



Algoritmo para priorização de pedestres em intersecções semaforizadas

Gomes LB*, Marte CL*, Amadio AT*, Siqueira CS*, Silva IBdeO* e Santos IRdos*

**Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de Transportes.*

Resumo. Tendo em mente o alto número de atropelamentos registrados na cidade de São Paulo e a necessidade de aumentar a segurança para pedestres em intersecções semaforizadas, o presente estudo visa à criação de uma base de dados a ser usada para treinar uma rede neural, cuja função será implementar um algoritmo para os controladores semaforicos, de forma a priorizar o fluxo de pedestres. Para isto, foi criado um micromodelo de tráfego no software VISSIM, da empresa alemã PTV, no qual foram propostos 5.320 cenários futuros. Estes foram gerados programando o VISSIM com um script, em linguagem Python, através de interface COM. A análise destes cenários gerou uma grande quantidade de dados, como: velocidades, tempo de espera de pedestres e veículos, assim como a formação de filas de veículos. Esses dados são usados para treinar a inteligência artificial de aprendizado profundo – a rede neural. Este estudo demonstrou que a automação de tarefas, através da interface COM, permite obter um número elevado de análises em tempo reduzido, além de dados que podem ser utilizados em outras análises, ou para o preparo de novas ferramentas, como a que se busca neste artigo.

Palavras-chave. *Simulação, micromodelo, VISSIM, Python, Interface COM.*

Abstract. Bearing in mind the high number of pedestrians being run over in the city of São Paulo and the need to increase pedestrian safety at traffic light intersections, this study aims to create a database to be used as feed for a neural network meant to implement an algorithm for traffic light controllers that prioritizes pedestrians. To reach this goal, a micromodel was created in the VISSIM software, from german PTV, and 5320 future scenarios were proposed, which were generated by programming VISSIM using Python language through the COM interface. The analysis of these scenarios generated a large amount of data - such as velocities and waiting times of pedestrians and vehicles and queuing for vehicles - data that is used to train the deep learning artificial intelligence - the neural network. This study demonstrated that the automation of tasks through the COM interface allows a high number of analyzes in a short time, enabling the collection of a large amount of data that can be used in other analyses, or for the preparation of new tools, such as the one sought in this work.



Keywords. *Simulation, micromodel, VISSIM, Python, COM interface.*

Introdução.

Acidentes. Tráfego motorizado e travessia de pedestres podem muitas vezes ser conturbados, principalmente se não houver sinalização e semaforização adequadas. Juntando-se a isto grandes tempos de espera, quando os pedestres são induzidos a tentar atravessar as vias em lugares e situações indevidas, aumentando potencialmente o risco de acidentes. Tempos semafóricos curtos demais para pedestres com dificuldades de locomoção aumentam ainda mais o risco de acidentes, muitas vezes fatais.

De 2000 até 2014, 48.879 internações por atropelamentos foram registradas em hospitais da cidade de São Paulo: 31,5% das vítimas possuíam 60 anos ou mais e 22,6%, até 19 anos. O gasto do Sistema Único de Saúde (SUS) com as vítimas destes atropelamentos superou 65 milhões de reais (LECI RODRIGUES et al., 2018). Já em 2019, dos registros de acidentes de trânsito na capital paulista, 44,9% dos fatais e 19,6% dos não fatais foram devidos a atropelamentos (OBSERVATÓRIO MOBILIDADE SEGURA, 2021).

Um possível encaminhamento de solução diminuir significativamente esse número de acidentes e alcançar o que a Prefeitura da Cidade de São Paulo (PMSP) e seu órgão gestor do trânsito, a Companhia de Engenharia de Tráfego (CET/SP): morte zero no trânsito, é trabalhar com o conceito de ruas completas.

Ruas Completas. SMITH et al. (2010) conceituam ruas completas como ruas desenhadas para acomodar com segurança usuários de todos os modais e com todas as capacidades e dificuldades de locomoção, sendo vistas como uma potencial solução para a coexistência de diversos modos de transportes.

Alguns elementos comuns em uma rua completa seriam calçadas, ciclofaixas/ciclovias, calçadas compartilhadas, faixas exclusivas para ônibus, paradas de ônibus seguras e acessíveis e travessias de pedestres frequentes e seguras. No entanto, é preciso notar que o design de uma rua completa deve atender ao seu entorno: uma rua completa em um espaço rural não terá o mesmo design de uma rua completa em espaço urbano. Por exemplo, devido às maiores velocidades, seria mais intuitivo pensar em ciclofaixas ao invés de ciclovias. Mas todas terão a preocupação de balancear segurança e conveniência a todos os usuários da via. (United States Department of Transportation, 2010)

Dos elementos acima citados, daremos ênfase neste artigo às travessias de pedestres.

Travessia de pedestres. COEUGNET et al. (2019) analisaram os fatores que influenciam a decisão de atravessar a faixa por parte dos pedestres. O tempo de espera mostrou-se especialmente decisivo, bem como os prazos dados aos pedestres – quanto mais tempo se passa esperando para atravessar, menores são os intervalos entre veículos que os pedestres aceitam para se arriscarem na travessia. Ou seja, pessoas com prazos apertados (e, portanto, com pressa) tendem a se arriscar mais, atravessando mesmo com curto intervalo entre veículos. Além disso,

há também a influência social: quando um pedestre vê o outro atravessar, ele tende a segui-lo, e se um motorista acena para alguém que está esperando na faixa, a pessoa que recebe o sinal tende a atravessar. Ambos os casos se verificam mesmo que o semáforo esteja vermelho.

