

Orrana Lhaynher Veloso de Sousa
Orientador(a): Deborah Maria Vieira Magalhães

Metodologia para Recomendação de Rotas de Evacuação em Cenário de Desastre Natural

Picos - PI
22 de novembro de 2019

Orrana Lhaynher Veloso de Sousa
Orientador(a): Deborah Maria Vieira Magalhães

Metodologia para Recomendação de Rotas de Evacuação em Cenário de Desastre Natural

Trabalho de Conclusão de Curso sobre uma
Metodologia de Recomendação de Rotas em
Desastres Naturais do Bacharelado em Siste-
mas de Informação pela Universidade Federal
do Piauí.

Universidade Federal do Piauí
Campus Senador Helvídio Nunes de Barros
Bacharelado em Sistemas de Informação

Picos - PI
22 de novembro de 2019

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

S725m Sousa, Orrana Lhaynher Veloso de.
Metodologia para recomendação de rotas de evacuação em cenário de desastre natural. / Orrana Lhaynher Veloso de Sousa.
-- Picos,PI, 2019.
69 f.
CD-ROM: 4 ¾ pol.

Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação. – Universidade Federal do Piauí, Picos, 2019.
“Orientador(A): Profa. Deborah Maria Vieira Magalhães.”

1. Desastre Natural. 2. Método de Evacuação. 3. Tecnologia – Prevenção Desastre. I. Título.

CDD 005

Elaborada por Rafael Gomes de Sousa CRB 3/1163

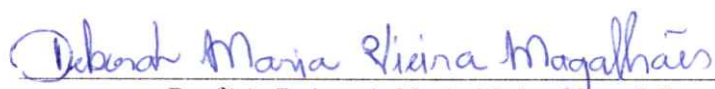
METODOLOGIA PARA RECOMENDAÇÃO DE ROTAS DE EVACUAÇÃO EM CENÁRIO
DE DESASTRE NATURAL

ORRANA LHAYNHER VELOSO DE SOUSA

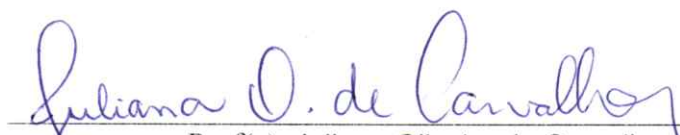
Monografia Aprovada como exigência parcial para obtenção do
grau de Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI, 03 de dezembro de 2019


Prof(a). Deborah Maria Vieira Magalhães


Prof(a). Romuere Rodrigues Veloso e Silva


Prof(a). Juliana Oliveira de Carvalho

Agradecimentos

Agradeço a Deus, que esteve comigo desde o primeiro dia e é a Ele que dedico a concretização deste trabalho.

Agradeço aos meus pais por sempre me incentivarem e acreditarem na minha capacidade de superar os obstáculos que a vida me apresentou. Além disso, agradeço também por demonstrarem muito amor e apoio, o que serviu de alicerce para as minhas realizações.

Agradeço a minha irmã, por todo o apoio e ajuda nos momentos difíceis.

Agradeço aos meus amigos, que me incentivaram e me ajudaram ao longo desta caminhada.

Agradeço a minha orientadora Deborah pelo apoio, correções, incentivo e suporte.

Agradeço a esta universidade, o corpo docente de Sistemas de Informação, direção e administração por toda a ajuda e educação dada.

Agradeço a toda a equipe do MATSim por desenvolver um software de excelente qualidade e por também me auxiliar com importantes dúvidas.

Agradeço a Simunto, empresa Suíça, por fornecer uma licença do software Via, possibilitando uma melhor visualização e análise dos resultados das simulações.

Por fim, agradeço a todos que de forma direta ou indireta contribuíram para a realização deste sonho.

Todas as conquistas começam com o simples ato de acreditar que elas são possíveis.
Autor Desconhecido

