

Controle de Semáforos Para Veículos de Emergência

Rodrigo Gonçalves de Branco

Universidade Federal de Mato Grosso do Sul
Faculdade de Computação
rodrigo.branco@ufms.br

Ronaldo Alves Ferreira

Universidade Federal de Mato Grosso do Sul
Faculdade de Computação
raf@facom.ufms.br

Edson Norberto Cáceres

Universidade Federal de Mato Grosso do Sul
Faculdade de Computação
edson@facom.ufms.br

RESUMO

O excesso de veículos em áreas metropolitanas gera sérios problemas para o deslocamento de socorristas no atendimento de emergências, atrasando os veículos de emergência nos momentos mais críticos. Uma das formas de reduzir os efeitos do congestionamento em um veículo de emergência (VE) é a utilização de preempção, i.e., mudança das fases dos semáforos ao longo de sua rota. Este trabalho propõe um algoritmo que utiliza o princípio de *shock-wave* para calcular o tempo para liberar uma fila de veículos que estão na frente de um VE esperando por um semáforo e determinar a temporização apropriada para disparar a preempção de um semáforo. Avaliações extensivas do algoritmo proposto simulando cenários reais de diferentes cidades do mundo e com diferentes níveis de congestionamento mostram que, para os cenários considerados, ele reduz em até 96,98% o tempo perdido pelo VE no trânsito, resultando em desempenhos superiores às soluções existentes na literatura.

PALAVRAS CHAVE. Controle de Semáforos, Sistema de Trânsito Inteligente, Preempção
Tópicos do artigo: Logística e Transportes, Simulação

ABSTRACT

The excess of vehicles in metropolitan areas creates problems for first responders during an emergency response, delaying emergency vehicles at critical moments. One of the ways to reduce the effects of congestion on an emergency vehicle (EV) is to use preemption, i.e., changing the traffic lights' phases along its route. This paper proposes an algorithm that uses the shock-wave principle to compute the time to dissipate a queue of vehicles ahead of the EV waiting at a traffic light and determine the appropriate timing to trigger the preemption of the traffic light. Extensive evaluations of the proposed algorithm simulating real scenarios of different cities and with varying levels of congestion show that, for the considered scenarios, it reduces up to 96.98% the loss time of the EV in traffic, resulting in better performances than the existing solutions.

KEYWORDS. Traffic Light Control, Intelligent Transportation System, Preemption
Paper topics: Logistics and Transportation, Simulation

1. Introdução

A mobilidade urbana tornou-se um problema sério nas grandes áreas metropolitanas. Em cidades como Los Angeles, Nova York e São Paulo, os engarrafamentos são frequentes e os motoristas gastam várias horas por dia apenas para se deslocarem entre casa e trabalho [Pishue, 2021]. O excesso de veículos nas vias gera sérios problemas para os socorristas, como bombeiros e paramédicos, atrasando os veículos de emergência em momentos críticos. Alguns estudos sugerem que a chance de sobrevivência de um paciente em parada cardíaca diminui entre 7% e 10% a cada minuto sem atendimento médico [Paniati e Amoni, 2006]. Para piorar a situação, os semáforos com programas de tempo fixo não priorizam os veículos de emergência, introduzindo atrasos extras, aumentando as chances de acidentes e colocando em risco a vida de pacientes e socorristas.

A maneira típica de se reduzir os efeitos do congestionamento em um veículo de emergência (VE) é usar *preempção* nos semáforos ao longo de sua rota. O mecanismo de *preempção* altera a política de semaforização em um cruzamento para conceder um intervalo verde (ou aumentar o intervalo verde) do semáforo na via em que o VE se aproxima e impor uma luz vermelha para as demais vias dessa interseção. As soluções propostas para esse problema de *preempção* usam diferentes métodos para detecção do VE (por exemplo, comunicação RF, RFID, VANET, V2V e V2I) [Hajiebrahimi e Iranmanesh, 2018; Almuraykhi e Akhlaq, 2019], algoritmos que consideram uma interseção ou um pequeno conjunto de interseções [Mu et al., 2018], e algoritmos para calcular o tempo que o VE leva para chegar à interseção [Qin e Khan, 2012; Djahel et al., 2015; Kapusta et al., 2017]. No entanto, essas propostas não apresentam melhorias significativas e consistentes no tempo de viagem dos VEs. Outras exigem infraestrutura cara e difícil de implantar (ou seja, *loops* de indução ou *Road Side Units* - RSU). Cabe salientar que o tempo de *preempção* médio necessário para bons resultados dessas estratégias pode ter um impacto muito grande nos tempos de viagem dos demais veículos.

O trabalho proposto em [Kapusta et al., 2017] é baseado no esvaziamento das filas de veículos que estão na via em que o VE está trafegando para que o VE perca o menor tempo possível em virtude de desacelerações. O objetivo do algoritmo é acionar as *preempções* somente por tempo suficiente para que o VE possa trafegar sem desacelerar, minimizando os efeitos dessa mudança na semaforização para os veículos que estão nas vias concorrentes desse cruzamento e que aguardam o fim da *preempção* para poder seguir em seus trajetos. Contudo, dependendo das características do semáforo e das condições do trânsito, esse algoritmo pode demorar muito, fazendo com que o VE perca tempo ou que a *preempção* ocorra muito cedo, atrapalhando assim os outros veículos desnecessariamente.

Este trabalho propõe um algoritmo que melhora vários aspectos do algoritmo [Kapusta et al., 2017] e que resulta em reduções significativas no tempo de viagem de um VE. Em particular, o tempo necessário para esvaziar a fila de veículos à frente do VE é um dos fatores que mais impactam em quanto tempo e com quanta antecedência a *preempção* deve iniciar. O algoritmo proposto utiliza o princípio de ondas de choque de movimentação de veículos em uma fila [Marshall e Berg, 1997; Kamalanathsharma e Hancock, 2010] para determinar o momento que a *preempção* deve iniciar de modo a evitar que o VE desacelere ou pare completamente em uma interseção.