ZHANG et al. (2017) criaram um esquema de temporização semafórica levando em consideração tanto os veículos quanto os pedestres. Os dois tipos de fluxos foram modelados matematicamente e então inseridos em uma função quadrática, cujo papel era gerar a programação semafórica de uma interseção. Os resultados mostraram que, com o método proposto, a programação semafórica poderia ser gerada em tempo real, considerando um horizonte de 15 segundos para a previsão de um estado, de forma a permitir que um dos dois modos seja liberado, sem influir significativamente no tempo de travessia do outro modo.

De acordo com TIAN et al. (2018), podem-se utilizar vários parâmetros para simular diferentes condições de vias. Dados os programas residentes no controlador semafórico, o banco de testes conduz a simulação e fornece medidas de desempenho, incluindo o tempo de espera de veículos e pedestres.

Eles realizaram testes de controle semafórico inteligente considerando seis elementos: a via, as faixas de pedestres, os veículos, os próprios pedestres, os controladores semafóricos e o simulador de tráfego. No estudo, foram analisados dois cenários: um com controle semafórico de Tempo Fixo e outro com o Controle Adaptativo dos autores. A simulação contava com 1804 segundos de viagem, fluxo de 1000 veículos, sendo 500 para cada sentido da via, e 5 semáforos implantados de forma aleatória. Os pedestres foram gerados com uma probabilidade de 0,001 por segundo e tem até 20 segundos para atravessar. Caso o semáforo tenha menos de 20 segundos de verde disponíveis, o pedestre espera até que o semáforo abra novamente.

O controle adaptativo possui a seguinte lógica: sempre que um veículo vai chegar em até 10 segundos no semáforo e não há pedestres no cruzamento, o semáforo dá 10 segundos de verde para o veículo, a menos que ele já esteja no tempo de verde – se for este o caso, não há prolongamento do tempo de verde. Caso contrário, ele segue o tempo fixo pré-estabelecido (mudança de estado semafórico a cada 60 segundos).

Todas as medidas de desempenho sofreram reduções significativas no cenário de controle adaptativo quando com parado ao cenário de tempo fixo. O tempo de espera médio para veículos passou de 91,998s para 1,664s – redução de 98,19% – o tempo máximo de espera para veículos passou de 293,637s para 29,134s – redução de 90,08%. O número médio de paradas passou de 3,195 para 0,175 – redução de 94,52%; o máximo, de 5 para 2, reduzindo em 60%. Tempo de espera médio para pedestres passou de 24,571s para 7,071s, reduzindo em 71,22%, e o máximo, de 66s para 49s, com uma redução de 25,76%.

Proposta. Neste artigo vamos mostrar simulações computacionais a serem utilizadas para gerar uma base de dados que servirá para o treinamento de uma rede neural, voltada a implementar um algoritmo para priorização de pedestres em interseções semaforizadas.

O estudo aqui desenvolvido faz parte do projeto Utilização de Single Board Computer “Labrador” – para o controle inteligente da rede semaforica da cidade de São Paulo, pelo Laboratório de Sistemas Integráveis (LSI-TEC), da Fundação Laboratório de Sistemas Integráveis – USP, por meio de uma parceria entre o L-SITec, a CET/SP, SMT (Secretaria Municipal de Transportes – PMSP) e o BNDES (Banco Nacional de Desenvolvimento Econômico e Sustentável), dando suporte ao Plano Nacional de Internet das Coisas através do programa BNDES Pilotos em IoT – Projeto Piloto Mobilidade Urbana. A frente de instalação e programação dos aparelhos necessários já está em estágio avançado, e o presente artigo descreve parte dos esforços para atender à demanda de adequação da programação dos semáforos às leis de semaforização.

Quanto à sua estrutura, a seção 2 expõe os materiais e métodos utilizados, a seção 3 traz os resultados junto à discussão dos mesmos e a 4ª seção apresenta as conclusões alcançadas através do estudo.

Materiais e métodos.

Materiais (dados). A simulação usada neste artigo foi gerada com base em dados de tráfego real disponibilizados pela CET/SP. O modelo representa uma interseção semaforizada na Cidade de São Paulo, entre a Avenida Rebouças e a Rua Oscar Freire, onde há duas saídas de uma estação do metrô, um corredor central de ônibus na Av. Rebouças com veículos que trafegam pelos dois sentidos e um importante hospital nas proximidades – ver Figura 1. Aproximadamente 4 mil pessoas por hora atravessam essa interseção. Os dados utilizados são do ano de 2018 e, portanto, o modelo também reflete as condições da interseção àquela época.