Resumo

Devido a geodinâmica terrestre, desastres naturais são fenômenos recorrentes que evidenciam a fragilidade da vida humana e não humana. Dentre estes, podemos citar incêndios, furacões, terremotos e outros, que destroem direta ou indiretamente infraestruturas, causando danos sociais que afetam indivíduos, cidades e países. O homem desenvolve métodos e tecnologias, como planejamento adequado e sistemas de notificação, a fim de prevenir ou reduzir os impactos de desastres. Como resultado desses fenômenos, indivíduos são levados à miséria, ficando sem moradia, saneamento básico e alimentos. Assim, a manutenção da vida é certamente o principal propósito de residentes, que buscam fugir de suas localidades através de rotas de evasão. Contudo, a definição de uma rota de fuga não é uma tarefa trivial devido a, por exemplo, congestionamentos, bloqueio de vias, abrigos indisponíveis, entre outros. Então, este trabalho propõe um método para a sugestão de rotas e simulação de demanda de evacuação baseada em agente. Essa sugestão será feita com a definição dos pontos de origem - através de coordenadas na área afetada, e pontos de destino - através da seleção do abrigo mais próximo e com disponibilidade de vagas de cada evacuado. Após isso, a recomendação será feita levando em consideração a disponibilidade de vias na rede rodoviária e a redução de possíveis congestionamentos, recomendando assim a melhor rota para cada evacuado nas circunstâncias dadas. Para testar a eficácia do método, é apresentada a simulação de um estudo de caso para o incêndio florestal *Camp Fire* no condado de *Butte* na Califórnia que aconteceu em novembro de 2018. Os resultados do estudo de caso demonstram que o método pode fornecer rotas de evasão que impactam positivamente no tempo de evacuação global.

Palavras-chaves: desastre natural, simulação baseada em agente, método de evacuação.

Abstract

Due to terrestrial geodynamics, natural disasters are recurrent phenomena that highlight the fragility of human and non-human life. These include fires, hurricanes, earthquakes and others that directly or indirectly destroy infrastructure, causing social damage affecting individuals, cities and countries. Man develops methods and technologies, such as proper planning and reporting systems, to prevent or reduce the impacts of disasters. As a result of these phenomena, individuals are driven into misery, homeless, sanitation and food. Thus, the maintenance of life is certainly the main purpose of residents, who seek to escape from their locations through escape routes. However, defining an escape route is not a trivial task due to, for example, traffic jams, roadblocks, unavailable shelters, among others. Thus, this paper proposes a method for route suggestion and agent-based evacuation demand simulation. This suggestion will be made by defining the points of origin - through coordinates in the affected area, and points of destination - by selecting the nearest shelter and availability of vacancies for each evacuees. Thereafter, the recommendation will be made taking into account the availability of roads on the road network and the reduction of possible congestion, thus recommending the best route for each evacuated under the given circumstances. To test the effectiveness of the method, we present a case study simulation for the Camp Fire forest fire in Butte County, California that took place in November 2018. The case study results demonstrate that the method can provide evasion routes that positively impact overall evacuation time.

Lista de ilustrações

| | |
|---|----|
| Figura 1 – Algoritmo de Dijkstra. Fonte: Baeldung (2019). | 19 |
| Figura 2 – Modelo de fluxo de tráfego do MATSim. Adaptado de Horni e Axhausen (2019). | 20 |
| Figura 3 – Algoritmo co-evolucionário do MATSim. Fonte: Horni e Axhausen (2019). | 21 |
| Figura 4 – O ciclo de vida da simulação no MATSim. Adaptado de Horni e Axhausen (2019). | 22 |
| Figura 5 – Incêndio <i>Camp Fire</i> visto do satélite Landsat 8, em 08 de novembro de 2018. Escala: 8 km por cm. Fonte: SFGATE (2018). | 31 |
| Figura 6 – Extensão da fumaça provocada pelo <i>Camp Fire</i> . Fonte: SFGATE (2018). | 32 |
| Figura 7 – Área queimada pelo incêndio <i>Camp Fire</i> . Escala: 2 km por cm. Fonte: Newsweek (2018). | 33 |
| Figura 8 – Metodologia de recomendação de rotas. | 34 |
| Figura 9 – Setores do cronograma de evacuação. Adaptado de Times (2018). . . . | 35 |
| Figura 10 – Pontos de origem dos evacuados. | 36 |
| Figura 11 – Localização de abrigos no mapa da área simulada. | 37 |
| Figura 12 – Fluxo de extração de dados do <i>OpenStreetMap</i> | 39 |
| Figura 13 – Rede rodoviária MATSim após a conversão. | 40 |
| Figura 14 – Instante em que toda a população entra na rede rodoviária na simulação 01. | 42 |
| Figura 15 – Instante em que toda a população entra na rede rodoviária na simulação 02. | 43 |
| Figura 16 – Instante em que toda a população entra na rede rodoviária na simulação 03. | 44 |
| Figura 17 – Distribuição dos agentes no abrigo <i>Glenn County Fairgrounds</i> na simulação nativa do MATSim. | 45 |
| Figura 18 – Três evacuados na mesma área sendo enviados para três abrigos diferentes. | 45 |
| Figura 19 – Distribuição dos agentes no abrigo <i>Glenn County Fairgrounds</i> na simulação da metodologia do trabalho. | 46 |
| Figura 20 – Três evacuados sendo enviados para os dois abrigos mais próximos. . . | 46 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 – Trabalhos relacionados. | 30 |
| Tabela 2 – Abrigos selecionados como destinos de evacuação. | 36 |
| Tabela 3 – Dados de dez evacuados da simulação nativa. | 47 |
| Tabela 4 – Dados de dez evacuados da segunda simulação. | 48 |
| Tabela 5 – Dados de dez evacuados da simulação da metodologia completa. | 48 |
| Tabela 6 – Tempo de evacuação total e congestionamentos. | 49 |