O algoritmo proposto foi avaliado usando extensivas simulações com cenários reais baseados em dados de circulação de veículos das cidades de Turim (Itália) e Colônia (Alemanha), analisando assim diferentes níveis de congestionamento. Além disso, nos cenários estudados, mostramos que os resultados da solução proposta são superiores a outras estratégias descritas na literatura, reduzindo em até 96,98% o tempo que o VE perde em sua rota em função de semáforos quando comparado com uma viagem de emergência sem *preempção*. O código-fonte de nossa im-

plementação em Python, os mapas para os cenários simulados e os resultados da simulação estão disponíveis publicamente em [de Branco et al., 2022].

O restante deste trabalho está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados e contextualiza o problema, analisando e identificando possíveis aprimoramentos no Algoritmo [Kapusta et al., 2017]. A Seção 3 descreve o algoritmo proposto, destacando as principais contribuições deste trabalho. A Seção 4 descreve brevemente a implementação da solução proposta. Na Seção 5, são apresentados os resultados de simulação comparando a solução proposta e outras abordagens existentes na literatura. A Seção 6 discute possíveis melhorias e extensões na solução proposta e a Seção 7 conclui este trabalho.

2. Contextualização e Trabalhos Relacionados

Idealmente, um VE atendendo uma emergência deveria trafegar na velocidade máxima das vias, com fluxo livre e com todos os semáforos em sua rota abertos para que ele não precisasse desacelerar. Nesse caso, o VE não perderia tempo devido ao tráfego e chegaria ao seu destino no menor tempo possível. Entretanto, esse cenário não ocorre na prática e vários trabalhos na literatura científica buscam minimizar o tempo perdido por um VE ao longo de um trajeto durante uma emergência. De modo geral, a maioria dos trabalhos usa dados em tempo real do VE e das condições de tráfego para atuar na programação dos semáforos existentes na rota do VE.

Alguns trabalhos usam sensores de aproximação para disparar a preempção de semáforos [Miyawaki et al., 1999]. Soluções que dependam fortemente do posicionamento dos sensores podem não funcionar bem quando o VE está longe do sensor e há congestionamento a sua frente em função de um sinal vermelho. Além disso, devido à área de cobertura, o sensor pode detectar um VE em uma via próxima e alterar a programação de um semáforo que o VE não cruzará [Collura et al., 2001].

Outras propostas descritas em [Khan et al., 2018; Moroi e Takami, 2015] usam cenários sintéticos pouco representativos e não validam seus resultados usando mapas reais, não capturando, portanto, o comportamento caótico do tráfego de uma cidade. Em [Moroi e Takami, 2015], por exemplo, o experimento é executado em um grid simples, enquanto em [Kapusta et al., 2017] o cenário avaliado possui apenas uma interseção. Outro trabalho [Djahel et al., 2015] mostra resultados para VEs usando diferentes rotas, de modo que o tempo médio pode não produzir a melhoria real da perda de tempo.

A solução determinística conhecida que apresenta os melhores resultados para se diminuir o tempo de viagem de um VE via preempção de semáforos foi proposta em [Kapusta et al., 2017] e sua abordagem está descrita no Algoritmo 2.1. O algoritmo calcula o tempo que o VE levará para chegar a um semáforo (Linha 2) e modifica sua programação dependendo do tamanho da fila existente diante daquele semáforo e à frente do VE. Note que o algoritmo assume que a posição do VE e a quantidade de veículos enfileirados no semáforo são conhecidas. Essas informações podem ser facilmente obtidas atualmente via *middlewares* de cidades inteligentes [Batista et al., 2016]. Na Linha 3, o algoritmo verifica se o tempo para o VE chegar ao semáforo está entre o tempo da duração do ciclo do semáforo multiplicado pelos fatores T_β e T_α (e.g., $T_\beta = 0.5$ e $T_\alpha = 3$ [Kapusta et al., 2017]), o que significa que o algoritmo pode ter de atuar na programação do semáforo em função do número de veículos enfileirados no semáforo. Dependendo do tamanho da fila de veículos, o algoritmo pode tanto aumentar o tempo de ciclo verde (fila maior que s_α , Linhas 4 e 5), quanto diminuir o tempo de ciclo vermelho (fila maior que s_β , Linhas 7 e 8). Em [Kapusta et al., 2017], os valores de s_α e s_β foram fixados em 10 e 15 metros, respectivamente. Note que o algoritmo inicia a preempção a partir do momento que o tempo de chegada do VE ao semáforo for inferior a T_α vezes a duração do ciclo e continuamente altera a duração das fases do semáforo enquanto o

tempo de chegada do VE ao semáforo for superior a T_β vezes a duração do ciclo (Linha 3). Quando o veículo estiver próximo do semáforo (*i.e.*, o tempo de chegada for inferior à duração do ciclo multiplicado pelo fator T_β , Linha 11), o algoritmo força a transição para a fase verde (Linha 16) ou então a mantém se o semáforo já estiver nessa condição (Linha 13).