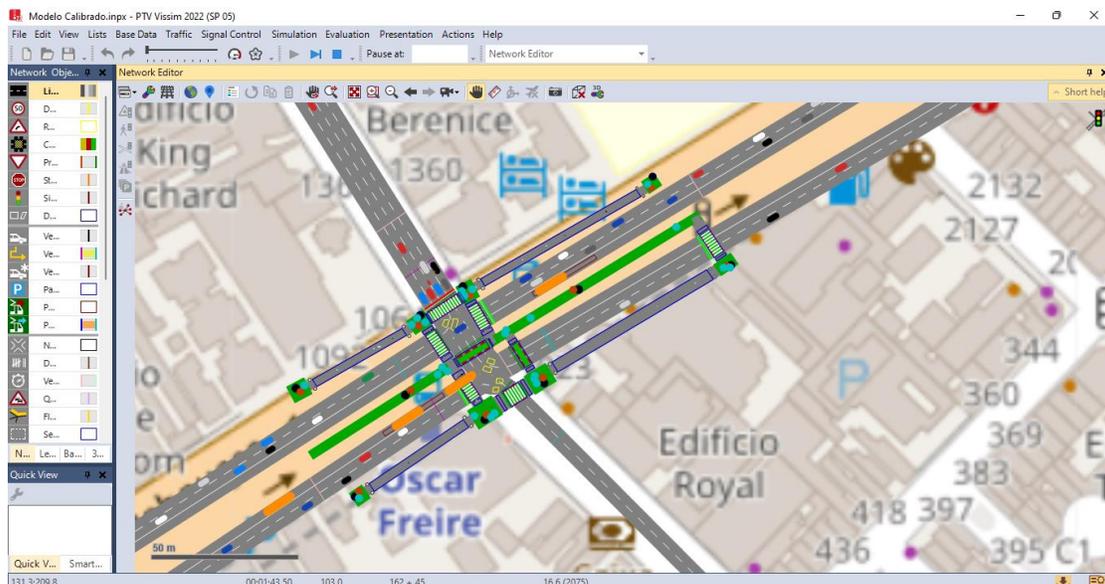


Figura 1. Modelo construído com base na interseção entre a Avenida Rebouças e a Rua Oscar Freire



Método – Simulação computacional.

Construção do micromodelo. O micromodelo de tráfego foi construído no software VISSIM pois se vislumbrava automatizar a criação de vários cenários, utilizando diferentes valores para alguns parâmetros, tais como programações semaforicas (tempos de ciclo) e fluxos de pedestres e de veículos. As rotas e paradas de ônibus, bem como semáforos e dispositivos de identificação de velocidade de veículos não foram alterados.

Para modelar o tráfego veicular da região, foram inseridos veículos de passeio, motocicletas e ônibus (em faixa exclusiva). As contagens veiculares foram realizadas das 7h às 9h, e, portanto, o período de simulação adotado foi do pico da manhã com duas horas, sendo quinze minutos de aquecimento. Para calibrar o modelo, foi feito o ajuste dos volumes e velocidades médias (km/h) de cada tipo de veículo. Arbitrou-se um erro percentual relativo máximo de 10% ao comparar os dados de saída do modelo (dados simulados) aos dados de radares disponibilizados pela CET (dados reais). Utilizou-se a estatística GEH para comparar os dados de volumes simulados versus os reais. Os dados, de saída do modelo, com os veículos calibrados e validados estão dispostos na Tabela 1.

Tabela 1: Dados de Saída do Modelo Calibrado

CALIBRAÇÃO						
Local	Volume (Carros)	Volume (motos)	Volume (ônibus)	Velocidade (carros)	Velocidade (motos)	Velocidade (ônibus)
Rebouças B/C	5062	469	0	39,37	40,71	0
Rebouças C/B	3726	0	0	31,33	0	0
Ônibus B/C	0	0	332	0	0	47,62
Ônibus C/B	0	0	257	0	0	47,63

Para calibrar os valores relativos a pedestres, o modelo considerou seis diferentes classes destes: Adultos (com diferenciação de sexo), Crianças (com diferenciação de sexo), Idosos e Cadeirantes. Idosos e Cadeirantes não foram diferenciados por sexo porque considerou-se que essa diferenciação seria pouco significativa para as duas classes, nas análises em questão. Além disso, essa não diferenciação permitiu economia de tempo na calibração de pedestres. No modelo há também uma quinta classe de pedestres chamada “Criança com Acompanhante”, que simula uma criança se locomovendo o tempo inteiro de mãos dadas com um adulto. Ela foi considerada adequadamente na calibração dos pedestres, mas como possuía baixo volume, em relação às demais classes de pedestres, esta não foi considerada nas demais análises.

As contagens de pedestres realizadas em campo pela CET apresentavam somente volume totais, sem diferenciar pedestres e sem analisar suas velocidades. Assim sendo, fez-se necessário definir as velocidades desejadas para cada tipo de pedestre e os volumes relativos de cada tipo na

simulação, registrados no VISSIM como desired speed distributions e os relative flows respectivamente.

Para o VISSIM, a desired speed para pedestres é a velocidade deles em um percurso sem obstáculos e interferências que os levem a desacelerar – como semáforos. Para os cadeirantes foi utilizada a desired speed padrão do VISSIM: 0,58 m/s. Para as outras classes de pedestres, as velocidades foram escolhidas com base em (JACOBSEN, 2011), supondo velocidades superiores para pessoas do sexo masculino um pouco mais altas do que para os outros (no estudo realizado, não foi considerada diferenciação de sexo entre idosos e cadeirantes). Os relative flows foram estimados com base na distribuição populacional no município de São Paulo. Segundo dados de 2010, a população do município se dividia em: 10,9% de meninos (até 14 anos), 10,6% de meninas (até 14 anos), 32,8% de homens (de 15 a 59 anos), 34,3% de mulheres (de 15 a 59 anos) e 11,7% de idosos (acima de 60 anos). Quanto à população com dificuldade severa ou impossibilidade de locomoção, um estudo da Prefeitura do Município de São Paulo de 2019 levantou que eram 1,94%, dentro do Censo Brasileiro de 2010 (PREFEITURA DO MUNICÍPIO DE SÃO PAULO, 2019) (IBGE, 2010).

No presente estudo admite-se que todos esses 1,94% são cadeirantes adultos, igualmente distribuídos entre homens e mulheres, e que 5% das mulheres estão acompanhando 5% das crianças (com igual distribuição entre meninos e meninas). Desta forma, obtiveram-se os valores apresentados na Tabela 2.

