N_l para $l = 1, \dots, l_{max}$
 $x = x_0$; {Gerar uma solução inicial}
 $l = 1$;
While $l \leq l_{max}$ **do**
 encontra o melhor vizinho x' de $N_l(x)$;
 If $f(x') < f(x)$ **then**
 $x = x'$;
 $l = 1$;
 else
 $l = l + 1$;
Print: melhor solução encontrada.

FONTE: Adaptado de BOCTOR (1993).

O funcionamento desta heurística pode ser melhor compreendido com a seguinte figura (3.10):

FIGURA 3.10 – EXEMPLO DA GERAÇÃO DE CONJUNTOS DE VIZINHANÇA PARA CHEGAR NO MÍNIMO DA FUNÇÃO



FONTE: O autor (2018).

Como a heurística VND é uma estratégia para melhorar a busca de uma solução local, esta tende a ficar presa em mínimos locais. Para contornar este problema o autor BOCTOR (1993), da VNS, melhorou a heurística VND dando a possibilidade de mudar o local de busca fazendo-se uso de diferentes vizinhanças para fugir de alguns ótimos locais, gerando assim mais diversificação no espaço de busca, conforme se pode notar no código a seguir da figura (3.11):

FIGURA 3.11 – ALGORITMO DA HEURÍSTICA VNS

Algoritmo: Variable Neighborhood Search - VNS

Input: um conjunto de estruturas de vizinhança N_k para $k = 1, \dots, k_{\max}$ para diversificar
um conjunto de estruturas de vizinhança N_l para $l = 1, \dots, l_{\max}$ para busca local

$x = x_0$; {Gerar uma solução inicial}

Repeat

$k = 1$;

While $k \leq k_{\max}$ **do**

Diversificar: escolhe uma solução aleatória x' da k -ésima vizinhança $N_k(x)$ de x ;

$l = 1$;

While $l \leq l_{\max}$ **do**

Busca local: encontra o melhor vizinho x'' de $N_l(x')$;

If $f(x'') < f(x')$ **then**

$x' = x''$;

$l = 1$;

else

$l = l + 1$;

if $f(x'') < f(x)$ **then**

$x = x''$;

{continuar a busca local com o conjunto de vizinhança N_k }

else

$k = k + 1$;

Until critério de parada

Print: melhor solução encontrada.

FONTE: Adaptado de BOCTOR (1993).

Para minimizar a função objetivo do modelo considerado foi utilizada a heurística VNS, cuja solução inicial “ x ” é uma matriz com tempos de verde e vermelho dos semáforos de cada cruzamento para cada período do dia, sendo “ x ” o mesmo que a variável “ t ” do modelo matemático. O algoritmo desenvolvido para minimizar a função objetivo do presente trabalho apresenta 8 diferentes estruturas para criação de conjuntos de vizinhança, onde:

$N_1(x)$: cada vizinho é gerado alterando um único elemento da matriz solução x com uma unidade para mais;

$N_2(x)$: cada vizinho é gerado alterando um único elemento da matriz solução x com uma unidade para menos;

$N_3(x)$: cada vizinho é gerado alterando uma linha da matriz solução x com uma unidade para mais;

$N_4(x)$: cada vizinho é gerado alterando uma linha da matriz solução x com uma unidade para menos;

$N_5(x)$: cada vizinho é gerado alterando uma coluna da matriz solução x com uma unidade para mais;

$N_6(x)$: cada vizinho é gerado alterando uma coluna da matriz solução x com uma unidade para menos;

$N_7(x)$: cada vizinho é gerado alterando a solução x com r unidades a mais para cada elemento da matriz, tal que, r varia aleatoriamente de zero à dois;

$N_8(x)$: cada vizinho é gerado alterando a solução x com r unidades a menos para cada elemento da matriz, tal que, r varia aleatoriamente de zero à dois;

Portanto, os conjuntos de vizinhança $N_1(x)$ e $N_2(x)$ geram vizinhos muito próximos ao vizinho de referência x , mas as vizinhanças $N_3(x)$, $N_4(x)$, $N_5(x)$ e $N_6(x)$ exploram vizinhos um pouco mais distantes de x e os conjuntos de vizinhança $N_7(x)$ e $N_8(x)$ são muito mais abrangentes que os demais, onde estes dois últimos são interessantes para fugir de mínimos locais, gerando uma diversificação no espaço de busca.

Na figura (3.12) a seguir, pode-se analisar a estrutura do algoritmo com base na heurística VNS.

FIGURA 3.12 – ALGORITMO VNS ADAPTADO

Algoritmo: VNS adaptado.

Input: um conjunto de estruturas de vizinhança N_k para $k = 1, \dots, 8$.

For $i = 1$ to 1000 **do**

 Gerar uma solução aleatoria x_i

$x =$ melhor solução x_i obtida. {solução inicial}

 cont = 1;

While cont \leq 20 **do**

For $k = 1$ to 6 **do**

 Gera vizinhança $N_k(x)$;

Busca local: encontra o melhor vizinho x' de $N_k(x)$;

 Cont = cont + 1;

If $f(x') < f(x)$ **then**

$x = x'$;

 cont = 0;

else

 Gera vizinhança $N_7(x)$ e $N_8(x)$;

Busca local: encontra o melhor vizinho x'' de $N_7(x)$ e $N_8(x)$;

if $f(x'') < f(x)$ **then**

$x = x''$;

end while;

Print: melhor solução encontrada.

FONTE: Adaptado de BOCTOR (1993).

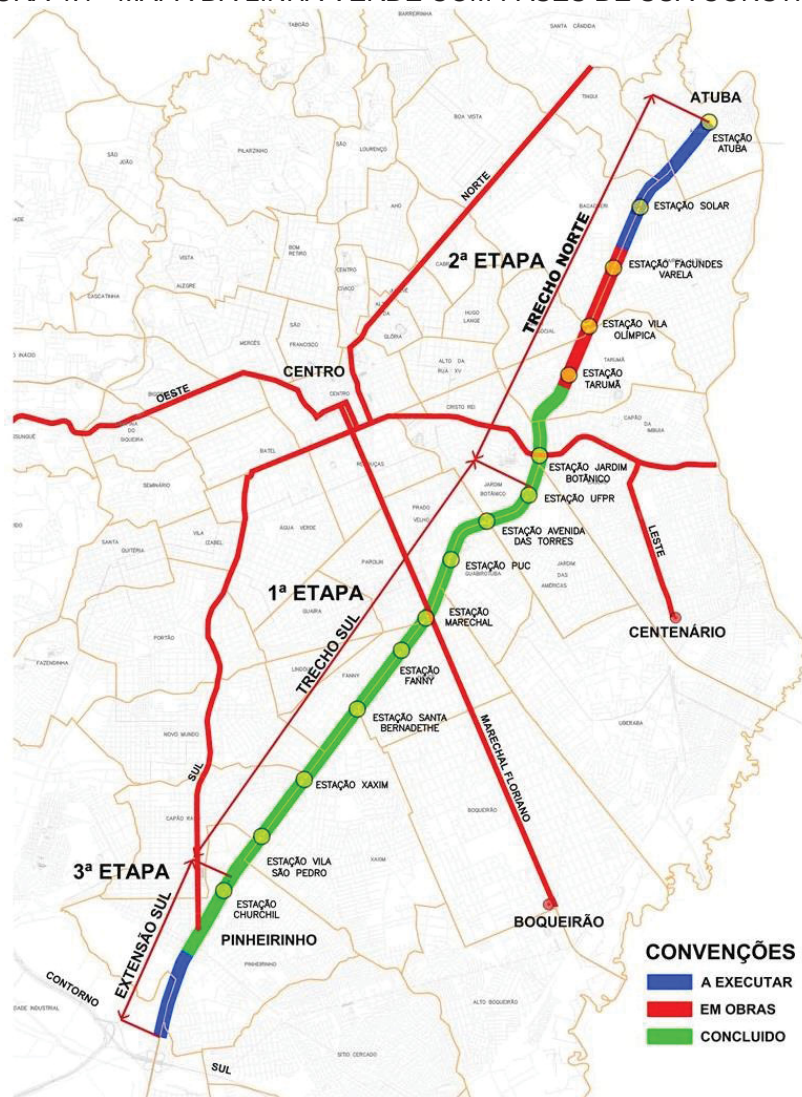
4 ESTUDO DO CASO E RESULTADOS

A seguir apresenta-se uma breve descrição do local onde foi realizado os estudos e os resultados obtidos para diversas alterações de parâmetros do modelo apresentado.

4.1 LOCAL DE ESTUDO

As simulações computacionais foram baseadas no modelo matemático descrito no capítulo (3), para um trecho de uma via denominada Linha Verde na cidade de Curitiba - PR, conforme ilustra as figuras (4.1), (4.2) e (4.3).

FIGURA 4.1 - MAPA DA LINHA VERDE COM FASES DE SUA CONSTRUÇÃO



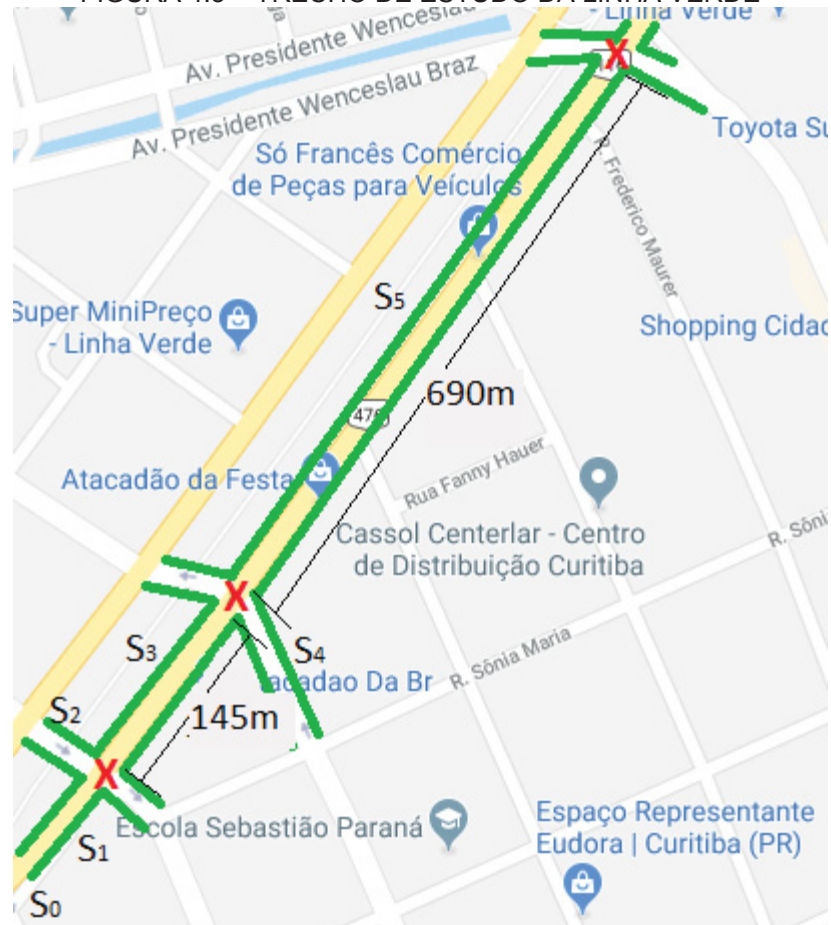
FONTE: CURITIBA (2017).

FIGURA 4.2 - TRECHO DA LINHA VERDE



FONTE: GAZETA DO POVO (2015).

FIGURA 4.3 – TRECHO DE ESTUDO DA LINHA VERDE



FONTE: GOOGLE MAPS (2019).

Na figura (4.3), os xises em vermelho indicam os cruzamentos semaforizados, os quais foram considerados neste trabalho. Entre os cruzamentos indicados situam-se os trechos 3 e 5, onde o trecho 3 possui 145 metros e o trecho 5 com comprimento de 690 metros. A velocidade máxima permitida nos trechos da via principal é de 70km/h e das vias transversais correspondente é de 60 km/h.

A Linha Verde é um projeto urbanístico inaugurado no dia 09 de maio de 2009, para substituir a antiga rodovia que cortava a capital paranaense em dois trechos, sendo eles as BR-116 e BR-376. Interliga o bairro do Pinheirinho na região sul da cidade ao bairro do Atuba na região norte do município, totalizando 18 km de via.

4.2 RESULTADOS

A seguir apresentam-se algumas tabelas e gráficos com os resultados da função objetivo do modelo para 7 diferentes tamanhos de passos, sendo estes de 1s, 2s, 3s, 4s, 5s, 6s e 10s e 5 tipos de velocidades médias do fluxo, onde estas são de 15km/h, 25km/h, 35km/h, 45km/h e 60km/h. Para cada combinação de velocidade e passo o modelo foi executado 10 vezes para analisar qual será a estabilidade dos resultados com a utilização da heurística VNS, totalizando 350 execuções do programa.

Nas tabelas (4.1) à (4.10) apresentam-se colunas de tempo(s), onde estes tempos são referentes ao tempo de execução do programa e colunas referentes ao valor da função objetivo (F.O).

TABELA 4.1 – RESULTADOS COM $v_m = 15$ Km/h e *passo* = 1, 2, 3 e 4s

Resultados	Passo 1s		Passo 2s		Passo 3s		Passo 4s	
	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)
1	45,655	712,875	52,641	133,595	63,792	49,020	72,169	48,637
2	45,362	477,443	58,121	121,850	62,677	65,595	67,292	82,475
3	44,285	720,995	53,233	138,627	64,169	50,985	78,887	53,696
4	46,181	529,958	52,948	183,406	63,089	48,563	70,124	53,823
5	44,602	645,006	54,601	120,910	62,309	65,825	72,335	66,479
6	47,471	556,861	52,319	155,367	62,734	49,483	69,791	53,845
7	44,632	648,646	54,690	254,148	64,837	49,176	76,936	54,410
8	45,268	507,969	54,411	146,972	59,303	85,709	64,821	99,171
9	45,563	512,103	54,256	119,722	61,088	74,992	86,710	52,613
10	46,930	478,656	54,715	146,587	64,667	49,252	69,001	31,763
Menor	44,285	477,443	52,319	119,722	59,303	48,563	64,821	31,763
Maior	47,471	720,995	58,121	254,148	64,837	85,709	86,710	99,171
Média	45,595	579,051	54,193	152,118	62,867	58,860	72,807	59,691
Desv. Padrão	0,972	89,494	1,567	38,628	1,611	12,704	6,107	17,912

FONTE: O autor (2018).