Lista de abreviaturas e siglas

| | |
|----------|--|
| OSM | OpenStreetMap |
| MATSim | Multi-Agent Transportation Simulator |
| XML | eXtensible Markup Language |
| GPS | Global Positioning System |
| ONU | Organização das Nações Unidas |
| MOEA | Multi-Objective Evolutionary Algorithms |
| JICA | Japan International Cooperation Agency |
| TDMMC | Tehran Disaster Mitigation and Management Center |
| WGS 84 | World Geodetic System |
| EPSG | European Petroleum Survey Group |
| EEP | Emergency Evacuation Planning |
| IP | Integer Programming |
| TRANSIMS | Transportation Analysis Simulation System |

Lista de símbolos

Σ Letra grega Sigma

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 14 |
| 1.1 | Contextualização | 14 |
| 1.2 | Objetivos | 15 |
| 1.3 | Metodologia | 15 |
| 1.4 | Contribuições | 15 |
| 1.5 | Estrutura do trabalho | 16 |
| 2 | Referencial teórico | 17 |
| 2.1 | Métodos de roteamento de evacuação | 17 |
| 2.2 | Simulação baseada em agentes | 17 |
| 2.3 | <i>OpenStreetMap</i> | 18 |
| 2.4 | Algoritmo Dijkstra | 18 |
| 2.5 | Algoritmo A* | 19 |
| 2.6 | Simulador de transporte multiagente | 19 |
| 2.6.1 | Modelo de fluxo de tráfego | 20 |
| 2.6.2 | Algoritmo co-evolutivo do MATSim | 20 |
| 2.6.3 | Etapas principais do processo de simulação e otimização | 21 |
| 2.6.4 | Arquivos de entrada | 23 |
| 2.6.5 | Equilíbrio de <i>Nash</i> | 25 |
| 2.7 | Via | 25 |
| 3 | Trabalhos relacionados | 26 |
| 3.1 | Trabalhos selecionados | 26 |
| 4 | Proposta | 31 |
| 4.1 | Cenário de desastre | 31 |
| 4.2 | Método | 33 |
| 4.2.1 | Definição do ponto de origem | 34 |
| 4.2.2 | Definição do ponto de destino | 36 |
| 4.2.3 | Recomendação de rota | 37 |
| 4.3 | Simulação | 38 |
| 4.3.1 | Criação da rede de simulação | 39 |
| 4.3.2 | Criação da população MATSim | 40 |
| 4.3.3 | Criação do arquivo de configuração | 40 |
| 5 | Resultados | 42 |

| | | |
|----------|---|-----------|
| 5.1 | Visualização dos eventos de simulação | 42 |
| 5.2 | Análise | 44 |
| 5.2.1 | Distribuição de evacuados pelos abrigos | 44 |
| 5.2.2 | Tempo de evacuação individual | 47 |
| 5.2.3 | Tempo de evacuação total | 48 |
| 6 | Conclusão | 50 |
| 7 | Publicações | 51 |
| | Referências | 52 |
| | Apêndices | 54 |
| | APÊNDICE A Script para a criação da rede MATSim | 55 |
| | APÊNDICE B Script para a criação da população MATSim | 56 |
| | APÊNDICE C Script para a criação do arquivo <i>networkChangeEvents</i> | 62 |
| | APÊNDICE D Script para a minimização de congestionamentos | 64 |
| | APÊNDICE E Script para simular descongestionamento | 66 |
| | APÊNDICE F Arquivo de configuração utilizado na simulação | 68 |

1 Introdução

1.1 Contextualização

Desastres naturais são fenômenos que ocorrem em função das mudanças climáticas que acontecem em nosso planeta. Dentre estes, desastres climáticos e geofísicos, como terremotos e tsunamis, mataram 1,3 milhão de pessoas nos últimos 20 anos e deixaram mais 4,4 bilhões de feridos, desabrigados ou em necessidade de ajuda de emergência, segundo [ONUBR \(2018\)](#). Estes são números estratégicos, se forem considerados o que eles significam em termos de oportunidades destruídas, de pessoas que perderam suas vidas, casas, escolas e hospitais.