Tabela 2: Classes de pedestres junto aos seus dados de velocidade e fluxo relativo

TIPO DE PEDESTRE	VELOCIDADE	FLUXO RELATIVO
Homem	5,51 km/h (1,53 m/s)	32,2
Idoso	3,24 km/h (0,90 m/s)	11,7
Cadeirante	2,09 km/h (0,58 m/s)	1,3
Criança c/ responsável	3,60 km/h (1,00 m/s)	2,8
Mulher	4,93 km/h (1,37 m/s)	32,0
Menino	3,96 km/h (1,10 m/s)	10,4
Menina	3,60 km/h (1,00 m/s)	10,1

Para os pedestres, a calibração se baseia no Modelo de Helbing, envolvendo aspectos comportamentais, alterando os parâmetros de walking behavior (comportamento de caminhada) no VISSIM e no VISWALK.

Como o único dado de referência disponível para pedestres eram os volumes totais fornecidos pela CET, o processo de calibração dos pedestres se deu através da mudança iterativa de todos os parâmetros citados anteriormente, com o volume total sendo comparado à contagem da CET entre 7h e 9h.

Foram classificados oito tipos de movimentos nas calçadas e dezesseis nas faixas. Os movimentos são pares de “ida e volta”. Nas calçadas, todos os movimentos estão ao longo das margens da Avenida Rebouças, conforme a figura 2. O par formado por P1 e P2 e o par formado por P3 e P4 estão próximos às saídas da estação de Metrô Oscar Freire da Linha 4 – Amarela. Já os pares formados por P5 e P6 e por P7 e P8 estão aproximadamente alinhados ao ponto de ônibus presente no canteiro central. Estes movimentos estão explicitados na Figura 2.

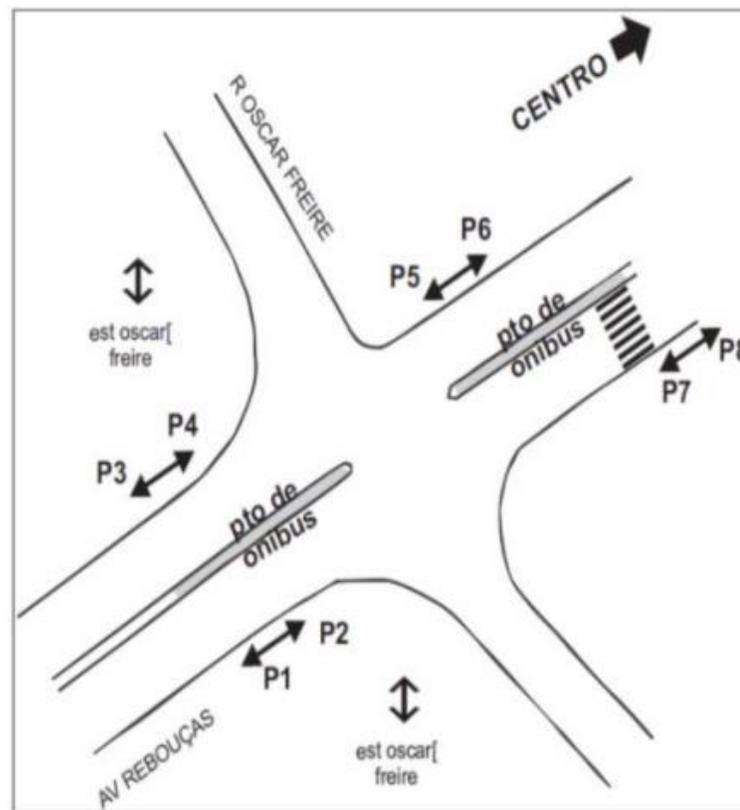


Figura 2. Movimentos de pedestres nas calçadas (CET, 2019).

Nas faixas, os movimentos P1 e P2 são os da faixa que se encontra no ponto em que o fluxo da Rua Oscar Freire adentra a Av. Rebouças. O par P3 e P4 é referente à faixa entre o lado do canteiro central que possui a parada de ônibus, e a calçada próxima ao ponto de entrada citado anteriormente. Este par também considera os movimentos ilegais de pedestres na diagonal, realizados com vista a alcançar a esquina no menor caminho possível. O par P5 e P6 representa os movimentos ilegais realizados por pedestres em um caminho que espelha aquele do par anterior em relação ao canteiro central, não existindo aqui faixa de pedestres. Também não se verificaram movimentos ilegais em diagonal neste lado. O par P7 e P8 é o da faixa ao longo do

ponto de saída do fluxo da Av. Rebouças para a Rua Oscar Freire. O par formado por P9 e P10 e aquele formado por P11 e P12 são aqueles que ligam o canteiro central às esquinas próximas às saídas da estação de Metrô Oscar Freire da Linha 4 – Amarela. O par P13 e P14 mostra os movimentos entre os dois pontos do canteiro central separados pelo cruzamento, no qual não existe faixa de pedestres. Por fim, os movimentos P15 e P16 são aqueles realizados na faixa entre o ponto de ônibus do canteiro central e a calçada no lado em que há a saída de fluxo da Av. Rebouças para a Rua Oscar Freire. Esta faixa se encontra distante da esquina. Os movimentos descritos estão explicitados na Figura 3.