Algoritmo 2.1: Algoritmo proposto em [Kapusta et al., 2017]

```
1 while VE está em Emergência do
2   TempoDeChegada  $\leftarrow \frac{d}{v_{VE}}$ ;
3   if  $T_\beta \cdot DuracaoDoCiclo < TempoDeChegada < T_\alpha \cdot DuracaoDoCiclo$  then
4     if TamanhoDaFila  $> s_\alpha$  then
5       Aumentar a duração do ciclo verde;
6     end
7     if TamanhoDaFila  $> s_\beta$  then
8       Diminuir a duração do ciclo vermelho;
9     end
10  end
11  if  $TempoDeChegada < T_\beta \cdot DuracaoDoCiclo$  then
12    if faseAtual == faseVerde then
13      Mantenha a Fase Verde;
14    end
15    if faseAtual == faseVermelha then
16      Altere para a Fase Verde;
17    end
18  end
19 end
20 Retorne para o programa original do semáforo;
```

Esse algoritmo proposto em [Kapusta et al., 2017] não considera três aspectos importantes. Primeiro, se a distância do VE até o semáforo for grande, a preempção não é iniciada, ou seja, as condições das Linhas 3 e 11 seriam falsas. Porém, dependendo do nível de congestionamento, a preempção deve ser iniciada o quanto antes para dissipar a fila de veículos à frente do VE e evitar que ele tenha de desacelerar. Segundo, o algoritmo ativa a preempção mesmo quando há pouco congestionamento na via, pois considera apenas a duração do ciclo do semáforo e os parâmetros T_α , T_β , s_α e s_β . Por exemplo, para uma duração de ciclo de 90 segundos e $T_\alpha = 3$, um VE trafegando a $70km/h$ irá ativar a preempção quando estiver a uma distância de $5,4km$ para os parâmetros $s_\alpha = 10m$ e $s_\beta = 15m$ utilizados em [Kapusta et al., 2017], o que corresponderia a ter apenas de dois a quatro veículos na fila do semáforo. E por último, em experimentos que realizamos, observamos que o algoritmo mantém a preempção ativa por longos períodos quando há congestionamento, pois ele simplesmente altera as durações das fases verde e vermelha (Linhas 5 e 8), quando na realidade ele deveria manter o semáforo verde para os veículos da via do VE e dissipar rapidamente a fila de veículos parados no semáforo. Cabe ressaltar que, como os autores não mencionam os fatores de aumento e redução dos ciclos (Linhas 5 e 8) e que não obtivemos as informações nem o código fonte para reprodução exata de seus resultados, utilizamos uma redução de 10% (definida empiricamente) para os dois casos.

3. Solução Proposta

Nesta seção, apresentamos um novo algoritmo que soluciona os problemas identificados no algoritmo da seção anterior e que reduz o tempo de viagem de um VE quando comparado a várias outras abordagens, conforme mostraremos na Seção 5. O algoritmo proposto considera o nível de congestionamento da via do VE para definir o tempo de preempção. Se o congestionamento for alto, a preempção ficará ativa por um período mais longo, ao passo que se o congestionamento for baixo ou inexistente, a preempção só será ativada para não desacelerar o VE, auxiliando assim o fluxo de veículos em vias concorrentes. Além disso, a preempção é ativada uma única vez e permanece ativa até que o VE cruze o semáforo.

O Algoritmo 3.1 mostra os passos de nossa solução. A ideia principal do algoritmo é utilizar o princípio de ondas de choque [Stephanopoulos et al., 1979; Marshall e Berg, 1997] para se determinar o tempo de dissipação de uma fila de veículos diante de um semáforo. Em [Stephanopoulos et al., 1979], o princípio de ondas de choque foi utilizado para modelar e analisar o comportamento de filas causadas por semáforos em interseções. A ideia básica é que as luzes do semáforo são vistas como ondas de choque, levando a condições de congestionamento próximo ao semáforo durante o período de luz vermelha e de liberação da fila durante o período de luz verde.

O tempo para o VE alcançar o semáforo é calculado na Linha 2 para se determinar quando a preempção deve iniciar. Porém, o algoritmo também considera o tempo necessário para esvaziar a fila de veículos a frente do VE (Q_{flush} , Linha 3) e o tempo para a alteração da fase atual do semáforo (t_{flush}), pois pode ser necessário fazer a transição para as luzes amarela e vermelha nas vias concorrentes à via do VE.

Para se determinar Q_{flush} , considere Q como o comprimento da fila de veículos na frente do semáforo em metros. Segundo o princípio de ondas de choque [Stephanopoulos et al., 1979; Marshall e Berg, 1997], o último veículo da fila Q começará a se mover após $\frac{Q \cdot k}{r}$ segundos, em que k é a densidade do tráfego em veículos por metro e r é a taxa do fluxo de saturação em veículos por segundo. Porém, é necessário também adicionar a Q_{flush} o tempo que esse último veículo levará para cruzar a interseção. Isso vai depender do tamanho da fila e da velocidade (v) e aceleração (a) desse veículo. Neste trabalho, consideramos que v é a velocidade máxima da via e $a = 2,6m/s^2$, conforme determinado em estudos anteriores [Lopez et al., 2018]. Se a fila for curta, o último veículo ainda estará acelerando quando cruzar a interseção (se a condição da Linha 4 for verdadeira) e levará $\sqrt{\frac{2Q}{a}}$ segundos. Caso contrário, se a fila for longa o suficiente para que o veículo atinja sua velocidade máxima e depois termine o restante do trajeto com velocidade constante, o veículo gastará $\frac{v}{a} + \frac{Q - \frac{v^2}{2a}}{v}$ segundos para cumprir esse percurso. Se a soma dos tempos de esvaziar a fila de veículos (Q_{flush}) e alterar o semáforo para o sinal verde (t_{flush}) for maior ou igual ao tempo de chegada do VE até o cruzamento (Linha 10), a preempção deve ser iniciada imediatamente (Linha 11). O algoritmo mantém a preempção até que o VE cruze a interseção e depois restaura a programação normal do semáforo (Linha 14).