TABELA 4.2 – RESULTADOS COM $v_m = 15$ Km/h e *passo* = 5, 6 e 10s

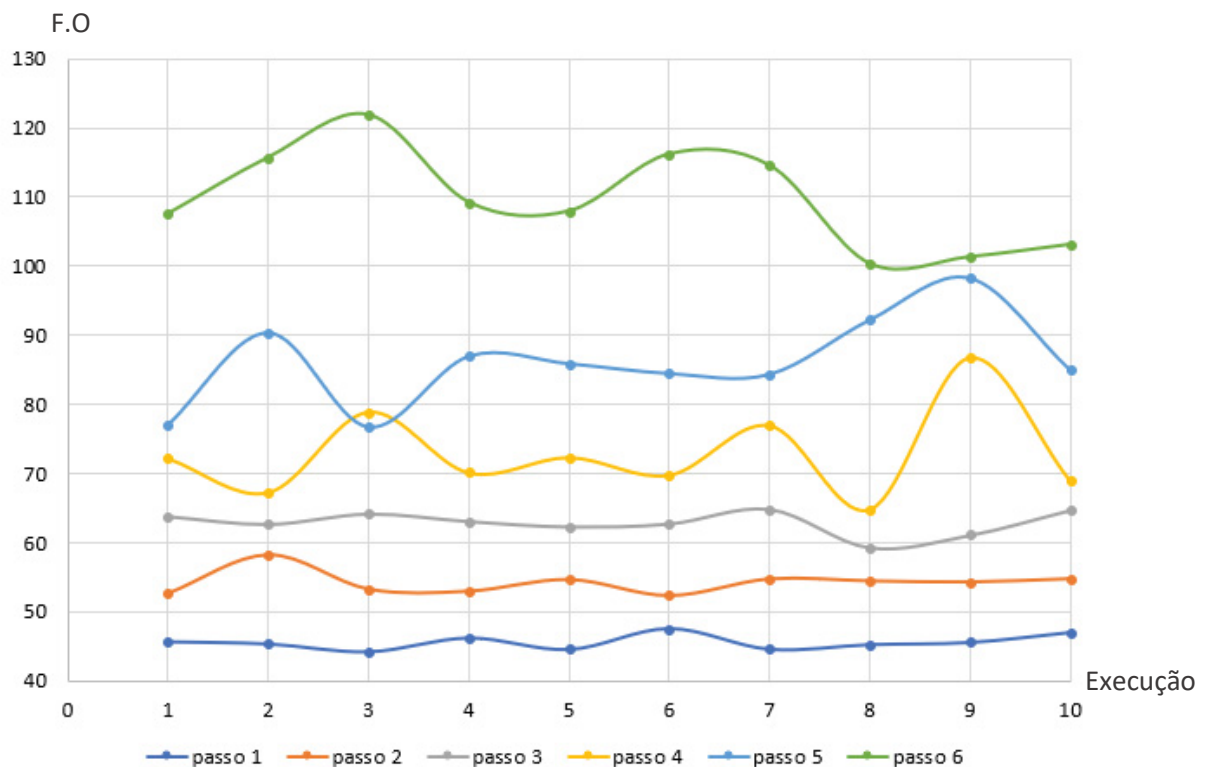
Resultados	Passo 5s		Passo 6s		Passo 10s	
	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)
1	77,043	20,100	107,584	9,693	3187,240	4,719
2	90,393	22,743	115,744	9,683	5268,486	4,679
3	76,662	21,725	121,911	9,462	3491,891	4,656
4	86,976	18,848	109,220	9,803	3653,835	4,709
5	85,886	20,431	107,943	10,325	3148,962	4,733
6	84,464	19,008	116,221	9,544	3146,095	4,652
7	84,334	43,856	114,578	9,590	4380,045	4,625
8	92,238	19,363	100,328	9,228	2369,889	4,690
9	98,335	19,086	101,304	10,252	2549,451	4,618
10	84,957	19,279	103,102	9,661	3657,519	4,736
Menor	76,662	18,848	100,328	9,228	2369,889	4,618
Maior	98,335	43,856	121,911	10,325	5268,486	4,736
Média	86,129	22,444	109,794	9,724	3485,341	4,682
Desv. Padrão	6,203	7,240	6,794	0,319	804,699	0,041

FONTE: O autor (2018).

Com as informações das tabelas (4.1) e (4.2) foram propostos os gráficos apresentados nas figuras (4.4) a (4.7), mas nas figuras (4.4), (4.5) e (4.6) não foram considerados os resultados da função objetivo (F.O) para o passo de 10s, pois os resultados deste possuem valores muito altos e uma grande oscilação entre os

resultados, tornando inconveniente sua representação nestes gráficos devido a escala. Os resultados da (F.O) para o passo de 10s foram muito mais elevados que os demais tamanhos de passos devido a um acúmulo de três fatores, sendo eles, a baixa velocidade média do fluxo, o tamanho do passo muito grande e a estrutura do programa para realizar as simulações que não se comporta muito bem para tamanho de passos grandes, devido a inclusão do “tempo morto” de 6s, onde se perde 3s no despacho dos veículos no início do verde da via principal e mais 3s perdido no início do verde da via secundária, mas como esses tempos de 3s é contabilizado em números de passos nas simulações, então para o passo de 10s o “tempo morto” passa a ser de 20 segundos. Como isso, os resultados apresentados oscilaram muito e consequentemente o desvio padrão tornou-se muito superior que os demais.

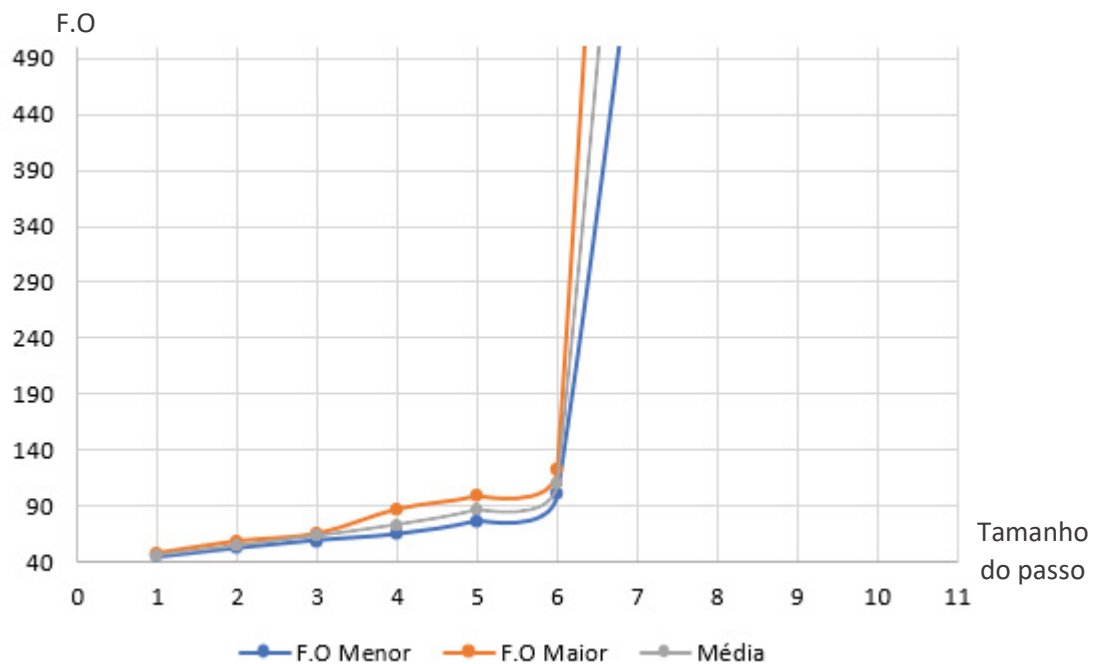
FIGURA 4.4 – GRÁFICO DOS RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 15$ Km/h



FONTE: O autor (2018).

Analisando a figura (4.4), percebe-se uma redução das oscilações dos resultados na medida em que o tamanho do passo tem se tornado menor e para passos menores ou igual a 3s as oscilações apresentam uma redução de forma mais significativa.

FIGURA 4.5 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS RESULTADOS DA F.O EM RELAÇÃO AO PASSO COM $v_m = 15$ Km/h

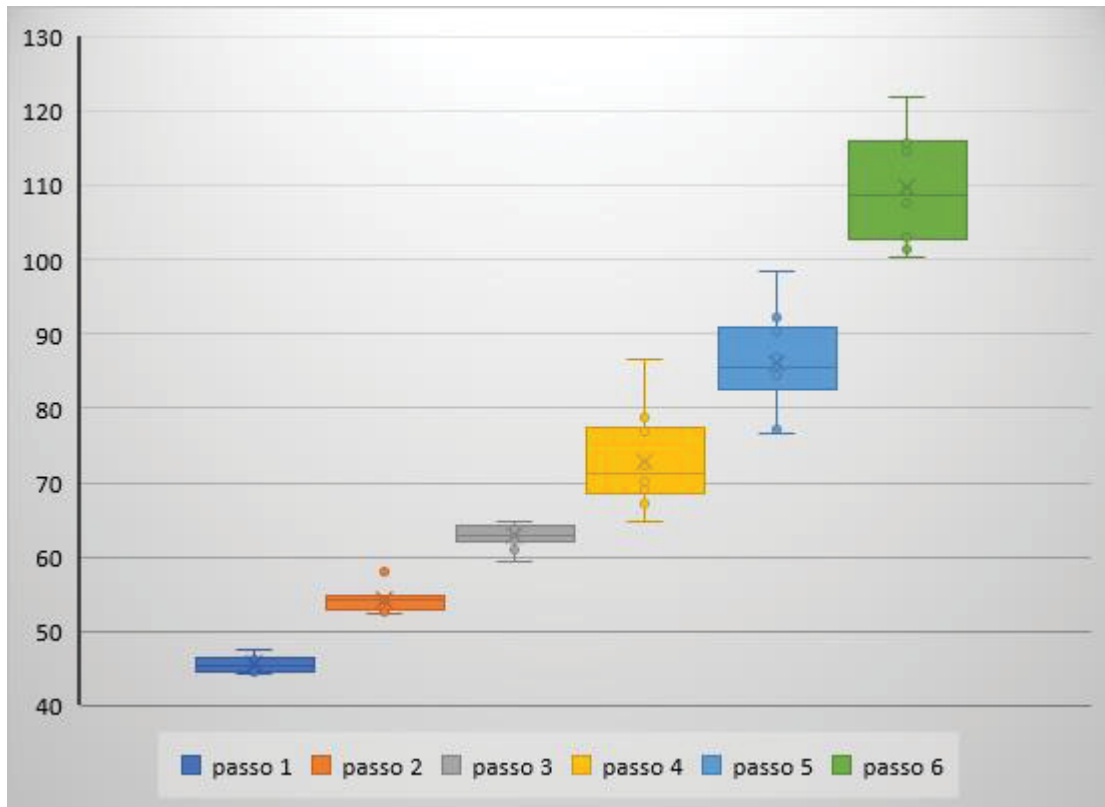


FONTE: O autor (2018).

Na figura (4.5) observa-se o comportamento dos menores, maiores e a média dos resultados da F.O, em que quanto maior o tamanho do passo maiores são os resultados, onde ocorreu um grande crescimento da F.O quando comparado o passo de 6s com o passo de 10s.

Os gráficos apresentados nas figuras (4.6), (4.10), (4.14), (4.18) e (4.22) revelam vários aspectos dos resultados da função objetivo, dentre eles: tendência central, variabilidade, simetria e possibilita visualizar valores atípicos. Tais aspectos podem ser avaliados através das informações presentes no gráfico, sendo elas: o mínimo, o primeiro quartil, a mediana, a média, o terceiro quartil e o máximo.

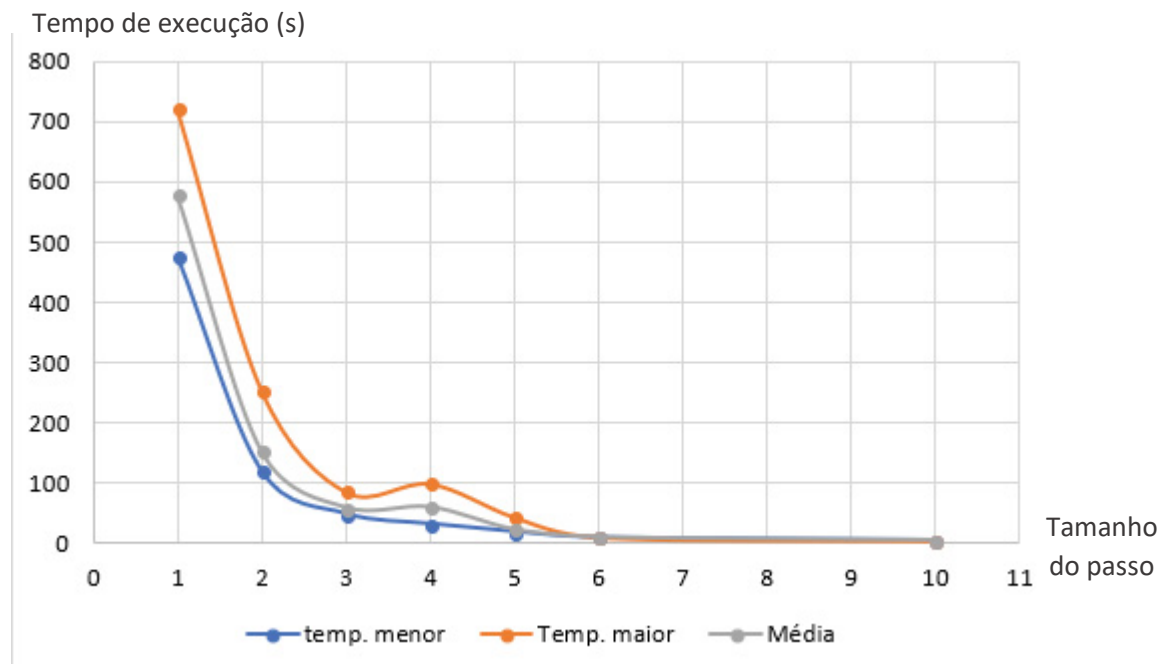
FIGURA 4.6 – GRÁFICO DAS VARIAÇÕES DE RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 15$ Km/h



FONTE: O autor (2018).

Na figura (4.6) é observado que a mediana, representada por um segmento de reta na horizontal dentro do retângulo, e a média, sendo está representada por um 'x', onde se nota uma aproximação dos valores entre as mesmas, indicando, pois, uma simetria dos resultados obtidos. O passo 2 apresentou um resultado atípico, muito maior que os valores gerados nesse passo. Tal valor atípico é representado no gráfico por um pequeno círculo.

FIGURA 4.7 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS TEMPOS DE EXECUÇÃO EM RELAÇÃO AO PASSO COM $v_m = 15$ Km/h



FONTE: O autor (2018).

Na figura (4.7) observa-se o comportamento dos menores, maiores e a média dos tempos de execução do programa, em que quanto maior o tamanho do passo, muito menor se torna o tempo de execução.

A seguir são apresentadas nas tabelas (4.3) e (4.4) os resultados obtidos quando a velocidade média é de 25 km/h para o tamanho do passo igual a 1, 2, 3 e 4s na tabela (4.3) e passo igual a 5, 6 e 10s na tabela (4.4).

TABELA 4.3 – RESULTADOS COM $v_m = 25$ Km/h e $passo = 1, 2, 3$ e $4s$

Resultados	Passo 1s		Passo 2s		Passo 3s		Passo 4s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	44,431	480,862	53,543	119,834	59,771	56,808	64,113	23,964
2	44,774	605,044	51,672	156,074	54,304	64,872	66,582	26,823
3	44,891	515,697	51,591	180,034	57,731	56,001	68,961	36,288
4	45,922	508,860	50,843	125,442	57,372	52,475	64,765	27,112
5	42,901	484,987	50,123	204,123	57,772	63,156	63,428	33,730
6	46,528	497,339	52,304	184,781	56,922	51,352	67,057	24,146
7	44,884	496,471	55,341	118,112	59,557	52,207	65,079	24,561
8	42,889	951,072	51,410	197,051	59,738	52,778	68,031	24,200
9	42,979	484,317	52,048	168,059	57,793	58,980	65,125	30,168
10	42,901	511,519	50,596	500,888	54,345	51,441	63,376	24,688
Menor	42,889	480,862	50,123	118,112	54,304	51,352	63,376	23,964
Maior	46,528	951,072	55,341	500,888	59,771	64,872	68,961	36,288
Média	44,310	553,617	51,947	195,440	57,531	56,007	65,652	27,568
Desv. Padrão	1,272	136,773	1,451	106,124	1,874	4,680	1,826	4,182

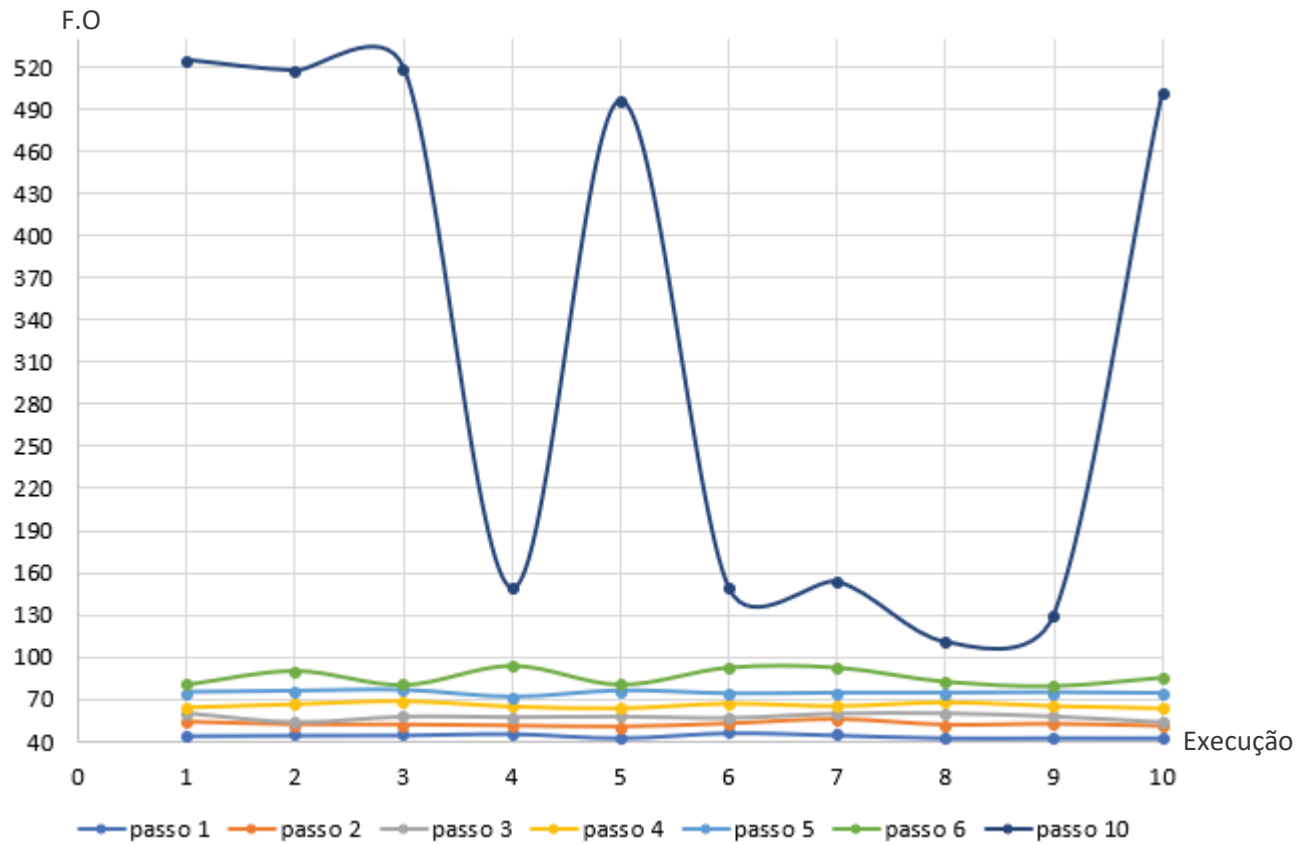
FONTE: O autor (2018).