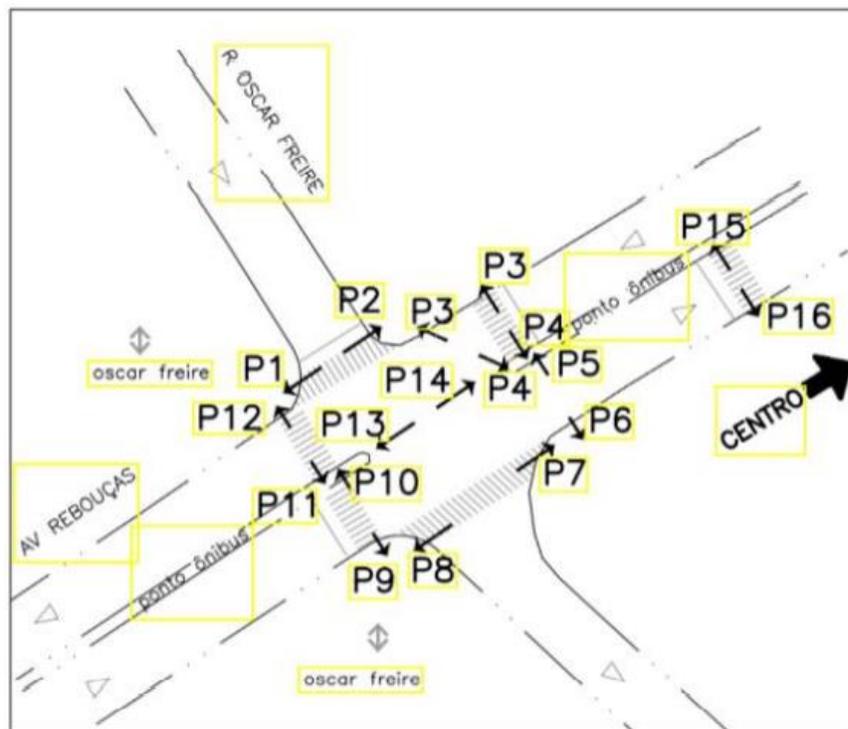


Figura 3. Movimentos de pedestres nos cruzamentos (CET, 2019).

Ao final do processo de calibração, os volumes de pedestres obtidos da simulação são os apresentados na Tabela 4, divididos por classe de pedestres para calçadas e para os cruzamentos. Nestas tabelas a diferenciação por gênero não foi considerada, pois os valores eram muito próximos.

Tabela 4: Volume de pedestres por classe

DISTRIBUIÇÃO DOS PEDESTRES SIMULADOS										
MOVIMENTO		ADULTO		IDOSO		CRIANÇA		CADEIRANTE		
C A L Ç A D A	P1	P1 + P2	215	264	40	52	87	97	4	4
	P2		49		12		10		0	
	P3	P3 + P4	956	1234	157	209	310	411	12	16
	P4		278		52		101		4	
	P5	P5 + P6	125	514	28	96	44	190	1	10
	P6		389		68		146		9	
	P7	P7 + P8	164	304	27	50	69	126	3	6
	P8		140		23		57		3	
C R U Z A M E N T O	P1	P1 + P2	180	594	37	123	53	199	4	9
	P2		414		86		146		5	
	P3	P3 + P4	87	232	20	36	26	86	3	5
	P4		145		16		60		2	
	P5	P5 + P6	47	77	2	9	14	31	2	3
	P6		30		7		17		1	
	P7	P7 + P8	191	245	41	49	52	88	5	6
	P8		54		8		36		1	
	P9	P9 + P10	517	748	98	140	193	280	3	5
	P10		231		42		87		2	
	P11	P11 + P12	231	541	48	116	81	191	6	12
	P12		310		68		110		6	
	P13	P13 + P14	24	58	4	8	11	18	1	2
	P14		34		4		7		1	
	P15	P15 + P16	19	47	4	11	3	16	0	1
	P16		28		7		13		1	

Plano de testes. O plano de testes inicial (um primeiro “pacote” de variações de demandas e tempos semafóricos a serem simulados) teve como objetivo modelar diferentes cenários, envolvendo o local de estudo por meio do software de microssimulação PTV VISSIM e da interface COM.

Esse plano de testes contou inicialmente com 600 cenários, com 20 ciclos semafóricos diferentes a serem testados e, para cada ciclo semafórico, 30 variações das demandas de veículos e pedestres. Essas variações foram geradas alterando os inputs de volumes: usando como referência os valores do cenário base, variaram-se aleatoriamente esses valores dentro de uma faixa de desvio máximo.

No plano de testes inicial, os inputs de pedestres foram variados em uma faixa de até 30% (para mais e para menos), e para os veículos a variação da faixa foi de 15% (também para mais e para menos).

O tempo total de ciclo da época dos dados disponibilizados pela CET (de 127 segundos) foi mantido, alterando-se apenas os tempos de verde e vermelho.

Sendo o estado inicial:



- Verde para Rebouças = 49 s;
- Vermelho inicial para Rebouças = 75 s;
- Amarelo (fixo) para Rebouças: 3 s.

Os tempos de verde e vermelho são incrementados e decrementados de 3s, de forma a manter o ciclo total de 127s. Este intervalo foi escolhido porque havia poucas diferenças com intervalos menores e ele ainda garante um número elevado de pontos a serem considerados. Com vistas a programar e automatizar o funcionamento do semáforo com o uso de linguagem Python.

Novamente, para fins práticos, só se alteraram os ciclos semaforicos que regem o cruzamento entre a Avenida Rebouças e a Rua Oscar Freire.