Se o tempo que o VE leva para alcançar o semáforo for inferior ao tempo necessário para esvaziar a fila de veículos a sua frente, o VE alcançará a fila ou o semáforo enquanto a fila ainda estiver sendo esvaziada, forçando-o a desacelerar. Para evitar essa situação, o Algoritmo 3.1 calcula periodicamente o tempo de chegada do VE e o tempo para esvaziar a fila de semáforos, iniciando a preempção se for necessário (Linha 10). Se a preempção ainda não foi iniciada, no próximo passo o VE estará mais próximo do semáforo e a fila de veículos pode estar maior ou menor. Nesse momento, uma nova verificação é feita, recalculando o tempo de chegada do VE e o tamanho da fila. Se for o momento adequado (condição verdadeira para a Linha 10), a preempção é iniciada. Como o algoritmo é executado constantemente e periodicamente (a cada um segundo

Algoritmo 3.1: Algoritmo Proposto

```

1 while VE está em Emergência do
2   TempoDeChegada  $\leftarrow \frac{d}{v_{VE}}$ ;
3    $Q_{flush} \leftarrow \frac{Q^k}{r}$ ;
4   if  $Q \leq \frac{v^2}{2a}$  then
5      $Q_{flush} \leftarrow Q_{flush} + \sqrt{\frac{2Q}{a}}$ ;
6   end
7   else
8      $Q_{flush} \leftarrow Q_{flush} + \frac{v}{a} + \frac{Q - \frac{v^2}{2a}}{v}$ ;
9   end
10  if TempoDeChegada - ( $Q_{flush} + t_{flush}$ )  $\leq 0$  then
11    Altere para a Fase Verde;
12  end
13 end
14 Retorne para o programa original do semáforo;

```

na nossa implementação) durante a emergência, eventualmente, a preempção será iniciada e será mantida assim até o VE cruzar a interseção. Nesse momento, a preempção é finalizada e o estado do semáforo é restaurado. Note que o algoritmo proposto ativa a preempção uma única vez, evitando assim a alternância de duração de fases do Algoritmo 2.1 e interferindo o mínimo possível no tráfego no entorno do semáforo.

4. Implementação

Considerando as características do problema de controle de semáforos, em que ações individuais dos veículos podem depender de outros elementos no cenário simulado, optamos por utilizar o simulador SUMO [Lopez et al., 2018], que é um simulador que captura a complexidade do trânsito de uma cidade simulando eventos microscópicos. O simulador SUMO oferece uma API (*Application Programming Interface*) para Controle de Tráfego (TraCI) que permite conectar módulos para obter os dados da simulação, como localização e velocidade de veículos, estado do semáforo e limites de velocidade de quarteirões. A API também permite a alteração do estado dos semáforos. O algoritmo proposto foi implementado em Python e integrado a um framework genérico que construímos para interagir com o SUMO e que também pode ser utilizado para implementar outros algoritmos de controle. A implementação é genérica o suficiente para interagir não só com o SUMO, mas também com um middleware de cidade inteligente, algo que pretendemos fazer em um trabalho futuro e integrá-lo ao middleware desenvolvido em [Batista et al., 2016]. O código fonte e os dados para reprodução dos resultados estão disponíveis em [de Branco et al., 2022].

5. Avaliação de Desempenho

Para avaliar e comparar o desempenho do algoritmo proposto na Seção 3, implementamos o algoritmo [Kapusta et al., 2017] e mais outros três algoritmos existentes na literatura. O primeiro usa etiquetas RFID e representa uma grande classe de algoritmos baseados em sensores. Ele detecta quando o VE está a uma distância de detecção dd em metros de um semáforo usando tags RFID, para então iniciar a preempção por nc ciclos (ou seja, tempo fixo). Tanto dd quanto nc são parâmetros fixos do algoritmo, ou seja, eles não mudam durante o tempo de viagem do VE. Experimentos preliminares mostraram que este algoritmo apresenta melhores resultados quando $dd = 100$ e $nc =$

5. Portanto, esses valores foram utilizados neste trabalho. O segundo algoritmo usa informações globais e lógica Fuzzy para inferir os níveis de ocupação e congestionamento das vias [Djahel et al., 2015]. Ele usa a ocupação e os diferentes níveis de emergência para o VE (baixo, médio e alto) para determinar se a preempção deve ser iniciada. Neste trabalho, definimos o parâmetro *el* (*Emergency Level*) igual a *high*, pois ele produz os melhores resultados para este algoritmo. O último algoritmo implementado é denominado “Tudo Verde”, um algoritmo teórico que dispara a preempção de todos os semáforos do percurso do VE ao mesmo tempo e os mantém acionados até o VE cruzar o último semáforo da rota. Apesar de ineficaz em casos reais, esta implementação serve como limite superior de melhoria possível para os outros algoritmos. Como os autores de [Kapusta et al., 2017] e [Djahel et al., 2015] não divulgaram os códigos-fontes de seus algoritmos, este trabalho os implementa usando apenas as descrições existentes nos respectivos artigos.

A primeira avaliação utiliza um cenário similar ao apresentado em [Kapusta et al., 2017]. Para isso, foi criado o *Cenário Uma Interseção*, composto de uma rede contendo uma interseção com um semáforo ao centro e oito vias em formato de cruz (uma via de origem e uma via de destino em cada uma das posições *Norte*, *Sul*, *Leste* e *Oeste*), com três quilômetros cada. Denominamos uma *rede* o conjunto de todas as vias de um dado cenário. Cada via de origem está ligada a três outras vias de destino (pois conversão em forma de U, ou *U-turn*, não é permitida). Portanto, para todos os pares de origem-destino possíveis (12 no total), um fluxo de veículos é definido. Os veículos de cada fluxo são gerados com probabilidade de 15% a cada segundo. Esta configuração é suficiente para saturar a rede no momento da entrada do VE, mas sem criar um backlog de inserção de veículos (quando o simulador não consegue inserir os veículos aptos na rede por falta de espaço físico disponível na via). Dessa forma, no momento da partida do VE, ele é instantaneamente inserido na rede. Além do VE, são inseridos, em média, aproximadamente 3532 veículos em cada execução do algoritmo. Nesse cenário, o VE percorre seis quilômetros entre a origem e o destino.