TABELA 4.4 – RESULTADOS COM $v_m = 25$ Km/h e $passo = 5, 6$ e $10s$

Resultados	Passo 5s		Passo 6s		Passo 10s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	74,652	18,412	80,884	9,712	524,595	4,648
2	75,457	18,539	90,203	9,441	517,022	4,558
3	76,103	19,895	80,610	8,700	518,726	5,155
4	71,189	41,110	93,721	8,645	148,817	4,539
5	75,729	20,439	80,834	9,929	495,660	4,829
6	73,579	18,683	92,504	8,646	149,207	4,558
7	74,105	35,648	92,505	8,881	153,508	4,594
8	74,126	19,913	82,793	9,024	110,754	4,811
9	74,451	19,064	79,889	9,153	129,853	4,576
10	73,912	18,628	85,588	10,363	500,662	4,575
Menor	71,189	18,412	79,889	8,645	110,754	4,539
Maior	76,103	41,110	93,721	10,363	524,595	5,155
Média	74,330	23,033	85,953	9,249	324,880	4,684
Desv. Padrão	1,311	7,797	5,400	0,564	186,965	0,186

FONTE: O autor (2018).

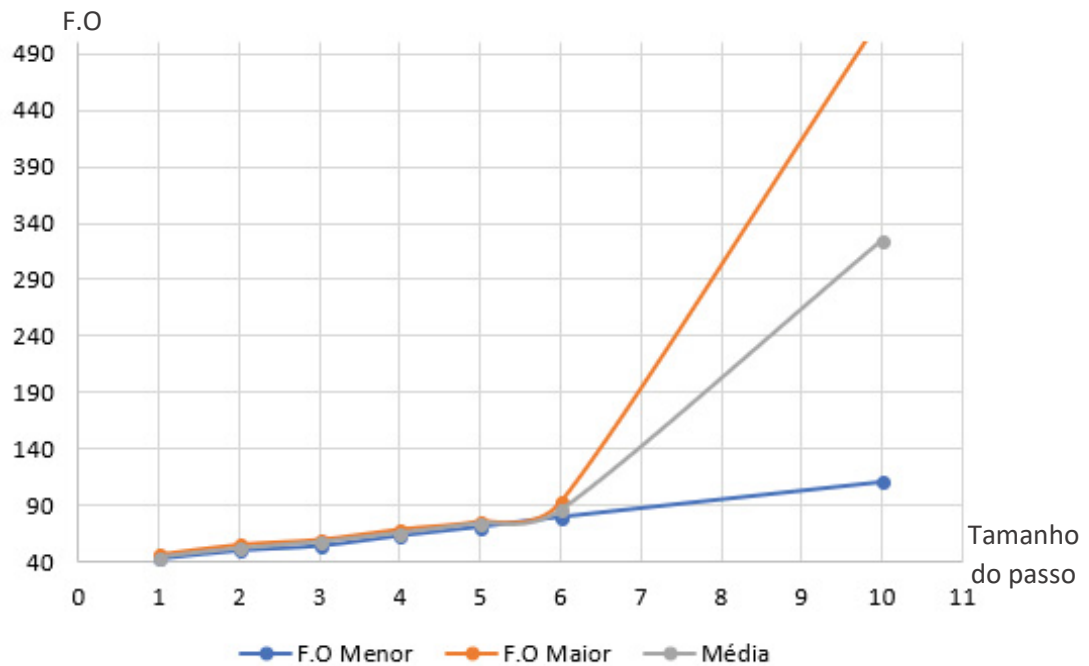
Usando as informações das tabelas (4.3) e (4.4) foi gerado os gráficos apresentados nas figuras (4.8) a (4.11).

FIGURA 4.8 – GRÁFICO DOS RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 25$ Km/h

FONTE: O autor (2018).

Analisando a figura (4.8), percebe-se uma redução das oscilações dos resultados a medida em que o tamanho do passo se torna menor. E as curvas não se sobrepõem umas às outras, mas permanecem próximas quando os passos são menores. Portanto, os resultados tornam-se mais estáveis para passos menores.

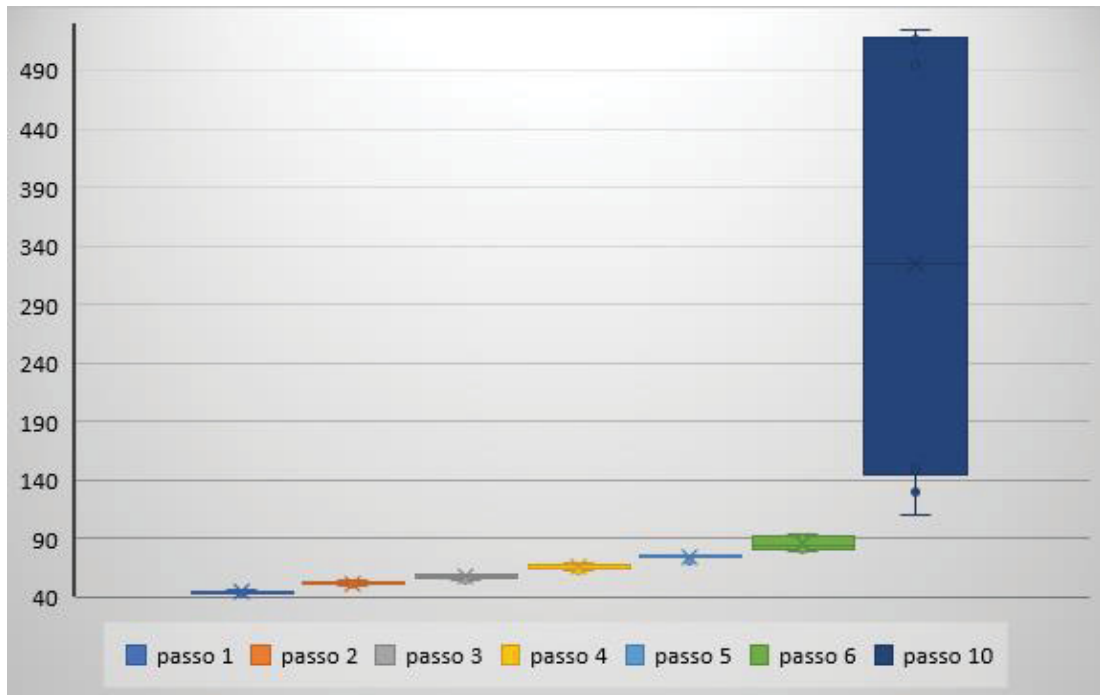
FIGURA 4.9 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS RESULTADOS DA F.O EM RELAÇÃO AO PASSO COM $v_m = 25$ Km/h



FONTE: O autor (2018).

Na figura (4.9) observa-se o comportamento dos menores, maiores e a média dos resultados da F.O, em que quanto maior o tamanho do passo, maiores são os resultados. E, quando se compara este gráfico com o da figura (4.5), nota-se que o crescimento da F.O entre os passos de 6s e 10s é muito menos agressivo. Observa-se quase uma linearidade na função que representa o menor resultado de cada tamanho de passo, representada em azul no gráfico.

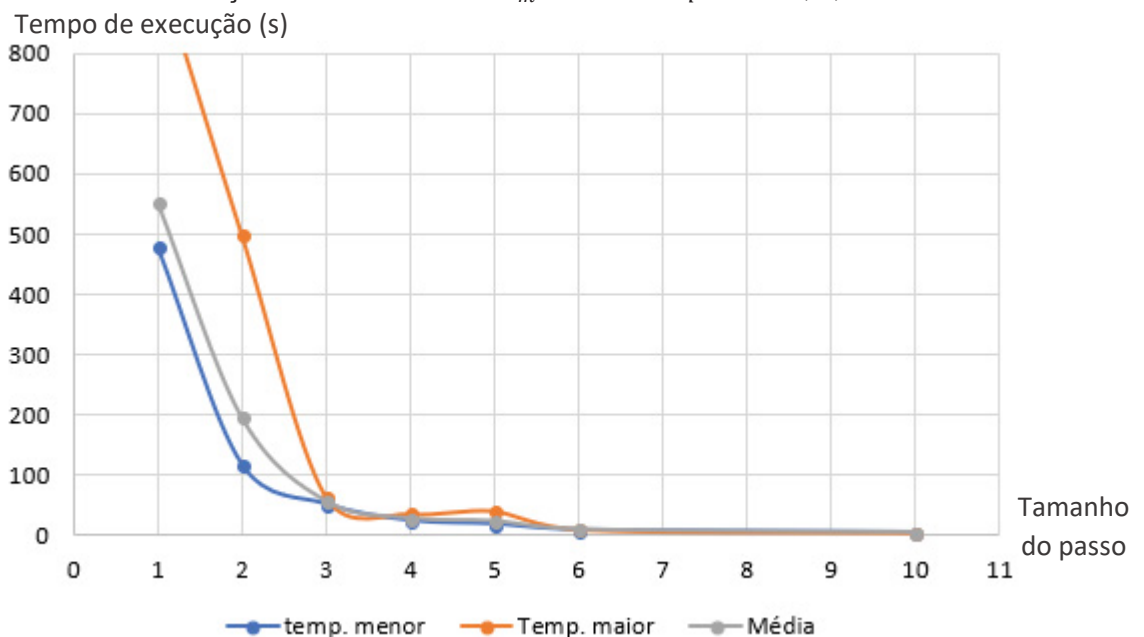
FIGURA 4.10 – GRÁFICO DAS VARIAÇÕES DE RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 25$ Km/h



FONTE: O autor (2018).

No gráfico da figura (4.10) nota-se que no passo 10 houve uma grande amplitude dos resultados, enquanto nos demais passos a amplitude é muito menor. O que torna inviável realizar simulações para uma futura programação nos semáforos com passo de 10s.

FIGURA 4.11 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS TEMPOS DE EXECUÇÃO EM RELAÇÃO AO PASSO COM $v_m = 25$ Km/h e $passo = 1, 2, 3$ e 4s.



FONTE: O autor (2018).

Na figura (4.11) observa-se o comportamento dos menores, maiores e a média dos tempos de execução do programa, em que quanto maior o tamanho do passo, muito menor se torna o tempo de execução.

A seguir são apresentados os resultados obtidos quando a velocidade média é de 35 km/h para o tamanho do passo igual a 1, 2, 3 e 4s na tabela (4.5) e passo igual a 5, 6 e 10s na tabela (4.6).

TABELA 4.5 – RESULTADOS COM $v_m = 35$ Km/h e *passo* = 1, 2, 3 e 4s

Resultados	Passo 1s		Passo 2s		Passo 3s		Passo 4s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	44,206	491,825	48,608	188,442	60,569	59,251	63,665	27,603
2	44,006	1160,220	52,497	126,370	53,449	89,582	63,982	31,968
3	44,384	479,035	50,527	116,314	59,100	52,469	67,032	27,546
4	42,553	569,184	52,712	119,353	61,331	50,606	66,000	27,263
5	45,261	520,491	49,750	138,144	55,617	54,476	64,324	25,829
6	44,848	688,892	49,062	116,513	55,617	57,244	65,414	26,255
7	44,357	528,617	50,462	117,472	56,095	50,497	62,995	26,024
8	45,316	585,078	50,563	127,780	58,878	48,436	64,219	31,034
9	43,119	514,264	50,717	161,530	56,495	53,536	61,421	41,010
10	43,677	480,044	50,413	158,853	60,610	48,525	65,926	32,594
Menor	42,553	479,035	48,608	116,314	53,449	48,436	61,421	25,829
Maior	45,316	1160,220	52,712	188,442	61,331	89,582	67,032	41,010
Média	44,173	601,765	50,531	137,077	57,776	56,462	64,498	29,713
Desv. Padrão	0,839	195,546	1,231	23,378	2,529	11,537	1,560	4,456

FONTE: O autor (2018).

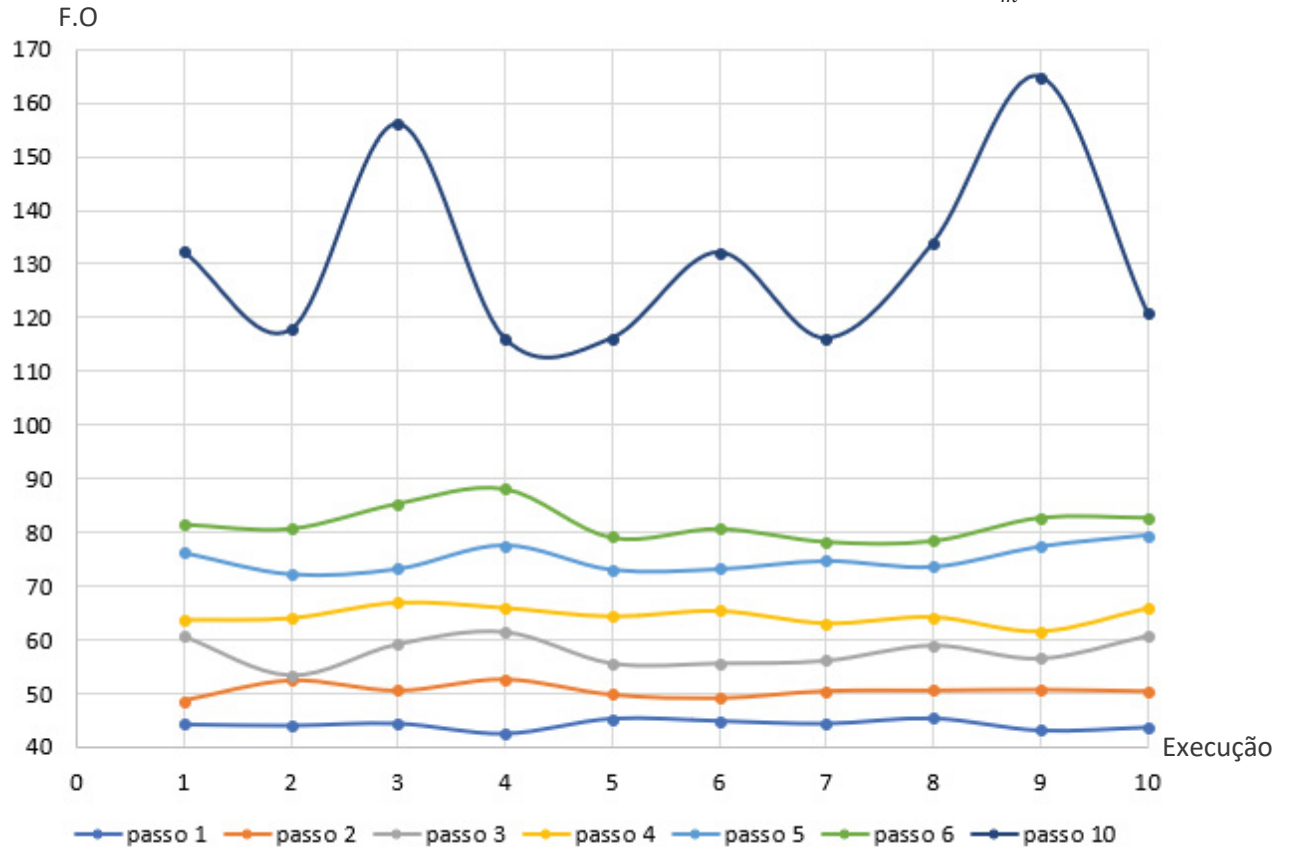
TABELA 4.6 – RESULTADOS COM $v_m = 35$ Km/h e *passo* = 5, 6 e 10

Resultados	Passo 5s		Passo 6s		Passo 10s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	76,198	18,551	81,522	17,093	132,322	4,865
2	72,247	44,240	80,724	9,387	117,761	4,825
3	73,208	22,066	85,302	9,965	156,091	5,013
4	77,556	18,701	88,011	9,694	116,208	6,345
5	73,072	23,457	79,099	10,020	116,208	4,759
6	73,175	19,227	80,724	13,674	132,202	4,699
7	74,678	18,664	78,333	19,714	116,208	7,976
8	73,597	25,014	78,487	8,700	133,969	4,753
9	77,327	22,391	82,751	10,182	164,868	4,780
10	79,393	18,502	82,751	11,377	121,012	4,722
Menor	72,247	18,502	78,333	8,700	116,208	4,699
Maior	79,393	44,240	88,011	19,714	164,868	7,976
Média	75,045	23,081	81,770	11,981	130,685	5,274
Desv. Padrão	2,293	7,404	2,921	3,508	16,489	1,015

FONTE: O autor (2018).