Para escrever o código, os ciclos foram alterados com base nos semáforos da Av. Rebouças. Isto é, no segundo plano de testes, os 20 cenários de ciclos semaforicos foram montados apenas para a Avenida Rebouças, sendo o ciclo da Oscar Freire calculado com base no da Rebouças, mantendo 127 segundos de ciclo e 3 segundos de amarelo em todos os casos. No cenário ID=0, por exemplo, as variáveis de verde, vermelho e amarelo representam o semáforo da Avenida Rebouças, enquanto os tempos da Oscar Freire são alterados e calculados pelo próprio código. O cenário de ID = 16, representa o ciclo semaforico do cenário base, com os tempos reais da região à época. Destaca-se que inicialmente foi dado foco majoritariamente a situações que dessem prioridade aos pedestres que pretendiam atravessar a Av. Rebouças, sendo estas situações também as mais críticas para os veículos da via.

Para o segundo plano de testes, as variações das demandas foram feitas de forma linear. Tomando-se como base os inputs de veículos e pedestres do cenário base calibrado, foram utilizadas as especificações abaixo:

- a) Atribuindo o valor de 100% ao fluxo de veículos no cenário base, os cenários futuros foram construídos alterando esse valor no range de 50% até 115%, com incremento (passo) de 5%, gerando um total de 14 pontos;
- b) Atribuindo o valor de 100% ao fluxo de pedestres no cenário base, os cenários futuros foram construídos alterando esse valor no range de 30% até 120%, com incremento (passo) de 5%, gerando um total de 19 pontos;
- c) Para as programações semaforicas, tomando como referência a Avenida Rebouças, que é a principal (ou troncal), se iniciou com períodos de 49 segundos de verde, 75 segundos de vermelho e 3 segundos de amarelo, o que totaliza 127 segundos (programação com o mesmo tempo semaforico que existe implementada atualmente pela CET, que prioriza o tráfego de veículos). Ao total, foram obtidas 20 programações semaforicas (cenários) diferentes.

Para a execução do código, têm-se dois conjuntos: o controlador e o executor. O controlador abre as variáveis iniciais, ciclos semafóricos e os inputs calibrados de veículos e pedestres e envia essas informações para o executor. O executor utiliza os dados iniciais de semáforos e varia, de forma linear entre os intervalos definidos, as quantidades de veículos e pedestres e em seguida recebe do VISSIM os valores dos indicadores de desempenho, fazendo o armazenamento temporário e posteriormente salvando-as em arquivos numa base de dados.

Para cada cenário futuro foram analisados os seguintes indicadores de desempenho (que são os valores de saída do VISSIM): tempo de viagem, tempo de espera e formação de filas. Esses indicadores foram obtidos tanto para veículos quanto para pedestres.

Para gerar os 5.320 cenários diferentes no VISSIM foi utilizada uma programação em linguagem Python, que utilizou a interface COM do VISSIM, de forma a obter para cada cenário os indicadores de desempenho acima citados.

O COM (Component Object Model) é um padrão que define a interface binária entre um aplicativo e um componente de software. No VISSIM, a interface COM permite implementar métodos a serem executados pelos objetos do programa, os quais são divididos em um esquema ramificado em árvore: IVissim contém INet e ISimulation, INet contém IAreas, INodes, IAreas e assim por diante. O esquema de hierarquia está representado na Figura 4.

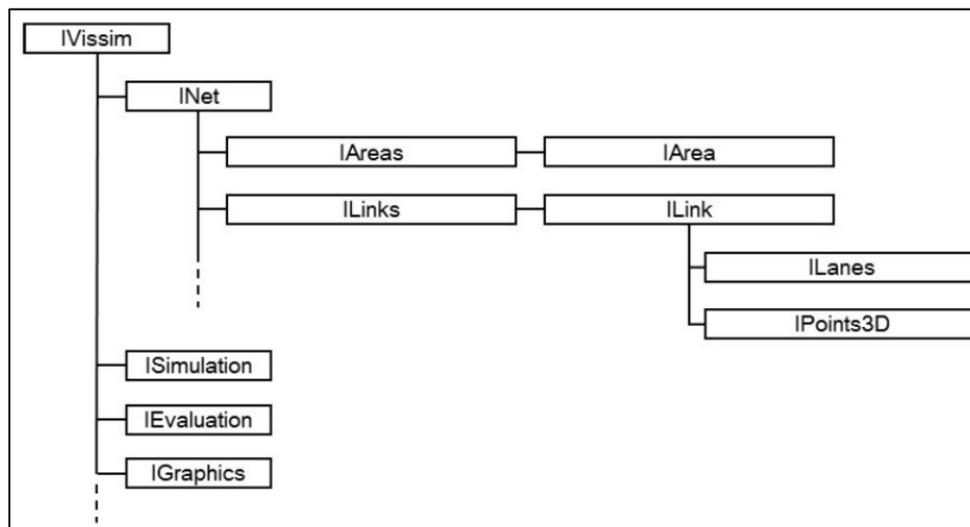


Figura 4. Esquema de hierarquia da interface COM para o VISSIM. (TETTAMANI e VARGA, 2018).

O programa funciona da seguinte forma: temos o micromodelo da interseção entre a Avenida Rebouças e a Rua Oscar Freire já calibrado e uma planilha que contém as variáveis iniciais – dados dos pedestres e dos veículos - que chamaremos de arquivos iniciais.

Além deles, há dois códigos preparados em Python, o *main*, que funciona como controlador, e o *executa*, que, como o nome indica, funciona como executor, sendo o responsável por utilizar a interface COM do VISSIM.