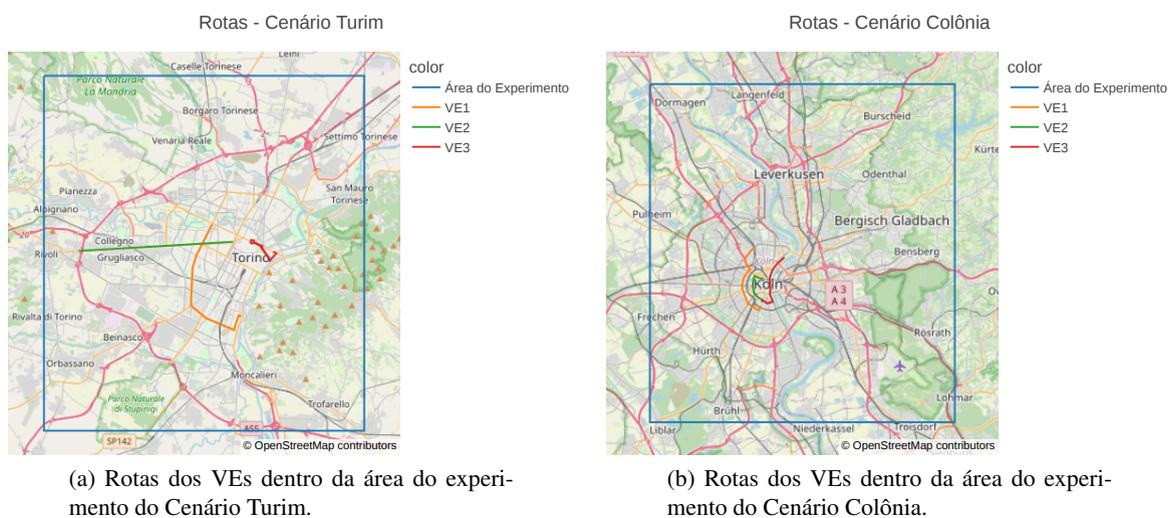


Figura 1: Rotas dos VEs dos Cenários de Turim e Colônia.

Considerando que um cenário com uma única interseção não é realista, usamos mais dois cenários em que todas as interseções da rota do VE são consideradas e são mais próximos da realidade de uma grande cidade. O primeiro foi adaptado de [Rapelli et al., 2021] e corresponde a uma área de 600km^2 da região da cidade de Turim (Itália), com dados reais do trânsito para um período

Tabela 1: Distância percorrida por cada VE e o número de semáforos que cada um cruza.

(a) Parâmetros das rotas dos VEs - cenário Turim.			(b) Parâmetros das rotas dos VEs - cenário Colônia.		
VE	Distância	Semáforos	VE	Distância	Semáforos
VE1	9.961,68m	35	VE1	8.351,42m	26
VE2	9.952,10m	19	VE2	5.044,34m	27
VE3	5.300,60m	33	VE3	6.344,31m	24

de 24 horas e com um total de 2.202.814 veículos. Nesse cenário, três VEs com suas respectivas rotas foram definidos. A utilização de três VEs possibilita a avaliação dos algoritmos em três rotas distintas. A Figura 1a apresenta a área do experimento e as rotas dos VEs. Foram inseridos 175.873 veículos durante o período avaliado das 6h às 8h e o VE começou a atender uma emergência às 7h.

O segundo cenário é baseado em [Varschen e Wagner, 2006] e corresponde a um intervalo de tráfego de 24 horas da cidade de Colônia (Alemanha) e um total de 1.549.612 veículos. As considerações para este cenário são as mesmas que os do cenário de Turin, inclusive para a janela de tempo de entrada do VE, que neste cenário resultou na inserção de 252.754 veículos. O mapa do cenário e as rotas dos VEs são mostrados na Figura 1b.

Para essas duas cidades também foram geradas viagens sintéticas para comparar com os dados reais de tráfego. Nesse caso, duas arestas são escolhidas aleatoriamente com distribuição uniforme para a origem e o destino de cada viagem, resultando em 180.001 viagens em Turin e 240.002 viagens em Colônia. As distâncias percorridas e as quantidades de semáforos cruzados por cada VE para cada cenário são apresentadas nas Tabelas 1a e 1b.

Para cada execução da simulação, o VE é sempre o mesmo e segue as rotas definidas nas Figuras 1a e 1b, enquanto os outros veículos são roteados dinamicamente. Caso algum dos outros veículos fique parado por mais de 300 segundos (nos casos de *deadlock*), o simulador o remove da rede, gerando um evento denominado *teletransporte*. O VE nunca é removido da simulação, mesmo que fique parado por mais de 300 segundos.

Os algoritmos são comparados utilizando o tempo perdido pelo VE. Considere que o *Melhor Tempo de Viagem (btt)* seja definido como o tempo de viagem do VE quando ele viaja à velocidade máxima em sua rota e não para em nenhum semáforo ou interseção. Considere também o *Tempo de Viagem Atual (att)* como o tempo de viagem efetivamente aferido para o VE. Assim, o *Tempo Perdido* é definido como $tl = att - btt$.