Em cenários de desastre, o principal objetivo da população é escapar da área afetada, consequentemente poupando vidas. De acordo com [Mattar \(2011\)](#), o deslocamento humano causado por efeitos climáticos adversos tem sido por muito tempo uma estratégia natural de adaptação à variação do meio ambiente. Dentre os tipos de deslocamentos populacionais, a migração intra-urbana se destaca, sendo feita através de estradas, na maioria dos casos, que podem levar para destinos seguros, fora da área de risco, e/ou para abrigos de segurança.

Entre as escolhas de destino de evacuação, deve-se destacar a indicação de abrigos como rota final. Esta recomendação tem como fatores propícios a localização em uma área segura e a disponibilidade de suprimentos, refeições, serviços médicos e outros serviços de apoio. Para aqueles que são incapazes de chegar a suas casas ou cujas casas são impossíveis de viver, esta opção mitiga a preocupação de onde alojar-se e oferece uma direção do que fazer após um desastre.

Com isso, deve-se considerar a rota a ser tomada e qual abrigo a ser recomendado, pois estes podem influenciar diretamente no tempo gasto na evasão. Em casos assim, a escolha errônea da rota e do abrigo terá grande impacto no número de vítimas em desastres naturais. Além disso, em áreas densamente povoadas aumenta-se também a probabilidade de que não haja vagas e, consequentemente, recursos disponíveis para todos, fazendo com que a escolha do abrigo seja um fator de extrema importância. Desse modo, um algoritmo de sugestão de rotas em que se considere vários fatores se faz imprescindível, já que considerar apenas um aspecto em uma evasão num cenário de desastre pode afetar a segurança dos evacuados.

Congruente com a importância da recomendação, a simulação deste tipo de cenário traz importantes benefícios, visto que possibilita a avaliação de métodos de evacuação e seus possíveis resultados. Em função disso, diversas ferramentas de simulação têm sido desenvolvidas, como o Simulador de Transporte Multiagente, também conhecido como

MATSim. Este simulador contém características relevantes para a simulação de evacuação em desastres naturais, tais como a modelagem baseada em agentes, a capacidade de suportar cenários em larga escala e a criação de uma solução ideal para o cenário em tempo aceitável (MATSIM, 2019).

1.2 Objetivos

Este trabalho tem como objetivo desenvolver um método de recomendação de rotas que auxilie na evacuação de residentes em desastres naturais, tendo como fator de tomada de decisão a lotação e distância de abrigos, a disponibilidade de vias e a redução de congestionamentos.

Os objetivos específicos são:

1. Possibilitar a evacuação em um tempo considerável;
2. Recomendar a rota apropriada para o usuário considerando vias e abrigos disponíveis;
3. Reduzir o tempo gasto em congestionamentos;
4. Avaliar o desempenho da metodologia utilizada através da simulação do cenário de desastre para a região do Condado de *Butte*, Califórnia.

1.3 Metodologia

A metodologia do presente trabalho reside na implementação de um método de sugestão de rota de evacuação considerando os parâmetros lotação e distância de abrigos, disponibilidade de vias e redução de congestionamentos, com o intuito de minimizar o número de vítimas. Ela está dividida em três etapas: definição de ponto de origem, definição de ponto de destino e recomendação de rota. Em complemento, uma simulação de demanda de evacuação foi construída para o incêndio florestal *Camp Fire*, Califórnia, com o propósito de analisar e validar o método.

1.4 Contribuições

As contribuições desse trabalho se dividem em duas categorias: contribuições para a criação de cenários de desastre e contribuições para o simulador MATSim. Na primeira categoria, as contribuições foram:

1. Modelagem de um cenário com base em um desastre real;
2. Realização de um cronograma de evacuação;

3. Definição de lotação em pontos de abrigos;
4. Recomendação de rota considerando estradas disponíveis, congestionamentos e disponibilidade de abrigos.