Usando as informações das tabelas (4.5) e (4.6) foram gerados os gráficos apresentados nas figuras (4.12) a (4.15).

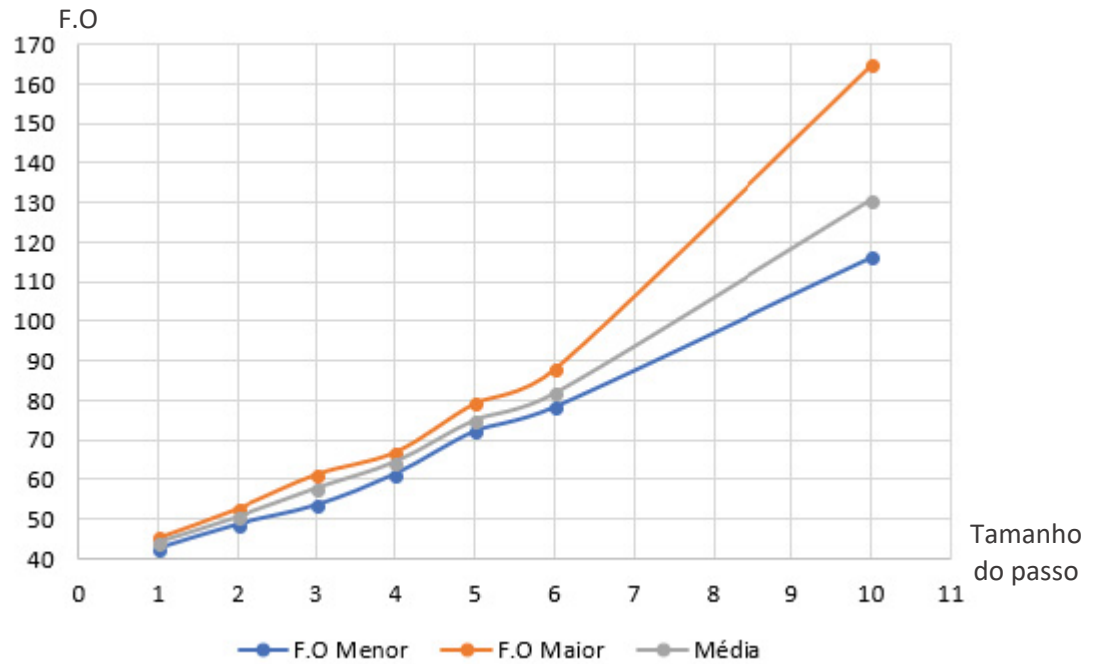
FIGURA 4.12 – GRÁFICO DOS RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 35$ Km/h



FONTE: O autor (2018).

Analisando a figura (4.12), percebe-se uma redução das oscilações dos resultados na medida em que o tamanho do passo se torna menor. E observa-se uma diminuição das oscilações do passo de tamanho 10s quando comparada com o da figura (4.8).

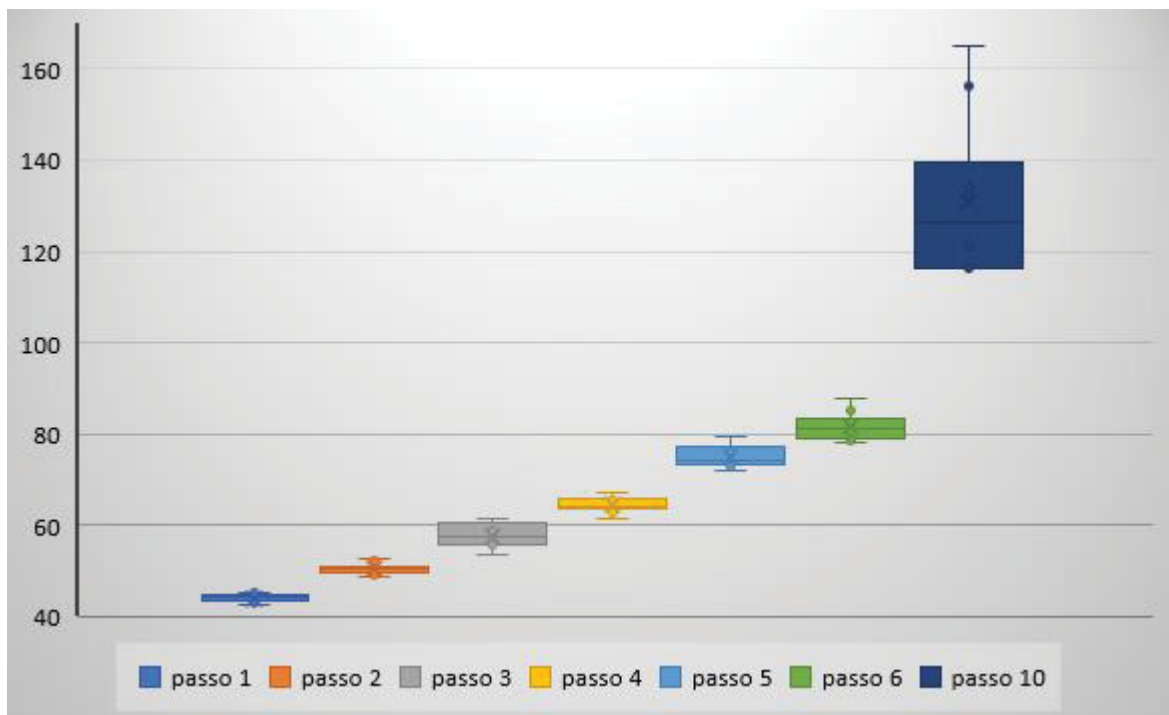
FIGURA 4.13 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS RESULTADOS DA F.O EM RELAÇÃO AO PASSO COM $v_m = 35$ Km/h



FONTE: O autor (2018).

Na figura (4.13) observa-se o comportamento dos menores, maiores e a média dos resultados da F.O, em que quanto maior for o tamanho do passo maiores são os resultados.

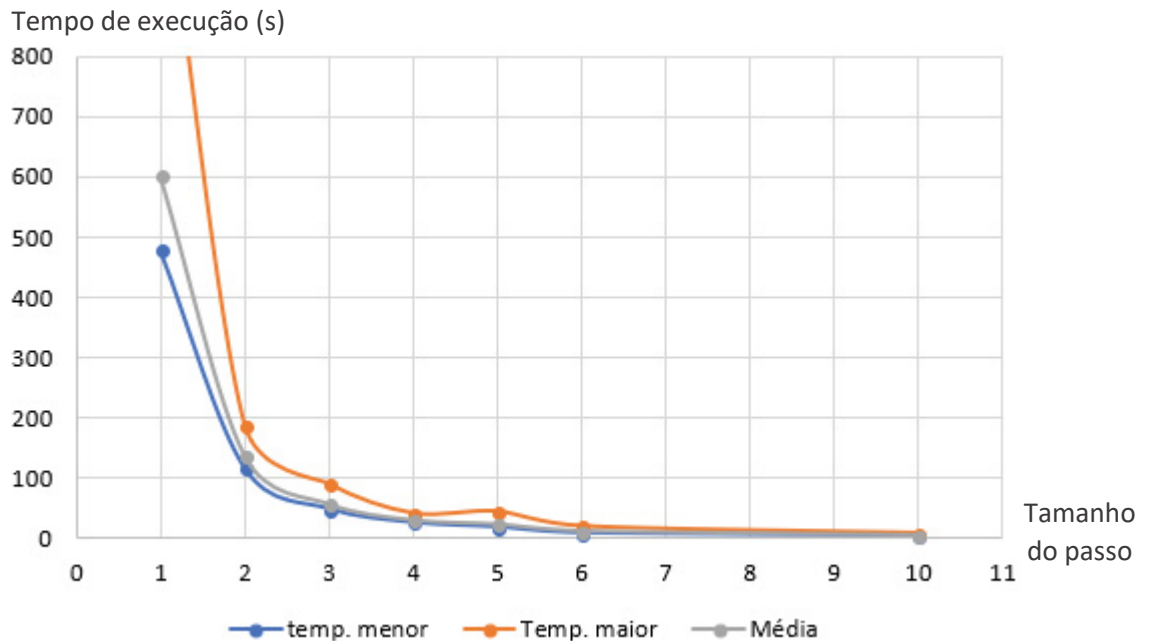
FIGURA 4.14 – GRÁFICO DAS VARIAÇÕES DE RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 35$ Km/h



FONTE: O autor (2018).

Analisando a variabilidade dos resultados para o passo de 10s na figura (4.14), nota-se uma melhora quando comparado com o passo 10s da figura (4.10).

FIGURA 4.15 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS TEMPOS DE EXECUÇÃO EM RELAÇÃO AO PASSO COM $v_m = 35$ Km/h



FONTE: O autor (2018).

Na figura (4.15) observa-se o comportamento dos menores, maiores e a média dos tempos de execução do programa, em que quanto maior for o tamanho do passo, muito menor se torna o tempo de execução.

A seguir são apresentados os resultados obtidos quando a velocidade média é de 45 km/h para o tamanho do passo igual a 1, 2, 3 e 4s na tabela (4.7) e passo igual a 5, 6 e 10s na tabela (4.8).

TABELA 4.7 – RESULTADOS COM $v_m = 45$ Km/h e $passo = 1, 2, 3$ e $4s$

Simulações	Passo 1s		Passo 2s		Passo 3s		Passo 4s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	44,538	543,098	51,960	120,443	57,197	127,723	68,781	24,970
2	44,371	1175,667	49,735	272,302	56,170	54,478	63,091	28,426
3	44,414	533,495	54,330	119,346	56,290	69,524	61,972	34,235
4	44,716	471,078	50,490	125,035	59,920	48,444	67,427	24,860
5	44,300	491,714	51,718	190,394	55,636	51,104	64,516	35,550
6	44,836	485,367	48,640	159,140	57,917	49,348	61,779	24,775
7	45,115	481,495	47,773	265,191	56,148	49,219	61,202	36,606
8	43,909	861,833	50,140	120,189	56,959	49,185	64,032	50,074
9	46,374	480,086	52,180	123,219	58,815	64,295	62,584	42,416
10	46,062	476,304	51,903	126,772	57,812	52,912	67,933	24,860
Menor	43,909	471,078	47,773	119,346	55,636	48,444	61,202	24,775
Maior	46,374	1175,667	54,330	272,302	59,920	127,723	68,781	50,074
Média	44,863	600,014	50,887	162,203	57,286	61,623	64,332	32,677
Desv. Padrão	0,747	221,996	1,822	57,476	1,277	23,042	2,626	8,281

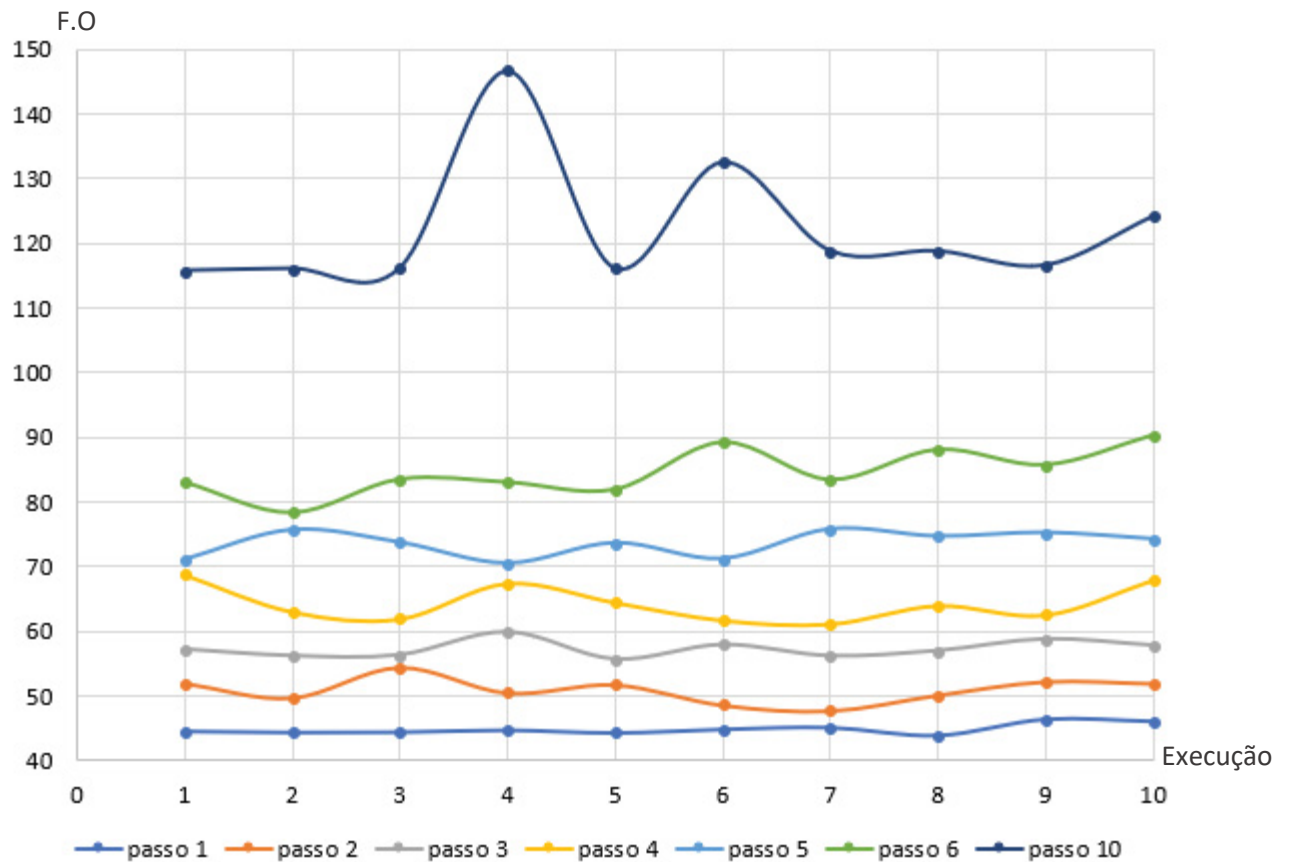
FONTE: O autor (2018).