Uma versão sem preempção é usada como base de comparação e a melhoria percentual é calculada por $(1 - \frac{tl \text{ com preempcao}}{tl \text{ sem preempcao}}) \times 100$. Considerando que não é conhecida a distribuição dos dados, o método de Thompson [Thompson, 1936] foi utilizado, e os resultados possuem um nível de confiança de 90% para os dados entre os percentis 15° e 85°, para 25 execuções independentes (excluindo o pior resultado em cada caso). Os experimentos foram executados em um servidor com dois processadores Xeon E5-2630, 396GB de memória RAM rodando o sistema operacional Linux/Debian 8.11, o Simulador Sumo 1.8 e o interpretador Python 3.5.3.

Para o *Cenário Uma Interseção*, o algoritmo proposto empata com o *Algoritmo Tudo Verde* (possuindo ambos mediana de 74, 5%, Figura 2a) e é superior a todos os outros. Neste cenário, quando o VE ingressa na simulação, a rede já está saturada e, portanto, o algoritmo proposto e o *Algoritmo Tudo Verde* iniciam a preempção imediatamente. Com a rede saturada, os *Algoritmos RFID* e [Kapusta et al., 2017] não detectam o VE por um bom tempo, o que diminui a melho-

ria do tempo perdido, enquanto o *Algoritmo [Djahel et al., 2015]* causou atrasos maiores quando comparado à versão Sem Preempção. Isso pode ser observado na Figura 2a.

Do ponto de vista de tempo de preempção médio, era esperado que os algoritmos que tiveram as maiores reduções do tempo de viagem do VE tivessem os valores mais altos para o tempo médio de preempção, pois eram acionados antes e permaneciam assim até que o VE cruzasse o semáforo. O tempo de preempção médio é equivalente para o Algoritmo proposto e o *Tudo Verde*, já que a preempção inicia e termina no mesmo momento, quando o VE cruzou o semáforo. Contudo, o *Algoritmo [Kapusta et al., 2017]* ficou ativo por mais tempo do que o Algoritmo proposto e o *Algoritmo Tudo Verde*, conforme apresentado pela Figura 2b. Isso é um indicativo da alternância das fases do semáforo na primeira parte do Algoritmo [Kapusta et al., 2017] (Linhas 3 a 10).

Por fim, é interessante notar que o *Algoritmo Tudo Verde* não obteve melhoria de 100% no tempo perdido pelo VE, demonstrando que há outros fatores que impactam essa métrica além dos atrasos provocados exclusivamente pelo semáforo.

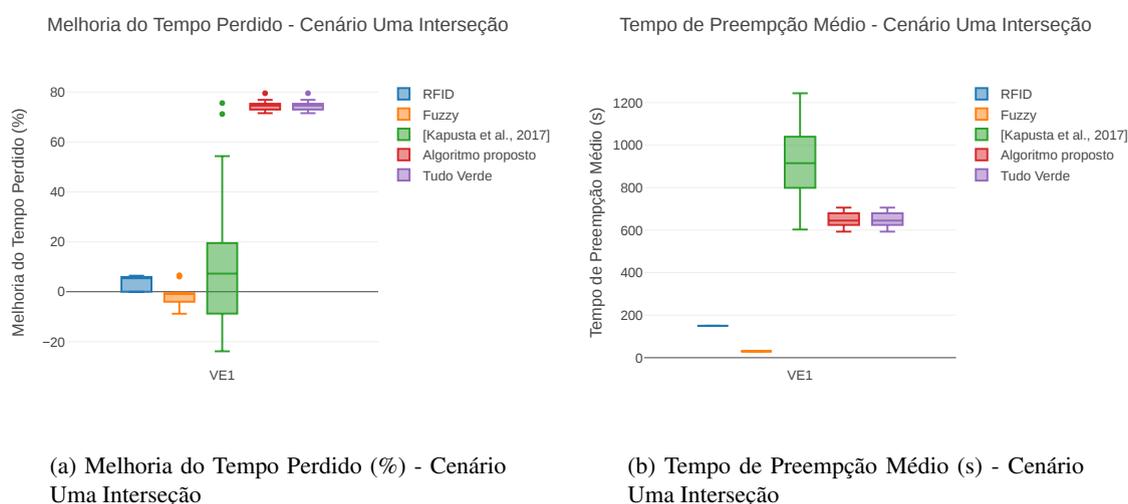


Figura 2: Resultados para o Cenário Uma Interseção.

Na avaliação utilizando o cenário com as demandas reais de Turim (Figura 3), com exceção de VE1 e VE1-Sintético, o Algoritmo proposto é consideravelmente superior aos demais algoritmos, inclusive obtendo melhores resultados que o *Algoritmo Tudo Verde* em alguns casos, com melhorias chegando até a 88,33% em sua mediana (Figura 3a). Esse desempenho superior ao *Algoritmo Tudo Verde* também explica o motivo do VE1 obter resultados não satisfatórios: é possível que a preempção esteja ativa, mas complicações no trânsito (congestionamento, acidentes etc.) podem impedir o bom andamento dos veículos. Neste caso, em todos os algoritmos, manter a preempção pode piorar a situação. Para o tempo de preempção médio (Figura 3b), o Algoritmo proposto apresenta valores mais baixos quando comparados ao Algoritmo [Kapusta et al., 2017]. E como esperado, como os semáforos ficam todos ativos e permanecem assim até o final, para o *Algoritmo Tudo Verde* o *Tempo de Preempção Médio* se mostra proibitivo em alguns casos¹.