Já na segunda categoria, a contribuição feita foi:

1. Criação de um algoritmo que remove rotas indisponíveis na rede rodoviária.

1.5 Estrutura do trabalho

Este trabalho está estruturado da seguinte forma:

- **Capítulo 2** - são apresentados os conceitos importantes sobre este trabalho;
- **Capítulo 3** - são mencionados os trabalhos correlatos a este;
- **Capítulo 4** - está exposta em detalhes a metodologia de desenvolvimento deste trabalho;
- **Capítulo 5** - é apresentado a conclusão deste projeto.

2 Referencial teórico

Nesta seção serão descritos conteúdos relevantes para o entendimento do trabalho. Na subseção 2.1 serão discutidos os métodos de roteamento de evacuação. Já na subseção 2.2, será apresentada a Modelagem baseada em agentes. Na subseção 2.3, será abordado o *OpenStreetMap*, um mapa livre e editável do mundo. Na subseção 2.4 e 2.5, o algoritmo de Dijkstra e o algoritmo A* são comentados, respectivamente. Enquanto que na subseção 2.6, o Simulador de Transporte Multiagente que será a *framework* de simulação é abordado. Por fim, na subseção 2.7, a ferramenta de visualização utilizada no projeto é discutida.

2.1 Métodos de roteamento de evacuação

De acordo com [Shahabi e Wilson \(2014\)](#), os métodos de roteamento de evacuação podem ser divididos em métodos descritivos e prescritivos. Os métodos descritivos são soluções que simulam visualmente uma dada situação de emergência. O objetivo é reproduzir a realidade o mais precisamente possível. Alguns dos métodos que se enquadram nessa categoria são: simulações baseadas em fluxo, simulações baseadas em agentes e simulações baseadas em atividades. O uso desse método traz benefícios à pesquisadores e desenvolvedores na área, pois permite a avaliação de políticas de evacuação antes de serem colocadas em prática em um ambiente similar ao real. Com isso, o maior número de pessoas pode ser salvo ao se adotar uma política que satisfaça objetivos escolhidos para uma evacuação em particular.

De outro modo, os métodos prescritivos determinam as estratégias ideais de roteamento de evacuação para alcançar algum objetivo de evacuação sem obrigatoriamente executar uma simulação. Ou seja, este tipo de método fornece um plano de evacuação em vez de visualizar um plano. Esses métodos usualmente utilizam técnicas matemáticas para coletar dados e recomendar rotas e/ou horários. O propósito é minimizar o tempo de evacuação, reduzir o congestionamento do tráfego ou diminuir a exposição ao perigo. Conseqüentemente, essas soluções são menos dependentes da geografia e geralmente são mais escaláveis.

2.2 Simulação baseada em agentes

Na Simulação Baseada em Agentes, componentes ativos ou tomadores de decisão são modelados e implementados usando conceitos e tecnologias relacionados a agentes. Ela foi fundamentada a partir de um método de modelagem de sistemas complexos que define

regras e comportamentos para componentes individuais (agentes), bem como o ambiente em que eles estão presentes. Cada agente então avalia individualmente sua situação e toma decisões com base no seu conjunto de regras definido. Com isso, o curso e comportamento de eventos podem ser observados através das interações entre eles.

Nesse tipo de simulação, agentes podem ser pessoas, animais, veículos, equipamentos, células, componentes de um sistema, etc. Eles são capazes de evoluir, permitindo que comportamentos imprevistos surjam, podendo exibir padrões de comportamento complexos e fornecendo informações valiosas sobre a dinâmica do sistema do mundo real que ele emula. A modelagem em que se baseia este tipo de simulação às vezes incorpora redes neurais, algoritmos evolutivos ou outras técnicas de aprendizado para permitir aprendizado e adaptação realistas.

2.3 *OpenStreetMap*

O *OpenStreetMap* (OSM) é um projeto de mapeamento colaborativo para criar um mapa livre e editável do mundo, [OpenStreetMap \(2019\)](#). É constituído por dados abertos: qualquer pessoa tem a liberdade de usar os dados para qualquer fim. Ele fornece dados a centenas de *sites* na *Internet*, aplicações de celular e outros dispositivos. Os mapas foram desenvolvidos e são mantidos com rigor por sua comunidade voluntária de mapeadores, que inserem e revisam dados de receptores GPS portáteis, fotografias aéreas, imagens de satélite e outras fontes livres.