TABELA 4.8 – RESULTADOS COM $v_m = 45$ Km/h e $passo = 5, 6$ e $10s$

Simulações	Passo 5s		Passo 6s		Passo 10s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	71,000	26,373	83,032	8,983	115,741	4,748
2	75,608	18,734	78,341	9,782	116,054	6,192
3	73,673	22,566	83,480	8,867	116,073	5,084
4	70,464	18,822	83,020	9,869	146,690	4,672
5	73,597	19,639	81,906	8,831	116,073	4,792
6	71,168	26,288	89,174	8,650	132,540	5,733
7	75,734	19,036	83,413	13,670	118,792	4,887
8	74,616	18,875	88,052	8,925	118,768	4,696
9	75,195	21,314	85,698	8,976	116,546	5,332
10	74,202	18,590	90,265	8,774	124,148	5,062
Menor	70,464	18,590	78,341	8,650	115,741	4,672
Maior	75,734	26,373	90,265	13,670	146,690	6,192
Média	73,526	21,024	84,638	9,533	122,142	5,120
Desv. Padrão	1,870	2,920	3,467	1,434	9,594	0,475

FONTE: O autor (2018).

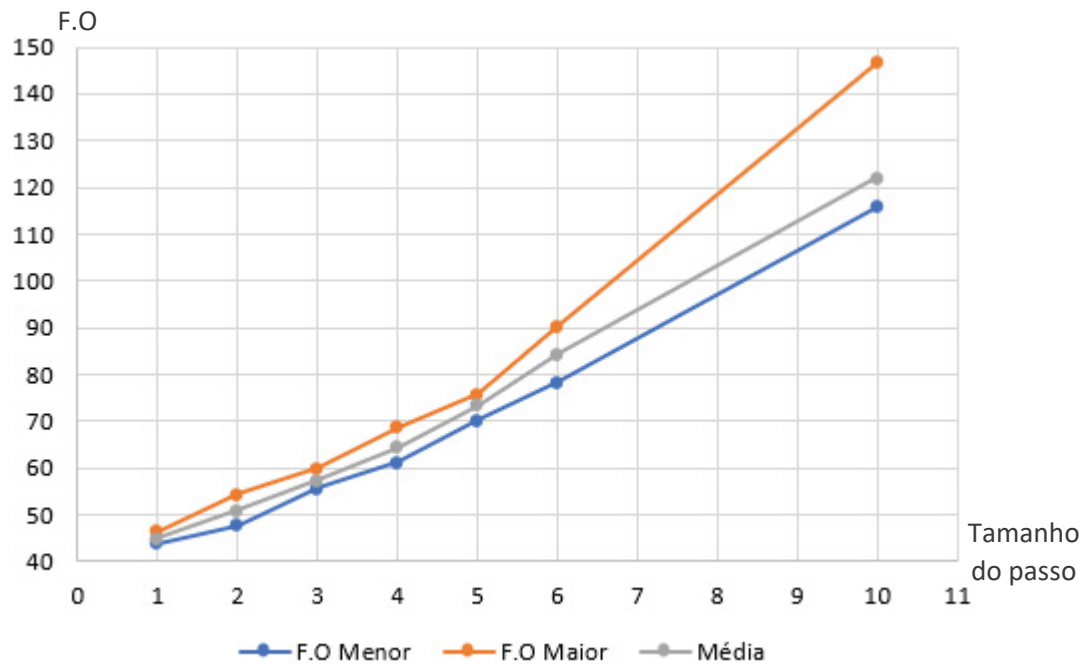
Usando as informações das tabelas (4.7) e (4.8) foi gerado os gráficos apresentados nas figuras (4.16) a (4.19).

FIGURA 4.16 – GRÁFICO DOS RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 45$ Km/h

FONTE: O autor (2018).

Analisando a figura (4.16), percebe-se uma redução das oscilações dos resultados na medida em que o tamanho do passo se torna menor. E que novamente ocorre uma diminuição das oscilações do passo de tamanho 10s quando comparada com os das figuras (4.8) e (4.12).

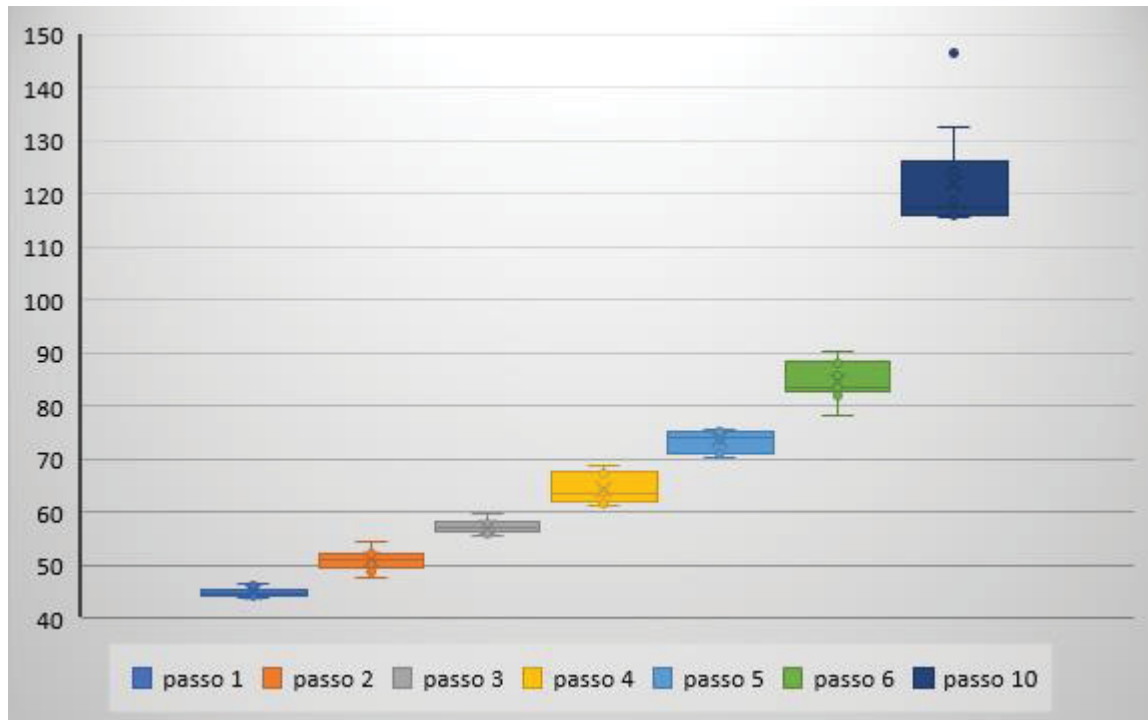
FIGURA 4.17 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS RESULTADOS DA F.O EM RELAÇÃO AO PASSO COM $v_m = 45$ Km/h



FONTE: O autor (2018).

Na figura (4.17) observa-se o comportamento dos menores, maiores e a média dos resultados da F.O, em que quanto maior o tamanho do passo, maiores são os resultados. E as funções que representam o menor resultado de cada tamanho de passo e a média dos resultados de cada passo, apresentam um comportamento quase linear.

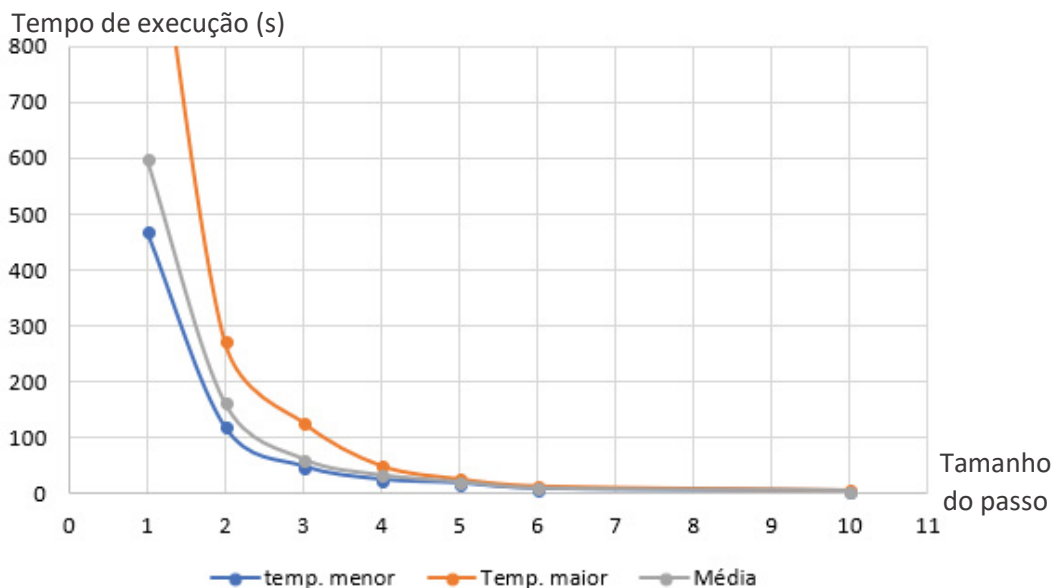
FIGURA 4.18 – GRÁFICO DAS VARIAÇÕES DE RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 45$ Km/h



FONTE: O autor (2018).

De acordo com o gráfico da figura (4.18) é possível notar uma perfeita simetria dos resultados para o passo de 2s, pois a diferença da mediana e a média é pequena e a distância da mediana para os quartis são muito próximas.

FIGURA 4.19 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS TEMPOS DE EXECUÇÃO EM RELAÇÃO AO PASSO COM $v_m = 45$ Km/h



FONTE: O autor (2018).

Na figura (4.19) observa-se o comportamento dos menores, maiores e a média dos tempos de execução do programa, em que quanto maior o tamanho do passo, muito menor se torna o tempo de execução. Por fim, são apresentados os resultados obtidos quando a velocidade média é de 60 km/h para o tamanho do passo igual a 1, 2, 3 e 4s na tabela (4.9) e passo igual a 5, 6 e 10s na tabela (4.10).

TABELA 4.9 – RESULTADOS COM $v_m = 60$ Km/h e *passo* = 1, 2, 3 e 4s

Resultados	Passo 1s		Passo 2s		Passo 3s		Passo 4s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	44,889	648,433	49,275	168,521	57,193	87,918	62,556	32,770
2	45,894	478,078	49,868	151,968	58,846	78,988	63,614	24,926
3	42,812	563,662	48,617	124,694	57,025	54,970	63,091	28,986
4	44,425	694,390	48,905	118,674	54,288	49,749	63,608	30,725
5	42,902	690,177	48,429	227,878	53,935	54,985	66,473	24,744
6	44,645	701,733	48,615	145,797	56,694	51,110	65,572	28,664
7	46,036	472,037	53,192	120,337	57,662	52,313	64,858	24,756
8	43,992	467,803	49,328	186,550	58,137	50,575	65,176	25,865
9	44,538	560,552	50,996	120,775	54,716	51,128	66,495	25,344
10	43,337	480,428	49,058	158,206	56,730	49,292	61,026	29,304
Menor	42,812	467,803	48,429	118,674	53,935	49,292	61,026	24,744
Maior	46,036	701,733	53,192	227,878	58,846	87,918	66,495	32,770
Média	44,347	575,729	49,628	152,340	56,523	58,103	64,247	27,608
Desv. Padrão	1,061	94,729	1,385	33,367	1,582	12,961	1,685	2,719

FONTE: O autor (2018).

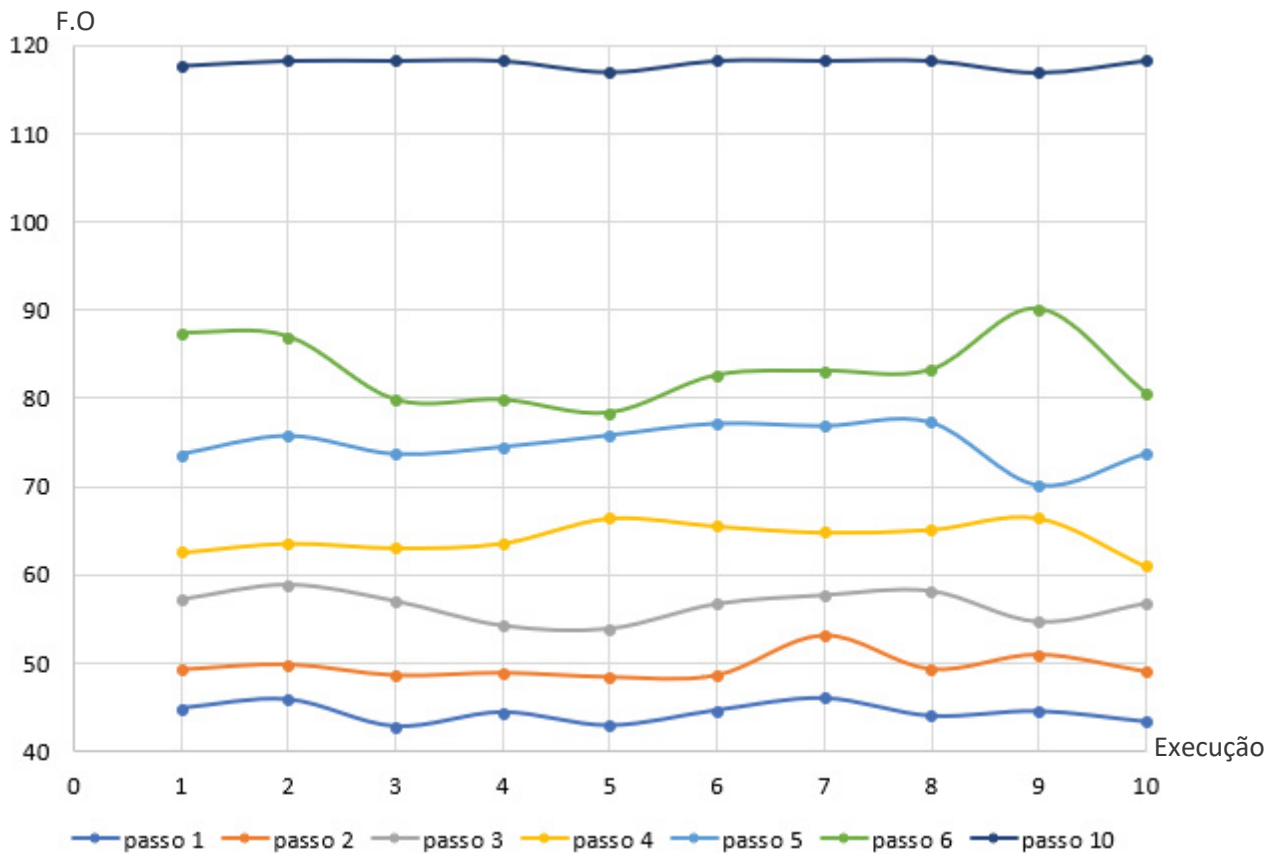
TABELA 4.10 – RESULTADOS COM $v_m = 60$ Km/h e *passo* = 5, 6 e 10s

Resultados	Passo 5s		Passo 6s		Passo 10s	
	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.	F.O(veíc.)	t. de proc.
1	73,625	20,528	87,417	10,326	117,716	4,648
2	75,744	18,412	86,987	12,480	118,295	5,390
3	73,651	25,339	79,851	9,712	118,295	4,973
4	74,464	18,647	79,851	8,646	118,295	4,558
5	75,802	18,502	78,382	9,061	117,020	6,240
6	77,147	18,628	82,616	9,531	118,295	4,576
7	76,864	25,701	83,099	8,917	118,295	4,558
8	77,274	18,411	83,218	8,663	118,295	4,774
9	70,071	22,573	90,113	9,659	116,951	4,594
10	73,666	19,822	80,647	18,285	118,295	4,558
Menor	70,071	18,411	78,382	8,646	116,951	4,558
Maior	77,274	25,701	90,113	18,285	118,295	6,240
Média	74,831	20,656	83,218	10,528	117,975	4,887
Desv. Padrão	2,103	2,732	3,639	2,797	0,524	0,517

FONTE: O autor (2018).

Usando as informações das tabelas (4.9) e (4.10) foi gerado os gráficos apresentados nas figuras (4.20) a (4.23).