¹Para navegar pelos gráficos deste trabalho e verificar outros valores que não as medianas, acesse https://nbviewer.org/github/rodrigogbranco/charts/blob/main/kapustaimp-paper.ipynb?flush_cache=true passando o mouse por cima dos *boxplots*

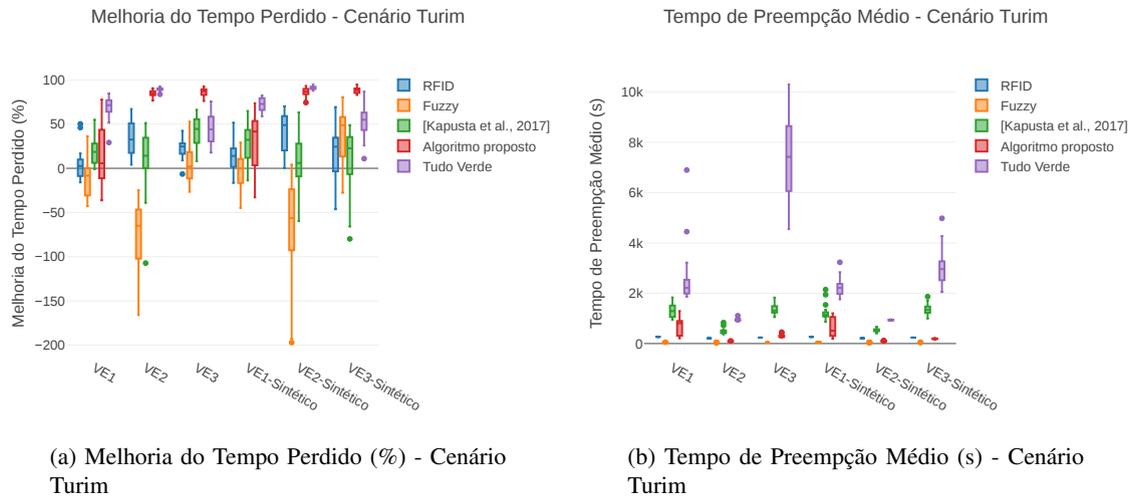


Figura 3: Resultados para o Cenário Turim.

Por fim, para o Cenário de Colônia (Figura 4), o Algoritmo Proposto também obteve bons resultados em comparação aos outros algoritmos, com melhorias atingindo 96, 98% para a sua mediana (Figura 4a). A exceção nesse cenário é para o VE3, cuja mediana é de 45, 88% (o segundo maior valor), mas inferior ao Algoritmo RFID, cuja mediana apresenta melhoria de 58, 17%. Contudo, neste cenário os Algoritmos 2.1 e Tudo Verde obtiveram resultados negativos. Nem sempre portanto é útil manter a preempção ativa, sendo imperativo que os Algoritmos detectem e evitem tais situações. Do ponto de vista do Tempo de Preempção Médio (Figura 4b), o Algoritmo Proposto apresenta baixos valores, reforçando o custo-benefício em sua utilização, quando comparado com os Algoritmos concorrentes.

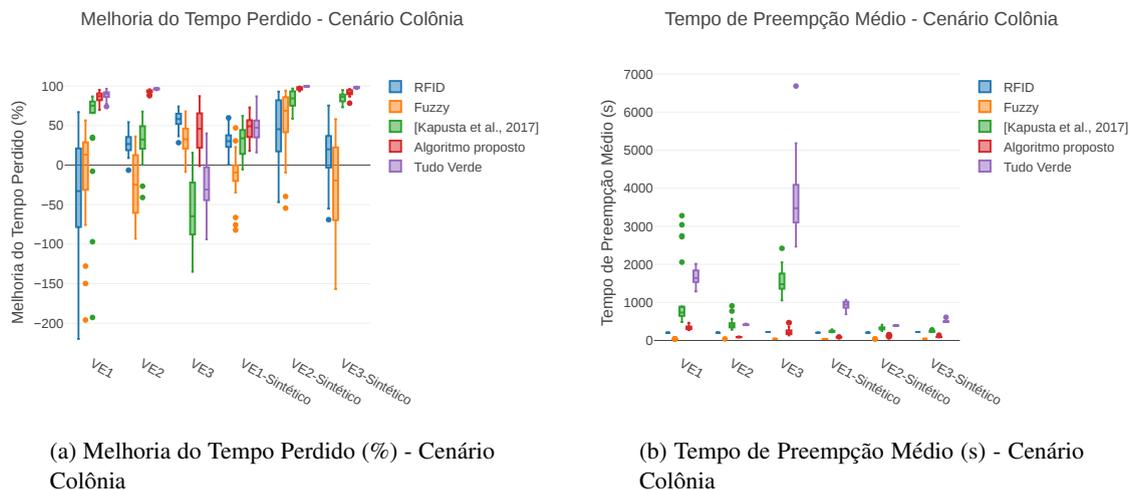


Figura 4: Resultados para o Cenário Colônia.

6. Trabalhos Futuros

Apesar de usarmos vários VEs nos experimentos, apenas um VE é inserido por vez em cada execução. Usamos essa estratégia para avaliar diferentes rotas numa mesma cidade. Em um cenário real com vários VEs, se os VEs compartilharem algumas interseções, o algoritmo de controle deve incluir uma política para decidir qual deles deve ter prioridade sobre os demais.

A falta de detecção de situações que bloqueiam o VE (*e.g.*, acidentes, redução do número de faixas em uma via etc.) e fazem com que as preempções fiquem ativas se mostrou um problema em todos os algoritmos e deve ser estudada em trabalhos futuros.

Finalmente, o algoritmo proposto não utiliza informações de outros semáforos ao longo do trajeto do VE. Acreditamos que se elas forem utilizadas de forma coordenada, poderemos atingir desempenhos ainda melhores. Porém, mecanismos de segurança devem também ser estudados para evitar que a preempção não cause acidentes. Por fim, esperamos integrar a nossa solução em um *middleware* de Cidade Inteligente [Batista et al., 2016].