A grande disponibilidade de informações de localização de alta qualidade propiciou o mapeamento do mercado de massa com base em receptores GPS acessíveis, computadores domésticos e a *Internet*. Embora uma série de projetos baseados em mapeamento criado pelo usuário tenha aparecido, "o OSM é provavelmente o projeto mais extenso e efetivo atualmente em desenvolvimento", [Haklay e Weber \(2008, p 2\)](#).

2.4 Algoritmo Dijkstra

O algoritmo de Dijkstra, concebido pelo cientista da computação holandês Edsger Dijkstra em 1956 e publicado em 1959¹, soluciona o problema do caminho mais curto num grafo dirigido ou não dirigido com arestas de peso não negativo, em tempo computacional $O(m + n \log n)$ ², onde m é o número de arestas e n é o número de vértices. Este algoritmo calcula o caminho mais curto de acordo com o custo mínimo entre as vértices de um grafo. Escolhido um vértice como raiz da busca, o algoritmo calcula o custo mínimo deste vértice para todos os demais vértices do grafo.

¹ **Algoritmo de Dijkstra**. Disponível em <http://bit.ly/334R1SE>. Acesso em 20 de novembro e 2019.

² **Algoritmo de Dijkstra!**; eximiaCo.tech. Disponível em <http://bit.ly/337ggDX>. Acesso em 20 de novembro de 2019.

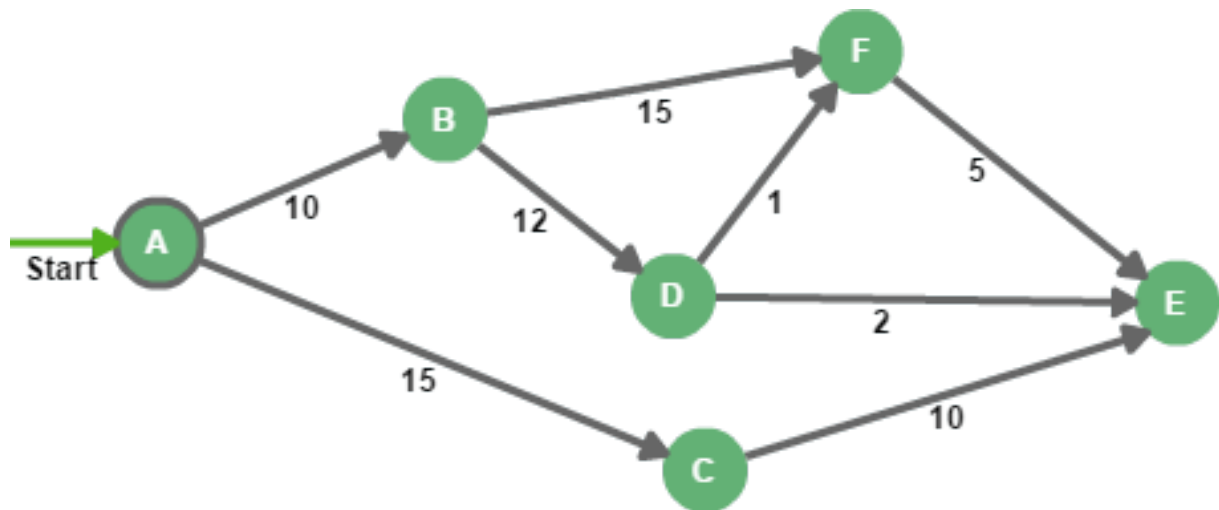


Figura 1: Algoritmo de Dijkstra. Fonte: Baeldung (2019).

2.5 Algoritmo A*

O A*, ou "A-star", é um algoritmo de busca de caminho e de grafo, que é frequentemente usado em ciência da computação devido ser um algoritmo completo e com eficiência ideal. Uma grande desvantagem prática é sua $O(b^d)$ complexidade do espaço, pois armazena todos os nós gerados na memória. Assim, em sistemas práticos de roteamento de viagem, geralmente é superado por algoritmos que podem pré-processar o gráfico para obter melhor desempenho, bem como abordagens limitadas à memória; no entanto, A* ainda é a melhor solução em muitos casos, de acordo com Zeng e Church (2009).