Assim, os arquivos iniciais são enviados para o *main* (controlador), que inicia o *executa*. O resultado final do processo é a geração de um banco de dados.

O funcionamento do programa está representado na Figura 5.

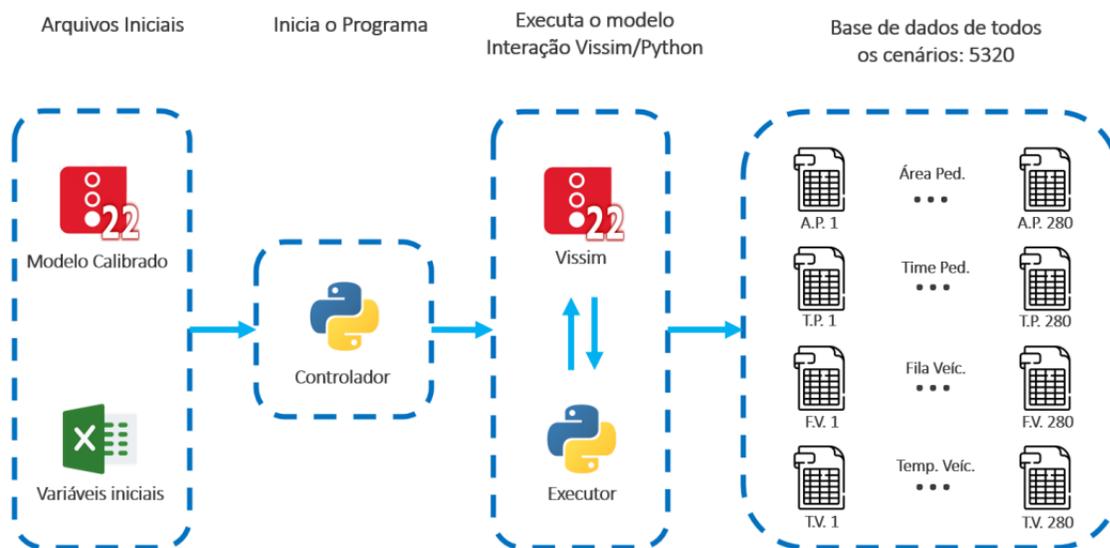


Figura 5. Fluxograma de funcionamento do programa.

Resultados e discussão. Após executadas as simulações, realizou-se a coleta dos resultados gerados por eles e armazenaram-se os dados na nuvem. Deu-se início então ao tratamento de dados. Para esta etapa foi criado um compilador que utiliza as bibliotecas Pandas e as do Python. Estas bibliotecas permitem trabalhar com *data frames* e interagir com o sistema operacional do computador.

São as principais funções do compilador:

- Criar *data frames* independentes para o armazenamento de cada tipo de resultado gerado;
- Iterar sobre os itens salvos para compilar todas as séries de dados do mesmo tipo em apenas uma estrutura (os *data frames* criados anteriormente);
- Salvar um arquivo .csv para cada *data frame*, de forma que os resultados do programa possam ser utilizados para analisar os resultados da simulação.

Em seguida, calculam-se dois índices, denominados alfa e beta. Alfa é uma variável utilizada como referência, tomada a partir do cenário base, armazenando um valor com o qual beta é comparado e aproximado. Cada simulação tem um beta calculado sobre seus resultados que é comparado ao alfa calculado.

O alfa é igual à razão entre o número de veículos e o número de pedestres do cenário base. Ele está representado na Equação 1:

$$alfa = \frac{\text{número de veículos}}{\text{número de pedestres}} \quad (1)$$

Também foram calculados valores de alfa(x), em que “x” é cada cenário alternativo. Ele está representado na Equação 2:

$$alfa(x) = \frac{\text{número de veículos em } x}{\text{número de pedestres em } x} \quad (2)$$

Por fim, calculou-se beta: o valor absoluto da diferença entre 1 e a razão entre alfa_x e alfa. Ele está representado na equação 3:

$$beta = abs((alfa_{(x)}) / alfa - 1) \quad (3)$$

O critério adotado foi o de manter a capacidade de operação da via próxima à do cenário-base, o que se alcança quando beta é minimizado.

O Gráfico 2 foi construído em uma planilha do Microsoft Office Excel a partir dos outputs dos códigos de compilação e de análise dos dados. Sua interpretação é de que as linhas são formadas pelos pontos que mais se aproximam do funcionamento original da via. Por exemplo: para 30% do valor base de pedestres e 50% do valor base de veículos o ciclo semafórico que preserva as condições de funcionamento da via é o ciclo 13.