7. Conclusão

Este trabalho apresenta o projeto e a avaliação de um algoritmo de preempção eficiente para melhorar o tempo de viagem de veículos de emergência. O algoritmo utiliza um melhor gerenciamento de filas e preempção que os algoritmos existentes na literatura. Além de diminuir substancialmente o tempo perdido pelo VE no trânsito, o algoritmo proposto ainda apresenta um Tempo De Preempção Médio inferior aos demais, o que auxilia o trânsito em vias concorrentes. Simulações extensivas em cenários reais e sintéticos mostram que, considerando o contexto experimental estabelecido, a solução reduz significativamente o tempo de viagem do VE, obtendo melhores resultados que as implementações dos outros algoritmos propostos para esse problema.

Agradecimentos

Esta pesquisa é parte do INCT da Internet do Futuro para Cidades Inteligentes, financiado por CNPq (proc. 465446/2014-0), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e FAPESP (procs. 14/50937-1 e 15/24485-9).

Referências

- Almuraykhi, K. M. e Akhlaq, M. (2019). STLS: Smart Traffic Lights System for Emergency Response Vehicles. In *2019 ICCIS*, p. 1–6, Sakaka, Saudi Arabia. IEEE, IEEE.
- Batista, D. M., Goldman, A., Hirata Jr., R., Kon, F., Costa, F. M., e Endler, M. (2016). InterSCity: Addressing Future Internet Research Challenges for Smart Cities. In *2016 7th NOF*, p. 1–6, Buzios, Brasil. IEEE. ISBN 978-1-5090-4671-3.
- Collura, J., Chang, J., Willhaus, E. W., e Gifford, J. (2001). Traffic signal preemption and priority: Technologies, past deployments, and system requirements. In *ITS America 11th Annual Meeting and Exposition, ITS: Connecting the Americas*.
- de Branco, R. G., Ferreira, R. A., e Cáceres, E. N. (2022). Traffic-Light Control for Emergency Vehicles Repository. <https://github.com/rodrigogbranco/traffic-light-control>.
- Djahel, S., Smith, N., Wang, S., e Murphy, J. (2015). Reducing emergency services response time in smart cities: An advanced adaptive and fuzzy approach. In *2015 ISC2*, p. 1–8, Guadalajara, Mexico. IEEE.

- Hajiebrahimi, S. e Iranmanesh, S. (2018). An adaptive control method of traffic signal-timing under emergency situations for smart cities. In *2018 3rd ICITE*, p. 225–230, Singapore. IEEE.
- Kamalanathsharma, R. K. e Hancock, K. L. (2010). Congestion-based emergency vehicle preemption. Technical report, Mid-Atlantic Universities Transportation Center, Pennsylvania, PA, USA.
- Kapusta, B., Miletić, M., Ivanjko, E., e Vujić, M. (2017). Preemptive traffic light control based on vehicle tracking and queue lengths. In *2017 International Symposium ELMAR*, p. 11–16. IEEE.
- Khan, A., Ullah, F., Kaleem, Z., Rahman, S. U., Anwar, H., e Cho, Y.-Z. (2018). EVP-STC: Emergency Vehicle Priority and Self-Organising Traffic Control at Intersections Using Internet-of-Things Platform. *IEEE Access*, 6:68242–68254.
- Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., e Wiessner, E. (2018). Microscopic traffic simulation using sumo. In *2018 21st ITSC*, p. 2575–2582, Maui, HI, USA. IEEE.
- Marshall, P. S. e Berg, W. D. (1997). Design guidelines for railroad preemption at signalized intersections. *ITE journal*, 67(2).
- Miyawaki, M., Yamashiro, Z., e Yoshida, T. (1999). Fast emergency pre-emption systems (fast). In *Proceedings 199 IEEE/IEEJ/JSAI IITSC*, p. 993–997, Tokyo, Japan. IEEE.
- Moroi, Y. e Takami, K. (2015). A method of securing priority-use routes for emergency vehicles using inter-vehicle and vehicle-road communication. In *2015 7th NTMS*, p. 1–5, Paris, France. IEEE.
- Mu, H., Song, Y., e Liu, L. (2018). Route-Based Signal Preemption Control of Emergency Vehicle. *Journal of Control Science and Engineering*, 2018.
- Paniati, J. F. e Amoni, M. (2006). *Traffic Signal Preemption for Emergency Vehicles: A Cross-Cutting Study*. U.S. Department of Transportation, 400 7th Street SW, Washington, DC 20590.
- Pishue, B. (2021). 2021 INRIX Global Traffic Scorecard.
- Qin, X. e Khan, A. M. (2012). Control strategies of traffic signal timing transition for emergency vehicle preemption. *Transportation Research Part C: Emerging Technologies*, 25:1–17. ISSN 0968-090X.
- Rapelli, M., Casetti, C., e Gagliardi, G. (2021). Vehicular traffic simulation in the city of turin from raw data. *IEEE TMC*, p. 1–1.
- Stephanopoulos, G., Michalopoulos, P. G., e Stephanopoulos, G. (1979). Modelling and analysis of traffic queue dynamics at signalized intersections. *Transportation Research Part A: General*, 13 (5):295–307. ISSN 0191-2607.
- Thompson, W. R. (1936). On confidence ranges for the median and other expectation distributions for populations of unknown distribution form. *The Annals of Mathematical Statistics*, 7(3):122–128. ISSN 00034851.
- Varschen, C. e Wagner, P. (2006). Mikroskopische modellierung der personenverkehrsnachfrage auf basis von zeitverwendungstagebüchern. *Integrierte Mikro-Simulation von Raum- und Verkehrsentwicklung. Theorie, Konzepte, Modelle, Praxis*, 81:63–69.