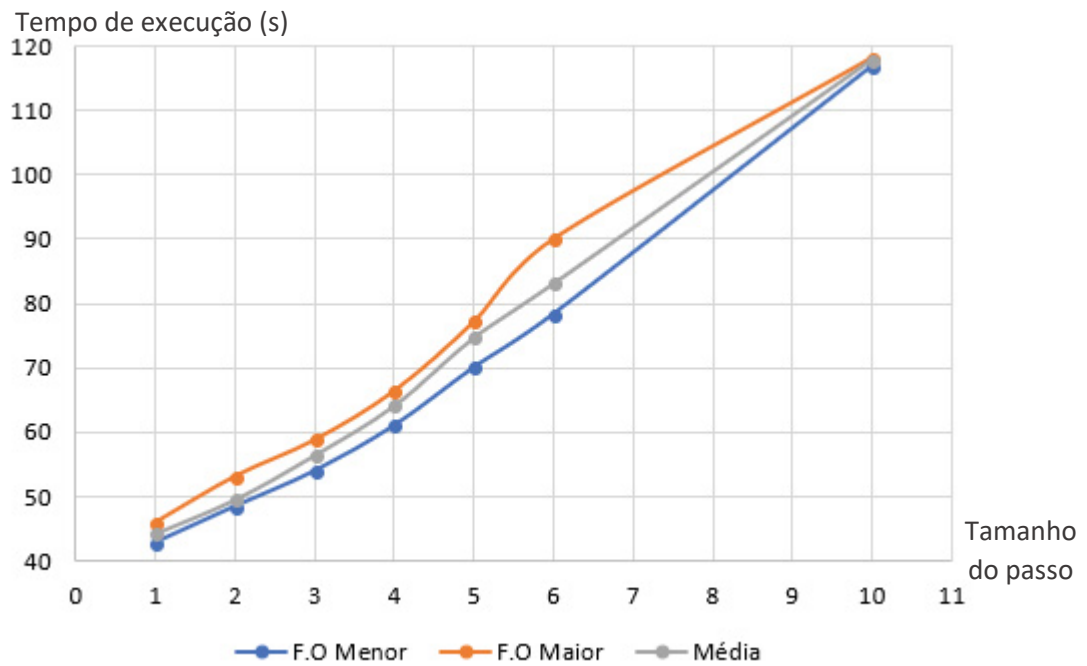
FIGURA 4.20 – GRÁFICO DOS RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 60$ Km/h



FONTE: O autor (2018).

Analisando a figura (4.20), percebe-se uma redução das oscilações dos resultados a medida em que o tamanho do passo se torna menor, exceto no caso em que o tamanho do passo é 10s, onde as oscilações foram mínimas.

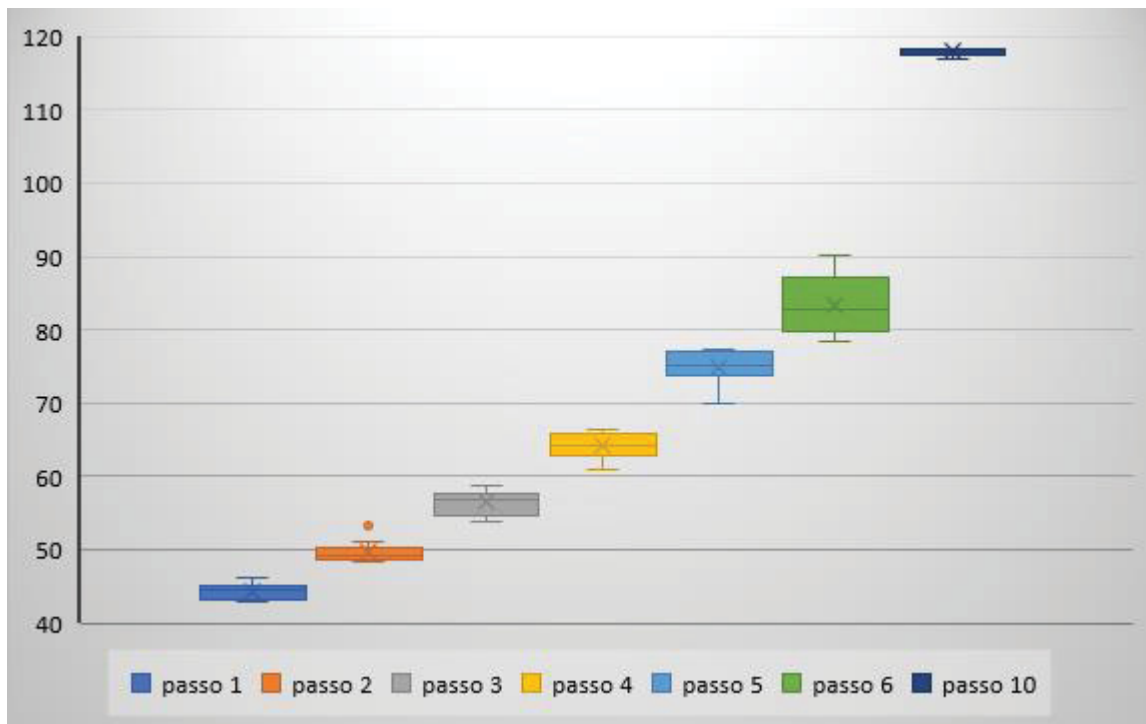
FIGURA 4.21 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS RESULTADOS DA F.O EM RELAÇÃO AO PASSO COM $v_m = 60$ Km/h



FONTE: O autor (2018).

Na figura (4.21) observa-se o comportamento dos menores, maiores e a média dos resultados da F.O, em que quanto maior for o tamanho do passo, maiores são os resultados. E as três funções apresentam um valor próximo no passo de tamanho de 10s.

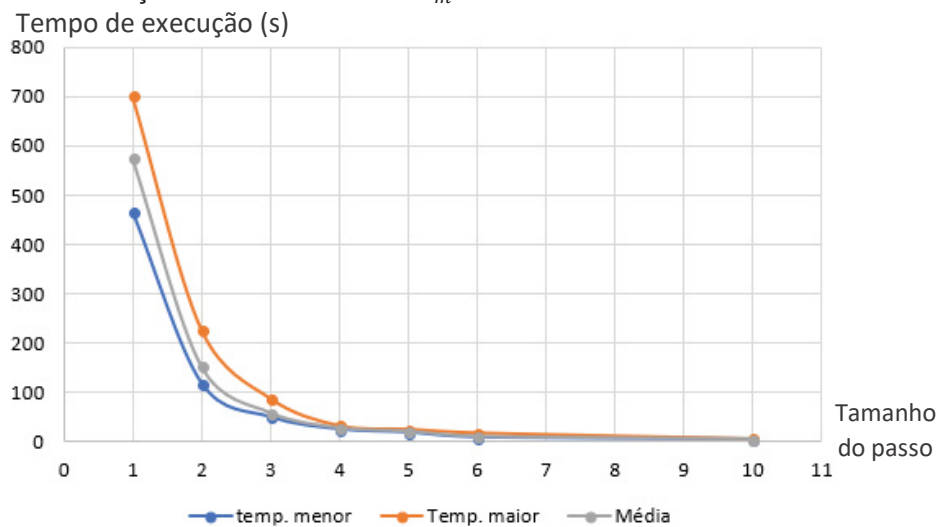
FIGURA 4.22 – GRÁFICO DAS VARIAÇÕES DE RESULTADOS DA F.O PARA CADA PASSO COM $v_m = 60$ Km/h



FONTE: O autor (2018).

O gráfico da figura (4.22) mostra uma redução quase linear dos resultados com a diminuição do tamanho do passo. E a variabilidade dos resultados são baixas para ambos os passos, onde se nota uma grande redução no passo 10s quando comparada com a figura (4.10).

FIGURA 4.23 – GRÁFICO DOS MENORES, MAIORES E A MÉDIA DOS TEMPOS DE EXECUÇÃO EM RELAÇÃO AO PASSO COM $v_m = 60$ Km/h



FONTE: O autor (2018).

Na figura (4.23) observa-se o comportamento dos menores, maiores e a média dos tempos de execução do programa, em que quanto maior for o tamanho do passo, menor se torna o tempo de execução.

Dentre as centenas de resultados gerados, conforme apresentados nas tabelas (4.1) a (4.10), nota-se que o melhor deles encontra-se na tabela (4.9) no passo de 1s quando a velocidade média é de 60 km/h. A tabela (4.11) exhibe a programação semafórica para cada cruzamento nos quatro períodos do dia adotados, onde os semáforos 1, 2 e 3 são referentes aos da via principal.

TABELA 4.11 – MELHORES TEMPOS DE VERDE E VERMELHO PARA CADA SEMÁFORO POR PERÍODO

Período	Semáforo 1		Semáforo 2		Semáforo 3	
	Aberto (s)	Fechado (s)	Aberto (s)	Fechado (s)	Aberto (s)	Fechado (s)
0h - 5h	86	34	72	48	67	53
5h - 9h	78	42	53	67	86	34
9h - 19h	72	48	53	67	87	33
19h - 0h	83	37	62	58	85	35
Valor da função objetivo (F.O) = 42,811466220 ≈ 43 veículos						

FONTE: O autor (2018).

LEGENDA: semáforo 1 – Linha Verde, esquina com a rua Oliveira Viana.

semáforo 2 – Linha Verde, esquina com a rua João Soares Barcelos.

semáforo 3 – Linha Verde, esquina com a rua Tenente Francisco Ferreira de Souza.

F.O – veículos acumulados.

Com base na tabela (4.11) foram gerados quatro diagramas de intervalos luminosos dos semáforos, sendo um diagrama para cada período considerado, conforme as tabelas (4.12) a (4.15) na sequência. O tempo de sinalização amarela em todos os períodos e semáforos é 4s, e o tempo de vermelho de limpeza é 1s para todos os casos. Tais valores são um arredondamento dos tempos que foram calculados para tais intervalos. Lembrando que o tempo de sinalização verde obtido nos resultados inclui em seu tempo, o tempo de sinalização amarela.

Os movimentos A e B representados como Mov. A e Mov. B nas tabelas a seguir, referem-se ao movimento na via principal (Mov. A) e ao movimento na via transversal (Mov. B) de cada cruzamento.

TABELA 4.12 – DIAGRAMA DE INTERVALOS LUMINOSOS DO 1º PERÍODO

Período	0h - 5h														
	Instantes (s)	0	63	67	68	72	73	74	82	86	87		115	119	120
Semáforo 1	Mov. A														
	Mov. B														
Semáforo 2	Mov. A														
	Mov. B														
Semáforo 3	Mov. A														
	Mov. B														
	Duração (s)	63	4	1	4	1	1	8	4	1	28	4	1		
	% do Ciclo ≈	53%	3%	1%	3%	1%	1%	7%	3%	1%	23%	3%	1%		

FONTE: O autor (2019).

TABELA 4.13 – DIAGRAMA DE INTERVALOS LUMINOSOS DO 2º PERÍODO

Período	5h - 9h															
	Instantes (s)	0	49	53	54		74	78	79	82	86	87		115	119	120
Semáforo 1	Mov. A															
	Mov. B															
Semáforo 2	Mov. A															
	Mov. B															
Semáforo 3	Mov. A															
	Mov. B															
	Duração (s)	49	4	1	20		4	1	3	4	1	28	4	1		
	% do Ciclo ≈	41%	3%	1%	17%		3%	1%	3%	3%	1%	23%	3%	1%		

FONTE: O autor (2019).

TABELA 4.14 – DIAGRAMA DE INTERVALOS LUMINOSOS DO 3º PERÍODO

Período	9h - 19h														
	Instantes (s)	0	49	53	54	68	72	73	83	87	88		115	119	120
Semáforo 1	Mov. A														
	Mov. B														
Semáforo 2	Mov. A														
	Mov. B														
Semáforo 3	Mov. A														
	Mov. B														
	Duração (s)	49	4	1	14	4	1	10	4	1	27	4	1		
	% do Ciclo ≈	41%	3%	1%	12%	3%	1%	8%	3%	1%	23%	3%	1%		

FONTE: O autor (2019).

TABELA 4.15 – DIAGRAMA DE INTERVALOS LUMINOSOS DO 4º PERÍODO

Período	19h - 0h													
	Instantes (s)	0	58	62	63	79	81	83	84	85	86	115	119	120
Semáforo 1	Mov. A													
	Mov. B													
Semáforo 2	Mov. A													
	Mov. B													
Semáforo 3	Mov. A													
	Mov. B													
	Duração (s)	58	4	1	16	4	1	1	1	29	4	1		
	% do Ciclo ≈	48%	3%	1%	13%	3%	1%	1%	1%	24%	3%	1%		

FONTE: O autor (2019).

Observando os diagramas exibidos nas tabelas (4.12), (4.13), (4.14) e (4.15), nota-se que para todos os períodos o 1º e 3º semáforos da via principal possui mais que 50% do seu tempo de ciclo com sinalização verde. E tal fato se nota para o 2º semáforo no primeiro e último período. Verificam-se também que os períodos 2 e 3 possuem uma programação semafórica semelhante.

Analisando individualmente as tabelas (4.1), (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), (4.9) e (4.10) nota-se na maioria dos casos uma tendência de aumento do desvio padrão da F.O, onde tal fato pode ser visto como uma sensibilidade maior na obtenção dos resultados. E observou-se uma redução do desvio padrão no tempo de execução, na medida em que o tamanho do passo aumenta. E percebe-se uma diminuição dos desvios padrões entre uma tabela e outra com o aumento da velocidade média.

Quando se observam os gráficos das figuras (4.4), (4.8), (4.12), (4.16), (4.20) nota-se grandes variações dos resultados da F.O para passos maiores, tanto que, se mostrou inconveniente representar a variação dos resultados para o $passo = 10s$ com $v_m = 15km/h$ devido ao tamanho das oscilações, pois quanto menor for a velocidade média, maiores são as variações.

E nos gráficos das figuras (4.5), (4.9), (4.13), (4.17), (4.21) observa-se uma tendência linearmente crescente da F.O em relação ao tamanho do passo, quando se aumenta a velocidade média.

Nos gráficos apresentados nas figuras (4.7), (4.11), (4.15), (4.19), (4.23) observa-se uma grande redução do tempo de execução do programa com o aumento

do *passo*. E as diferentes velocidades médias consideradas não obtiveram influências no comportamento dos gráficos. Portanto, foi possível estimar uma aproximação do tempo de execução em função do *passo*, da seguinte forma,

$$TE_{passo} \approx \frac{TE_1}{passo^2} \quad (5.1)$$

onde,

TE_{passo} : tempo de execução do programa para um *passo* [s];

TE_1 : tempo de execução do programa para um *passo* = 1s;

passo: tamanho do passo [s];

A equação (5.1) é restrita as características do algoritmo de otimização adotado deste trabalho, mas a redução do tempo de execução com o aumento do tamanho do passo é justificável pela diminuição do espaço de busca com o uso de passos maiores.

Novos resultados foram gerados para uma velocidade média do fluxo de 60km/h e tamanhos de passos de 1s a 6s, mas sem considerar nas simulações o “tempo morto” calculado de 6s. Tais resultados são apresentados nas tabelas (4.16) e (4.17) a seguir:

TABELA 4.16 – RESULTADOS COM $v_m = 60$ Km/h e *passo* = 1, 2 e 3s, SEM A INCLUSÃO DO TEMPO MORTO

Resultados	Passo 1s		Passo 2s		Passo 3s	
	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)
1	39,952	486,663	39,425	125,402	39,202	56,808
2	41,755	478,243	40,093	164,888	38,954	52,758
3	40,229	980,698	39,750	127,019	40,728	50,137
4	39,933	481,473	40,280	212,808	38,438	87,743
5	38,763	494,514	42,130	140,153	40,505	49,956
6	39,751	485,828	42,554	125,023	41,011	123,652
7	38,345	740,970	39,954	159,377	38,438	50,877
8	39,650	510,668	38,380	152,326	40,556	49,250
9	39,713	521,729	39,351	226,763	39,989	49,704
10	42,300	545,296	39,070	149,630	42,254	49,503
Menor	38,345	478,243	38,380	125,023	38,438	49,250
Maior	42,300	980,698	42,554	226,763	42,254	123,652
Média	40,039	572,608	40,099	158,339	40,007	62,039
Desv. Padrão	1,199	163,395	1,306	35,451	1,238	24,620

FONTE: O autor (2019).

TABELA 4.17 – RESULTADOS COM $v_m = 60$ Km/h e $passo = 4, 5$ e $6s$, SEM A INCLUSÃO DO TEMPO MORTO

Resultados	Passo 4s		Passo 5s		Passo 6s	
	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)	F.O(veíc.)	Tempo(s)
1	40,220	25,995	39,861	24,648	39,295	11,604
2	40,087	33,297	39,925	19,544	39,997	18,543
3	41,553	25,875	41,616	20,405	41,143	18,470
4	39,151	26,815	39,554	22,916	39,755	18,678
5	39,948	26,288	39,925	19,873	39,170	21,514
6	38,840	25,803	41,432	31,848	44,256	18,296
7	40,220	25,682	41,616	19,810	39,525	18,144
8	40,306	25,733	41,629	29,667	41,728	31,678
9	39,073	28,059	39,925	27,918	42,750	18,873
10	41,939	30,847	39,509	20,042	42,756	18,929
Menor	38,840	25,682	39,509	19,544	39,170	11,604
Maior	41,939	33,297	41,629	31,848	44,256	31,678
Média	40,134	27,439	40,499	23,667	41,037	19,473
Desv. Padrão	1,007	2,608	0,937	4,626	1,772	4,959

FONTE: O autor (2019).