2.6 Simulador de transporte multiagente

O Simulador de Transporte Multiagente, em inglês *Multi-Agent Transportation Simulator* (MATSim), é uma ferramenta que oferece uma estrutura para realizar simulações de transporte baseadas em agentes em larga escala. Cada agente (pessoa) otimiza repetidas vezes seu cronograma diário de atividades ao mesmo tempo que compartilha espaços na infraestrutura de transporte com os demais agentes ao longo da simulação. Tal cronograma representa uma lista de atividades, como, por exemplo, estar em *casa* ou no *trabalho* e se deslocar de *carro* até o shopping). O cronograma conta ainda com informações temporais, por exemplo, sair de casa às 7:00, e informações adicionais, como a rota detalhada de casa pro shopping. Em uma simulação típica do MATSim, os dados referentes a trajetória a qual se deseja simular são otimizados em uma rede de transporte definida, esse é um dos recursos fundamentais do simulador, tornando-o adequado para uso em estudos de políticas de evacuação.

2.6.1 Modelo de fluxo de tráfego

O MATSim oferece dois mobsims internos: Simulação Baseada em Fila (QSim - *Queue Simulation*) e Simulação de Fila de Eventos Discretos Java (JDEQSim - *Java Discrete Event Queue Simulation*); além disso, simulações de mobilidade externa podem ser adicionadas. Ele utiliza a abordagem baseada em fila computacionalmente eficiente, como mostrado na Figura 2. Um carro que entra em um *link* de rede (ou seja, um segmento de estrada) a partir de um cruzamento é incluído no fim da fila de espera. Ele mantém-se lá até que o tempo para percorrer o *link* com fluxo livre tenha passado, ele esteja no início da fila de espera e o próximo *link* permita entrada.

O modelo de fluxo de tráfego do MATSim é vigorosamente baseado nos dois atributos de *link*: capacidade de armazenamento e capacidade de fluxo. A capacidade de armazenamento define o número de carros que cabem em um *link* de rede, enquanto a capacidade de fluxo especifica a capacidade de saída de um *link*, ou seja, quantos viajantes podem deixar o respectivo *link* por etapa de tempo, sendo assim um atributo individual do *link*.

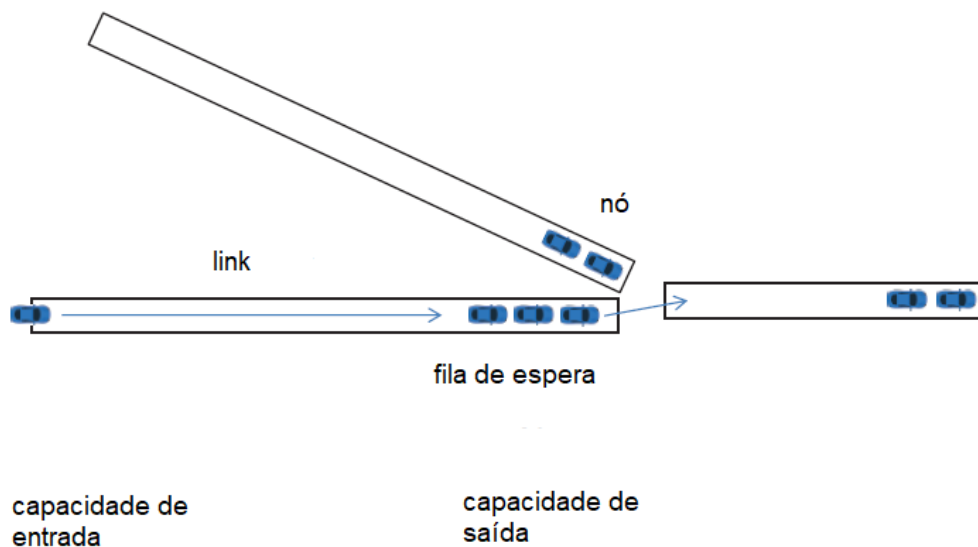


Figura 2: Modelo de fluxo de tráfego do MATSim. Adaptado de [Horni e Axhausen \(2019\)](#).

2.6.2 Algoritmo co-evolutivo do MATSim

O equilíbrio do MATSim é obtido através de um algoritmo co-evolutivo. Esses algoritmos co-evoluem espécies distintas sujeitas a interação (por exemplo, competição). No MATSim, os indivíduos são descritos por seus planos, onde uma pessoa simboliza uma espécie. Com o algoritmo co-evolutivo, a otimização é realizada em termos de planos dos agentes, ou seja, em todo o plano diário de atividades e viagens. Ele obtém mais do que os equilíbrios de tráfego padrão, que desconsideram as atividades. Eventualmente, é obtido

um equilíbrio, sujeito a restrições, onde os agentes não podem melhorar ainda mais seus planos unilateralmente.