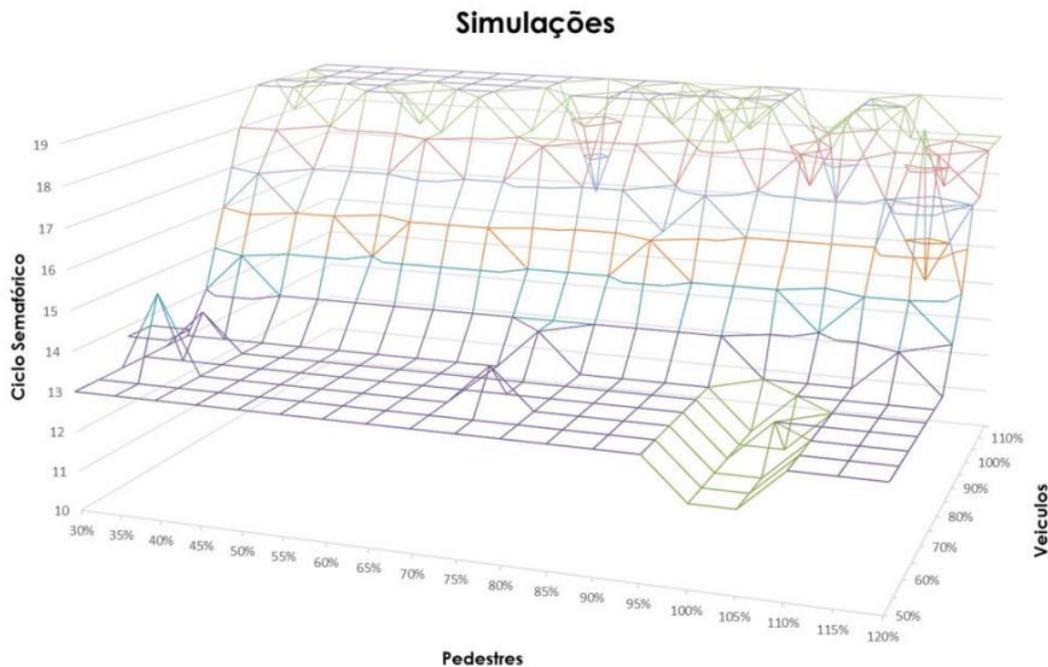


Gráfico 2. Análise dos resultados sem suavização.

Observa-se que as interações entre veículos e pedestres afetam sobremaneira a eficiência dos ciclos semafóricos utilizados, havendo picos e vales bastante expressivos. O ciclo 16, que é o do cenário base, no entanto parece ser adequado para ele. Curiosamente, ele se encaixa na área em que há o maior vale no gráfico, ou seja, tentar prever uma sinalização para ele através de extrapolações de outros cenários seria bastante ineficaz, reforçando a importância da modelagem.

Além disso, nota-se a confiabilidade dos programas criados para a automatização de processos no VISSIM.

O gráfico foi cortado do cenário 13 para cima porque ele é o único que assegura 90 segundos aos pedestres e ainda garante o funcionamento da interseção nas condições iniciais para os veículos. Cria-se abertura, portanto, para o estudo do impacto sobre pedestres.

Conclusões e trabalhos futuros. A compreensão do funcionamento do tráfego e das dificuldades enfrentadas pelo pedestre para realizar travessias seguras têm sido estudado em micromodelos e, através do uso da interface COM, alcançamos um grande número de cenários e análises em tempo reduzido.

Neste artigo diversas análises foram realizadas em tempo reduzido, permitindo encontrar ciclos semafóricos que permitem manter as condições de funcionamento da via para diversas demandas de motoristas e pedestres.



A partir daqui, é possível buscar outras soluções que incrementem a segurança nas vias tanto para pedestres como para veículos, como com o uso de inteligência artificial para controle semafórico.

Referências.

SMITH, R., REED, S., BAKER, S. (2010). Street design: Part 1. Complete streets. Public roads, 74(1), 12-17.

LECI RODRIGUES, ARMOND, et al. ACIDENTES DE TRÂNSITO POR ATROPELAMENTOS NA CIDADE DE SÃO PAULO: SÉRIE HISTÓRICA. Arquivos Catarinenses de Medicina. 2018.

COEUGNET, S.; CAHOUR, B.; KRAIEM, S. Risk-taking, emotions and socio-cognitive dynamics of pedestrian street-crossing decision-making in the city. Transportation Research Part F, n 65, págs. 141–157. 2019.

ZHANG, Y.; SU, R.; GAO, K.; ZHANG, Y. Traffic Light Scheduling for Pedestrians and Vehicles. 2017 IEEE Conference on Control Technology and Applications (CCTA). August 27-30, 2017. Kohala Coast, Hawai'i, USA.

TIAN, Y.; MA, X.; LUO, C.; ZHANG, Y. A Testbed for Intelligent Control of Traffic Lights at Pedestrian Crossings on a Road. 2018.

OBSERVATÓRIO MOBILIDADE SEGURA. Acidentes de Trânsito, 2021. Disponível em: http://mobilidadesegura.prefeitura.sp.gov.br/QvAJAXZfc/opendoc.htm?document=Painel_Mobilidade_Segura.qvw&host=QVS%40c65v27i&anonymous=true. Acesso em: 14 mar. 2022.

PTV AG. PTV Vissim 2022 User Manual. 2021.

DENATRAN. MANUAL BRASILEIRO DE SINALIZAÇÃO DE TRÂNSITO VOLUME V – SINALIZAÇÃO SEMAFÓRICA. 2014.

Introdução ao Win32 e C++. Módulo 2. Usando COM em seu programa Windows-Based. Disponível em <https://learn.microsoft.com/pt-br/windows/win32/learnwin32/module-2--using-com-in-your-windows-program>. Acesso em: 13 de fevereiro de 2023.

TETTAMANTI, T., VARGA, I. Development of road traffic control by using integrated VISSIM-MATLAB simulation environment. Periodica Polytechnica Civil Engineering, 56(1), pp. 43–49, 2012. <https://doi.org/10.3311/pp.ci.2012-1.05>.

JACOBSEN, A. C. MICROSSIMULAÇÃO DA TRAVESSIA DE PEDESTRES. Universidade Federal do Rio Grande do Sul. 2011.

Censo Brasileiro de 2010. IBGE. 2010.

PREFEITURA DO MUNICÍPIO DE SÃO PAULO. Dados atualizados do IBGE - Censo de 2010, Brasil e São Paulo (capital). 2019. Disponível em: . Acesso em: 20 abr. 2021.