Quando se estuda os resultados das tabelas (4.16) e (4.17), percebe-se que os valores ficam apenas entre 38 e 40, ou seja, o impacto do tamanho do passo é mínimo. E os desvios padrão são baixos, o que também acontece nas tabelas (4.9) e (4.10), onde a velocidade média do fluxo é igual.

Buscando-se entender os benefícios da utilização do método de busca VNS, foi gerada a tabela (4.18), onde se compara os resultados e tempos de processamento médios com o uso da heurística VNS em relação ao uso do método aleatório de busca. Os parâmetros considerados para obtenção da tabela (4.18) são: velocidade média de fluxo de 60km/h e tamanhos de passos de 1, 2, 3, 4, 5, 6 e 10s.

TABELA 4.18 – RESULTADOS DA HEURÍSTICA VNS E BUSCA ALEATÓRIA COM $v_m = 60$ Km/h e $passo = 1, 2, 3, 4, 5, 6$ e $10s$

	Passo 1s		Passo 2s		Passo 3s		Passo 4s		Passo 5s		Passo 6s		Passo 10s	
	F.O	Tempo	F.O	Tempo	F.O	Tempo	F.O	Tempo	F.O	Tempo	F.O	Tempo	F.O	Tempo
Média VNS	44,35	575,73	49,63	152,34	56,52	58,10	64,25	27,61	74,83	20,66	83,22	10,53	117,98	4,89
Média Aleat.	44,69	7897,94	50,34	1070,94	57,30	753,48	64,41	229,00	74,12	145,73	81,27	112,27	117,79	36,12
Melhoria(%)	1%	1272%	1%	603%	1%	1197%	0%	729%	-1%	606%	-2%	966%	0%	639%

FONTE: O autor (2019).

Os resultados na linha “Média VNS” da tabela (4.18) foram retirados da linha “Média” das tabelas (4.9) e (4.10). Para se compara o tempo de processamento do programa com a utilização de ambos os métodos de busca, foi considerado que os resultados da F.O fossem próximos para ambos os métodos, tal que, a diferença não ficasse maior que 2 veículos. Por tal motivo a porcentagem de melhoria da F.O na tabela (4.18) varia apenas de -2% a 1%.

5 CONCLUSÕES E RECOMENDAÇÕES

Conclui-se que pequenas mudanças no tamanho do passo não geram grandes melhoras na função objetivo quando o passo já é considerado pequeno. O que torna pouco vantajoso utilizar o menor passo, por exemplo, *passo* igual a 1 segundo, pois este implica em um elevado tempo de execução do programa em relação a passos que são até 2s maiores e geram resultados muito próximos dos obtidos com o *passo* igual a 1s. Portanto, após uma análise dos parâmetros, um tamanho de passo que oferece bons resultados e um tempo razoável de execução, é o *passo* igual a 3 segundos.

Quando a sensibilidade dos parâmetros utilizados no programa é analisada, nota-se que para mudanças na velocidade média para 10 km/h a mais ou a menos não serão gerados grandes impactos no fluxo, mas sim, a forma que se encontra a programação dos semáforos. Portanto, é melhor realizar uma programação mais eficiente nos semáforos do que realizar uma pequena mudança na velocidade permitida na via.

Com a inclusão do tempo morto nas simulações houve um aumento médio de 10,5% nos resultados, o que demonstra a importância de otimizar o tempo morto de maneira que a segurança seja preservada.

Conclui-se que a utilização da heurística VNS para a busca de um valor ótimo é pelo menos 7 vezes mais rápida que a busca aleatória para o modelo de simulação de tráfego adotado. E a heurística apresenta melhores resultados ainda, quando a região de busca se torna maior, onde tal fato ocorre nos passos pequenos.

Recomenda-se para trabalhos futuros a realização de um estudo para determinar a melhor quantidade de períodos e os seus intervalos na programação semaforica. E outro ponto importante a ser estudado é a complementação do modelo apresentado para realizar a sincronização dos semáforos com a utilização de um tempo de defasagem entre os mesmos, sendo tal tempo uma nova variável a ser considerada no modelo.

REFERÊNCIAS

- AKÇELIK, R. **Travel Time Functions for Transport Planning Purposes: Davidson's Function, Its Time-Dependent Form and an Alternative Travel Time Function**. Australian Road Research 21 (3), p. 49-59, 2000.
- ARAÚJO, S. C. **Controlador de tráfego: Semáforo inteligente**. Monografia (Graduação em Engenharia de Computação) – Setor de Ciências e Tecnologia, CENTRO UNIVERSITÁRIO DE BRASÍLIA, Brasília, DF, 2006.
- BOCTOR, F. **Discrete optimization and multi-neighbourhood, local improvement heuristic**, Doc. de travail 93-35, FSA, Université Laval, 1993.
- BONZANI, I.; CUMIN, L. M. G. **Critical analysis and perspectives on the hydrodynamic approach for the mathematical theory of vehicular traffic**. Elsevier: Mathematical and Computer Modelling, v. 50, p. 526-541, 2009.
- BRASIL. **Código de trânsito brasileiro**. Disponível em: <<https://www.jusbrasil.com.br/topicos/10632298/paragrafo-1-artigo-1-da-lei-n-9503-de-23-de-setembro-de-1997>> Acesso em: 20 nov. 2017.
- BRETHERTON, R. D.; BOWEN, G. T. **Latest Developments in SCOOT: SCOOT version 3.1**. Proceedings IEEE 6th International Conference on Road Traffic Control. London, 1996.
- BRETHERTON, R. D.; WOOD, K.; BOWEN, G. T. **SCOOT Version 4**. Proceedings IEEE 9th International Conference on Traffic monitoring and control. London, 1998.
- CARMO, A. R. do. **Programação semaforica uma proposta de otimização apoiada em modelo geo-temporal**. Campinas, SP [s.n.], 2016.
- CRISTIANI, E.; PICCOLI, B.; TOSIN, A. **Multiscale modeling of granular flows with application to crowd dynamics**. Society for Industrial and Applied Mathematics: Multiscale Modeling & Simulation, v. 9, p. 155-182, 2011.
- RIO PARANÁ. Disponível em: <<http://www.panorama.sp.gov.br/?pag=T1RjPU9EZz1PVFU9T0dVPU9HST1PVEE9T0dFPU9HRT0=&idmenu=264>>. Acesso em: 01 de junho de 2018.
- CUCCI, J. N. **Engenharia de tráfego urbano: O controle semaforico centralizado e a operação de campo**. Campinas: Mackenzie, 2012, 15 p. Notas de aula.
- CURITIBA. Prefeitura Municipal de Curitiba. **Linha Verde, evolução da linha integrada**. Disponível em: <<http://www.curitiba.pr.gov.br/fotos/album-prefeitura-retoma- hoje-leiloes-em-bolsa-para-ajudar-conclusao-da-linha-verde/28305>>. Acesso em: 24 nov. 2017.

DUTRA, C. B. **Avaliação da eficiência de métodos de coordenação semafórica em vias arteriais**. 203 f. Dissertação (Mestrado em Engenharia Civil) – Escola de São Carlos, Universidade de São Paulo, São Carlos, 2005.

FOLHA DE S. PAULO. **Falha em semáforo causa nó em cruzamento de avenidas em São Paulo**. Disponível em: <<http://www1.folha.uol.com.br/cotidiano/2017/02/1855266-falha-em-semaforo-causa-no-em-cruzamento-de-avenidas-em-sao-paulo.shtml>>. Acesso em: 25 nov. 2017.

GAZIS, D.; HERMAN, R.; MARADUDINT, A. The problem of the amber signal light in traffic flow. **Operations Research**, online V. 8, p.112–132, 1960.

GOBBO, A.F. **Proposta de aplicação do sistema de inferência Neuro-Fuzzi para otimização de tráfego**. Dissertação Mestrado, CEFET, Curitiba, PR, 2005.

GOOGLE MAPS. **Trecho da linha verde em Curitiba**. Acesso em 11 fev. 2019.

GONZAGA, E. W. L.; ALMEIDA, P. E. M.; CARRANO E. G. Otimização da Programação Semafórica em Tempo Real com Base em Modelagem Matemática. XII BRAZILIAN CONGRESS ON COMPUTATIONAL INTELLIGENCE, 2015, Curitiba - Paraná – Brasil, **Anais...**

GRAMANI, L. M. **Descrições microscópica, macroscópica e cinética do fluxo de tráfego veicular**. São Carlos: SBMAC, v. 46, 2010.

HEINEN, M. R.; SÁ, C. C. de; SILVEIRA, F. K.; CESCINETTO, C.; SOHN, G. Jr. Controle inteligente de semáforos utilizando redes neurais artificiais com funções de base radial. Anais do **EATI**, Frederico Westphalen – RS, n. 1 p. 38-45, 2013.

JONER, S.; VOLPI, N. M. P. Sincronização de semáforos: modelo matemático para uma aplicação ao tráfego de Curitiba. **Holos**, v. 2, 2013.

KAVISKI, E. **Comunicação Pessoal**. Professor do Departamento de Hidráulica e Saneamento, Universidade Federal do Paraná, Curitiba, 2017.

KIRKPATRICK, S.; GELATT, C.D.; VECCHI, M.P. Optimization by simulated annealing. *Science*, **220**, (4598), 1983, p.671– 680.

KREMER, G. M. **Uma introdução à equação de Boltzmann**. São Paulo: Ed. Universidade de São Paulo, 2005.

LACORTT, M.; KRIPKA, M.; KRIPKA, R.M.L. Modelos Matemáticos para Otimização do Tráfego Urbano SemafORIZADO. **Tema**, São Carlos, v. 14, n. 3, 2013.

LITTLE, J. C. The synchronization of traffic signals by mixed-integer linear programming. **Operations Research**, v. 14, p. 568-594, 1966.

LITTLE, J. C.; KELSON M. D.; GARTNER, N. H. Maxband: A program for setting signals on arteries and triangular networks. **Transportation Research Record**, v.795, p. 40-46, 1981.

LOUREIRO, C. F. G.; GOMES, M. J. T. L.; LEANDRO, C. H. P. **Avaliação do desempenho nos períodos de pico do tráfego de interseções semaforizadas com controle centralizado em tempo fixo e real**. Sinal de Trânsito. Disponível em: < <http://www.sinaldetransito.com.br/artigos/scoot.pdf> >. Acesso: 26 de fevereiro de 2018.

MLADENOVIC, M.; HANSEN, P. **Variable neighborhood search**. Computers and operations research. v. 24, ed. 11, p.1097–1100, 1997.

MOREIRA, A. Folha de S. Paulo. **Trânsito em São Paulo**. Disponível em: <<http://fotografia.folha.uol.com.br/galerias/11487-transito-em-sao-paulo>>. Acesso em: 27 nov. 2017.

OLIVEIRA, J. C. de. **Sincronização de Semáforos via minimização do acúmulo de veículos**. Monografia (Graduação em Matemática Industrial) – Setor de Ciências Exatas, Universidade Federal do Paraná, Curitiba, 2015.

PACHECO, C. A. C.; TURETTI, A. **A alavancagem da economia paranaense face ao incremento da arrecadação do imposto sobre a circulação de mercadorias e serviços gerados pelas indústrias de automóveis**. Monografia (Especialista em Auditoria Integral) – Setor de Ciências Sociais Aplicadas, Universidade Federal do Paraná, Londrina, 2003.

PAIVA NETO, P.M.; CASTRO NETO, M.M.; LOUREIRO, C.F.G. Progressão em Tempo Real versus Tempo Fixo por Banda de Passagem em Períodos de Baixa Demanda – Estudo de Caso. Anais do **XVII Congresso de Pesquisa e Ensino em Transportes**, ANPET, Rio de Janeiro, Comunicações Técnicas, CD-Rom, 2003.

PARAVISI, M.; MUSSE, S. R.; BICHO, A. de L. Modelagem e simulação do tráfego de veículos e controle semaforico em um ambiente virtual. **Vetor**, Rio Grande, v.16, n.2, p. 16-37, 2006.

PIAI, J.C.; CERVANTES, S.G. de. Um modelo para tráfego urbano e suas otimizações. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 17, 2010, Bonito. **Anais...**

PIETRANTONIO, H.; BONETTI, W. Jr. **Semáforos atuados pelo tráfego: Utilização, Parametrização, Monitoração e Análise de Desempenho**. São Paulo, 2003. 92p. Relatório de trabalho - Departamento de Engenharia de Transportes. Escola Politécnica da Universidade de São Paulo.

PINTO, N. et al. **Hidrologia Básica**. São Paulo: Edgard Blucher, Fundação Nacional de Material Escolar, 1976.

ROBERTSON, D.I. **Research on the Transyt and Scoot methods of signal coordination**. *ITE Journal*, em(CD-ROM) ITE Digital Library, 1997, (1986), 36–40.

ROMANI, M. **Materiais semafóricos: Especificação Técnica**. Secretaria Municipal de Trânsito – SETRAN. Paraná. Disponível em:

<<http://www.setran.curitiba.pr.gov.br/uploads/arquivos/e28bba51f7b6ca7f27944bfbd746a804f62e10bd.pdf>>. Acesso em: 25 de novembro de 2017.

ROSA, A. **Linha Verde**. Gazeta do Povo. Disponível em:

<<http://www.gazetadopovo.com.br/vida-e-cidadania/prefeitura-reempacota-obras-para-tornar-a-linha-verde-sustentavel-ck9cyjbrgd97ooqfexbaw0aye.>>. Acesso em: 24 nov. 2017.

SIEMENS. **SCOOT User Guide**. Disponível em:

<https://www.siemens.co.uk/traffic/pool/downloads/handbooks/utc/666_hf_16940_000.pdf>. Acesso em: 26 de fevereiro de 2018.

SOUSA, L. A. P. **Taxas de desaceleração e tempos de percepção e reação dos motoristas em interseções semaforizadas**. 90 f. Dissertação (Mestrado em Engenharia de Transportes) – COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

TRLSoftware. **Transyt 12**. Disponível em:

<https://trlsoftware.co.uk/support/products/transyt_12>. Acesso em: 26 de fevereiro de 2018a.

TRLSoftware. **Transyt 12**. Disponível em:

<https://trlsoftware.co.uk/support/products/transyt_13>. Acesso em: 26 de fevereiro de 2018b.

TRLSoftware. **Transyt 12**. Disponível em:

<https://trlsoftware.co.uk/support/products/transyt_14>. Acesso em: 26 de fevereiro de 2018.c

TRLSoftware. **Transyt 12**. Disponível em:

<https://trlsoftware.co.uk/support/products/transyt_15>. Acesso em: 26 de fevereiro de 2018d.

VILANOVA, L. M. **Fundamentos da programação semafórica**. Disponível

em: <<http://www.sinaldetransito.com.br/artigos/programacao.pdf>> Acesso em: 10 de janeiro de 2018.

VILANOVA, L. M. **SIRI: Um novo simulador para redes de semáforo**. Disponível

em: <http://meusite.mackenzie.br/professor_cucci/texto13.pdf>. Acesso em: 05 de dezembro de 2017.

VILANOVA, L. M. **SIRI: Um simulador mesoscópico para redes de semáforos**.

Sinal de trânsito, 2008. Disponível em:

<<http://www.sinaldetransito.com.br/artigos/siri.pdf>>. Acesso em: 22 de fevereiro de 2018.

VINCENT, R.A.; MITCHELL, A.I.; ROBERTSON, D.I. **User's Guide to TRANSYT Version 8**. Transport and Road Research Laboratory – TRRL, Laboratory Report 888, Crowthorne, Inglaterra, 1980.

GLOSSÁRIO

BANDA DE PASSAGEM: se caracteriza pelo ajuste dos instantes de abertura de cada semáforo, de forma a proporcionar a passagem de um pelotão por toda a via, mantendo-se uma certa velocidade média.

DEFASAGEM: É a diferença entre os inícios de sinalização verde de semáforos consecutivos de uma mesma via.

ESTÁGIO: É a parte do ciclo compreendida entre quaisquer dois inícios consecutivos de sinalização verde de um mesmo semáforo, ou seja, quanto um mesmo semáforo completa os três sinais luminosos, sendo eles, verde, amarelo e vermelho.

FASE: É associada ao circuito elétrico que alimenta o conjunto de lâmpadas da mesma cor de um determinado grupo semafórico.

LINK: É um termo usado para representar um grupo de trânsito em atividades relacionadas a esquematização de sistemas, simuladores e trabalhos gráficos.

MONTADOR (ASSEMBLER): é o programa do sistema responsável por traduzir o código *assembly* em linguagem de máquina.

PASSO: Intervalo de tempo dado em segundo, o qual ocorre as simulações de um programa e que também é usado para discretizar os dados de entrada do programa.

PELOTÃO: Um grupo de veículos próximos uns aos outros que se deslocam conjuntamente em um mesmo sentido de uma via.

PLANO: Denomina-se plano semafórico, ou simplesmente plano, ao conjunto de parâmetros introduzidos no controlador ou na central de controle que determina a sequência e as durações dos tempos exibidos pelo semáforo. O período de vigência de um plano está estabelecido na tabela horária.

TEMPO DE CICLO: Tempo para completar uma sequência inteira dos estágios de um semáforo em que cada um dos estágios obrigatórios ocorre, uma vez para cada semáforo de um cruzamento. Vide figura (1).

FIGURA 1 – Ciclo



FONTE: O autor (2019).

Mov. A e Mov. B: Representam movimentos conflitantes dos veículos que disputam um mesmo cruzamento.

TRECHO: Espaço de uma via compreendido entre dois cruzamentos consecutivos.

APÊNDICE 1 – CÓDIGO DO PROGRAMA ELABORADO EM PASCAL

```

program sr3N3_vns_passos_tperdido;

uses
  wincrt, sysutils;
var
  arq2 : text;
  i: byte;
  resultado: single;
  ti, tf, tt: tdatetime;

function simula():single;

type
  vetr = array[0..120000] of single;
  vets = array[1..3] of single;
  vetb = array[1..10] of byte;
  veti = array[0..10] of integer;

  regd = record
    fob : single;
    dt : array[-1..3,1..3,0..1] of byte; {período,semáforo,estado}
  end;

  regv = record { fluxos e armazenamentos }
    qre,
    qrs,
    qts,
    st,sr : vets;
  end;

var
  arq : text;
  qe : array[1..3] of ^vetr;
  vd,
  vt,
  vdm : regd;
  s : vetb;
  hp,
  tl,
  ts : veti;
  p : shortint;
  i,b,continua,contador,raio,vermelhoMin,vermelhoMax,semente,ing,w,ciclo
: byte;
  k,sol_ale, inicio : integer;
  qft,
  qfr : single; { fluxo máximo nos semáforos }
  vv : array[0..1] of regv;

const
  vf : single = 60.0; { velocidade média no semáforo - km/h }
  t_ciclo :byte = 120; {o tempo de ciclo deve ser um divisor de 3600}
  passo : integer = 1; { o tamanho do passo deve ser um divisor do tempo
de ciclo}
  t_perdido_princ: byte = 3;{tempo perdido do inicio ao fim do amarelo +
metade do tempo de vermelho limpeza nos semaf. da principal}
  t_perdido_secund: byte = 3;{tempo perdido do inicio ao fim do amarelo +
metade do tempo de vermelho limpeza nos semaf. da secundaria}

```

```

arr : vets  = (0.8,0.8,0.8); {proporção de veic. que viram da
trensversal p/ a principal}
atr : vets  = (0.5,0.5,0.5); {proporção de veic. que viram da
principal p/ a transversal}

procedure calc_qe;

var
  arq : text;
  i   : byte;
  a,j,k,fin,ultimo : integer;
  s   : string;

begin

{-----          Leitura dos dados          -----}
---}

  assign(arq,'C:\Users\Danilo\Documents\Danilo\facul
Danilo\Mestrado\Programa alt_VNS_passo e t perdido\lverde.txt');
  reset (arq);

  for i := 1 to 3 do
    begin

      readln(arq,s);

      for j := 0 to 23 do { percorre cada horas do dia}
        begin

hora}          k := 3600*j div passo;          {Inicio de cada intervalo de 1
              readln(arq,a,qe[i]^[k]);

              qe[i]^[k] := (qe[i]^[k]*passo)/3600; { veículos/passo }
              fin := k +(3600 div passo)-1; {Fim de cada intervalo de 1 hora}
              for a := k+1 to fin do
                qe[i]^[a] := qe[i]^[k];

              end;
              ultimo := 86400 div passo;
              qe[i]^[ultimo] := qe[i]^[0];
              end;

      close(arq);

    end;

{-----}
---}
procedure armaz_a(qe1,qe2,qs1,v1,qf:single; var qs2,v2:single);
{
  Sol. da eq. da cons. de massa - semáforo aberto
}
var
  ax : single;

begin

  ax := 0.5*(qe1 + qe2 - qs1) + v1;

  if ax < 0.0

```

```

    then
      ax := 0.0;

    if ax < qf
      then
        qs2 := ax
      else
        qs2 := qf;

    v2 := ax - qs2;

  end;

procedure armaz_f(qe1,qe2,qs1,v1:single; var qs2,v2:single);
{
  Sol. da eq. da cons. de massa - semáforo fechado
}
begin

  qs2 := 0.0;
  v2 := 0.5*(qe1 + qe2 - qs1) + v1;

  if v2 < 0.0
    then
      v2 := 0.0;

  end;

procedure calc(p:byte);
{ p = período }
var
  t : integer;
  a,
  i,k : byte;
  fob : single;

begin

  for t := t1[p]+1 to t1[p+1] do
    begin

      a := b;
      b := 1 - b;

      vv[b].qre[1] := qe[1]^t;

      for i := 1 to 3 do { determina o estado dos 3 semáforos }
        begin

          k := i + 1;

          if t > ts[i] then { alterna estado do semáforo }
            begin
              inicio:=ts[i];
              s[i] := 1 - s[i];
              ts[i] := ts[i] + vd.dt[p,i,s[i]];
            end;

          if s[i] <> 0 then
            begin { semáforo aberto principal }

```

```

principal}
    if t > inicio + t_perdido_princ then {Tempo perdido
        armaz_a(vv[a].qre[i],vv[b].qre[i],vv[a].qrs[i],
        vv[a].sr[i],qfr,vv[b].qrs[i],vv[b].sr[i])
    else
        armaz_f(vv[a].qre[i],vv[b].qre[i],vv[a].qrs[i],
        vv[a].sr[i],vv[b].qrs[i],vv[b].sr[i]);
    {----- Fim tempo perdido -----}
    if i < 3 then
        begin
            vv[b].qre[k] := arr[k]*vv[b].qrs[i];

            armaz_f(qe[k]^[t-1],qe[k]^[t],vv[a].qts[k],
            vv[a].st[k],vv[b].qts[k],vv[b].st[k]);
        end;

    end
    else
        begin { semáforo fechado principal }

            armaz_f(vv[a].qre[i],vv[b].qre[i],vv[a].qrs[i],
            vv[a].sr[i],vv[b].qrs[i],vv[b].sr[i]);

            if i < 3 then
                begin
                    if t > inicio + t_perdido_secund then {Tempo
perdido principal}
                        begin
                            armaz_a(qe[k]^[t-
1],qe[k]^[t],vv[a].qts[k],
vv[a].st[k],qft,vv[b].qts[k],vv[b].st[k]);

                                vv[b].qre[k] := atr[k]*vv[b].qts[k];
                            end
                        else
                            armaz_f(qe[k]^[t-1],qe[k]^[t],vv[a].qts[k],
                            vv[a].st[k],vv[b].qts[k],vv[b].st[k]);
                        {----- Fim tempo perdido -----}
                            end;

                    end;
                end;
            end;

            fob := 0.0;
            for i := 1 to 3 do
                fob := fob + vv[b].sr[i] + vv[b].st[i];

            if fob > vd.fob then
                vd.fob := fob; {Armazena o maior numero de veiculos retidos em
um ciclo no periodo p}

            end;

        end;

    {----- Final do procedimento Calc. -----}
    ----}

Procedure executa();

```

```

var
  i,p:byte;
  begin

    p := 0;
    for i := 1 to 3 do
      begin
        ts[i] := t1[0] + vd.dt[-1,i,0];
        s [i] := 0; { estado do semáforo p/rua principal: 0 = fechado;
1 = aberto }
      end;

    b := 0; { cond.inicial(t=0) - sem. fechados - ruas vazias com
fluxos <> zero }

    vv[b].qre[1] := qe[1]^0;
    vv[b].qts[2] := qe[2]^0;
    vv[b].qts[3] := qe[3]^0;

    vv[b].qre[2] := atr[2]*vv[b].qts[2];
    vv[b].qre[3] := atr[3]*vv[b].qts[3];

    for i := 1 to 3 do
      begin
        vv[b].qrs[i] := 0.0;
        vv[b].st [i] := 0.0;
        vv[b].sr [i] := 0.0;
      end;

    vd.fob := 0.0; {Inicializa o n° de veic. retidos}
    for p := 0 to 3 do
      calc(p);

      (*writeln(arq,k:7,vd.fob);*) {Imprime cada resul. encontrado}

      {Verifica se o valor da F.O atual é menor que a anterior, se sim,
armazenda o menor valor}
      if vdm.fob > vd.fob then

        vdm := vd;

    end;

  begin

    for i := 1 to 3 do
      new(qe[i]);

    calc_qe;

    qft := passo*vf/(2.5*3.6);
    qfr := 3.0*qft/2;           { veic/passo na principal }
    qft := 2.0*qft/3;          { veic/passo na transversal }

    {----- definição dos períodos -----}
    --}
    {Hora de inicio de cada periodo}
    hp[0] := 0;
    hp[1] := 5;

```

```

hp[2] := 9;
hp[3] := 19;
hp[4] := 24;

{inicio de cada periodo em passos}
for i := 0 to 4 do
  tl[i] := (3600 div passo)*hp[i];
{-----}
-}
assign (arq, 'C:\Users\Danilo\Documents\Danilo\facul
Danilo\Mestrado\Programa alt_VNS_passo e t perdido\teste.txt');
rewrite(arq);
randomize;
vdm.fob := 1.0e20; {Inicialização da função obj. com um valor grande}

{----- Intervalo de Vermelho -----}
----}

vermelhoMin := 30 div passo; {tempo minimo de vermelho}
semente := 41 div passo; {tempo de vermelho maximo a mais que o minimo}
VermelhoMax := 70;
ing := round(10 div passo); {ingremento para gerar vizinhança}
ciclo := t_ciclo div passo; {Tempo de ciclo em relação ao passo}
sol_ale := 12000 div passo;

{----- Encontra melhor solução inicial para n iterções -----}
- }

for k := 1 to sol_ale do
  begin
    { sorteio das variáveis de estado }
    for p := -1 to 3 do { p = período }
      for i := 1 to 3 do { i = local }
        begin
          vd.dt[p,i,0] := vermelhoMin + random(semente); { s = 0 -
fechado }
          vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
          end;
        executa;
      end;

    continua:=0;
    contador:=0;
    vt:=vdm;
    while continua <= 20 do
      begin
        {----- Vizinho a cima -----}
        ---}
        for w := 1 to ing do
          for p := -1 to 3 do { p = período }
            for i := 1 to 3 do { i = local }
              begin
                vd.dt[p,i,0] := vt.dt[p,i,0]+ w; { s = 0 - fechado }

                if vd.dt[p,i,0] > vermelhoMax then
                  vd.dt[p,i,0]:=vd.dt[p,i,0] - w
                else
                  begin
                    vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
                    executa;
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;

```

```

end;

{-----Vizinho a baixo-----}
---}
for w := 1 to ing do
  for p := -1 to 3 do { p = período }
    for i := 1 to 3 do { i = local }
      begin
        vd.dt[p,i,0] := vt.dt[p,i,0]- w; { s = 0 - fechado }

        if vd.dt[p,i,0] < vermelhoMin then
          vd.dt[p,i,0]:=vd.dt[p,i,0] + w
        else
          begin
            vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
            executa;
          end;
        end;
      end;
end;

{-----}
{----- Vizinho a cima em tudo -----}
---}
for w:= 1 to ing do
  for p := -1 to 3 do { p = período }
    begin
      for i := 1 to 3 do { i = local }
        begin
          vd.dt[p,i,0] := vt.dt[p,i,0]+w; { s = 0 - fechado }

          if vd.dt[p,i,0] > vermelhoMax then
            vd.dt[p,i,0]:=vd.dt[p,i,0] -w;

            vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
          end;
          executa;
        end;
      end;
end;

{-----Vizinho a baixo em tudo -----}
----}
for w:= 1 to ing do
  for p := -1 to 3 do { p = período }
    begin
      for i := 1 to 3 do { i = local }
        begin
          vd.dt[p,i,0] := vt.dt[p,i,0]- w; { s = 0 - fechado }

          if vd.dt[p,i,0] < vermelhoMin then
            vd.dt[p,i,0]:= vd.dt[p,i,0] + w;

            vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
          end;
          executa;
        end;
      end;
end;

{-----}
{----- Vizinho a cima em tuda vertical -----}
---}

```

```

for w:= 1 to ing do
  for i := 1 to 3 do { i = local  }
  begin
    for p := -1 to 3 do { p = período }
    begin
      vd.dt[p,i,0] := vt.dt[p,i,0]+ w;    { s = 0 - fechado }

      if vd.dt[p,i,0] > vermelhoMax then
        vd.dt[p,i,0]:= vd.dt[p,i,0] - w;

      vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
    end;
  executa;
end;

{-----Vizinho a baixo em toda vertical -----}
----}
for w:= 1 to ing do
  for i := 1 to 3 do { i = local  }
  begin
    for p := -1 to 3 do { p = período }
    begin
      vd.dt[p,i,0] := vt.dt[p,i,0]- w;    { s = 0 - fechado }

      if vd.dt[p,i,0] < vermelhoMin then
        vd.dt[p,i,0]:=vd.dt[p,i,0] + w;

      vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
    end;
  executa;
end;

{-----}
---}
continua:=continua +1;
if vt.fob > vdm.fob then
  begin
    continua:=0;
    vt:=vdm;
    contador:=contador +1;
  end
else
  begin
    for k := 1 to 100*ing do
    begin

      for p := -1 to 3 do { p = período }
      for i := 1 to 3 do { i = local  }
      begin
        raio:=random(vermelhoMin);
        vd.dt[p,i,0] := vt.dt[p,i,0] + raio;    { s = 0 - fechado }

        if vd.dt[p,i,0] > vermelhoMax then
          vd.dt[p,i,0]:= vd.dt[p,i,0]- raio;

        vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto }
      end;
    executa;
  end;
end;

```

```

end;

{ sorteio aleatorio da vizinhança para - }
for k := 1 to 100*ing do
begin
for p := -1 to 3 do { p = período }
for i := 1 to 3 do { i = local }
begin
raio:=random(vermelhoMin);
vd.dt[p,i,0] := vt.dt[p,i,0] - raio; { s = 0 - fechado }

if vd.dt[p,i,0] < vermelhoMin then
vd.dt[p,i,0]:= vd.dt[p,i,0] + raio;

vd.dt[p,i,1] := ciclo - vd.dt[p,i,0]; { s = 1 - aberto
}
end;
executa;
end;

end;

if vdm.fob <= vt.fob then
vt:=vdm;

{-----}
---}
end;
{----- Imprime resultado -----}
---}
writeln(arq);
writeln(arq, 'Fob min', vdm.fob);

for p := -1 to 3 do
begin
writeln(arq);
write (arq,p:2);
if p < 0
then
write(arq, ' ')
else
write(arq, ' Início', hp[p]:3, 'h');
for i := 1 to 3 do
write(arq, '
Sem.', i:1, passo*vdm.dt[p,i,1]:4, 'A', passo*vdm.dt[p,i,0]:4, 'F');
end;
writeln(arq);
writeln(arq, contador:2);

close(arq);
simula:= vdm.fob;
end;

begin

assign (arq2, 'C:\Users\Danilo\Documents\Danilo\facul
Danilo\Mestrado\Programa alt_VNS_passo e t perdido\R_passols.txt');
rewrite(arq2);

for i:= 1 to 1 do
begin

```

```
ti:=now;
resultado:= simula();
tf:=now;
tt:= tf - ti;
writeln(arq2,' ',resultado:2:8, ' O tempo : ', tt*100000:3:3);
end;
close(arq2);
end.
```