O MATSim não utiliza um algoritmo evolutivo ao invés de um algoritmo co-evolutivo, pois o primeiro levaria a um sistema ideal, já que a otimização seria empregada com uma função de adequação global (ou populacional). Em vez disso, o algoritmo co-evolutivo provoca um equilíbrio (estocástico) do usuário, pois a otimização é realizada em termos de funções de pontuação individuais e dentro do conjunto de planos de um agente.

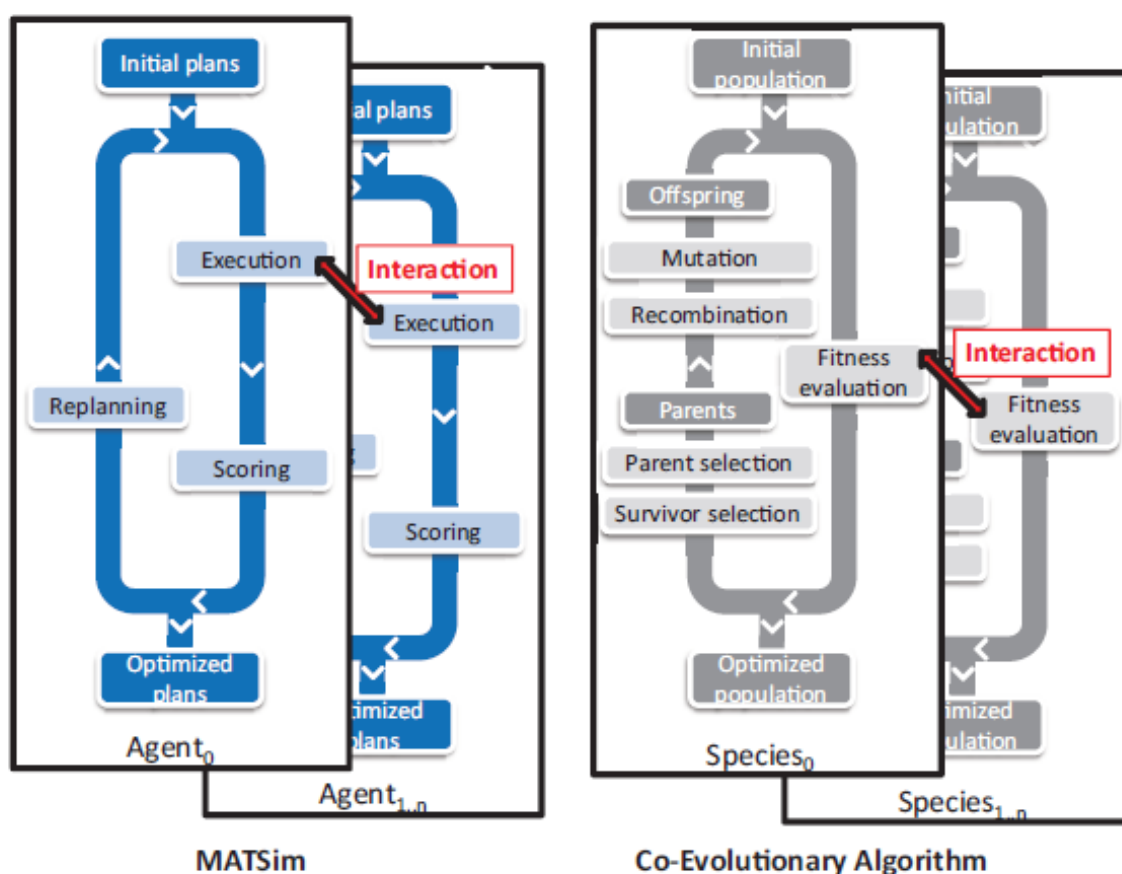


Figura 3: Algoritmo co-evolucionário do MATSim. Fonte: Horni e Axhausen (2019).

2.6.3 Etapas principais do processo de simulação e otimização

Conforme ilustrado na Figura 4, as cinco etapas principais são: demanda inicial, execução, pontuação, replanejamento e análise.

Na primeira etapa, a *Demanda* inicial descreve o comportamento de mobilidade a ser simulado. Abrange uma lista completa de agentes com seus respectivos cronogramas diários. Na etapa de execução, também conhecida como Simulação de Mobilidade (mobsim) todos os planos dos agentes são executados juntos. Isto é, agentes e veículos são simulados na rede rodoviária do cenário. Desse modo, um agente pode impactar em outros agen-

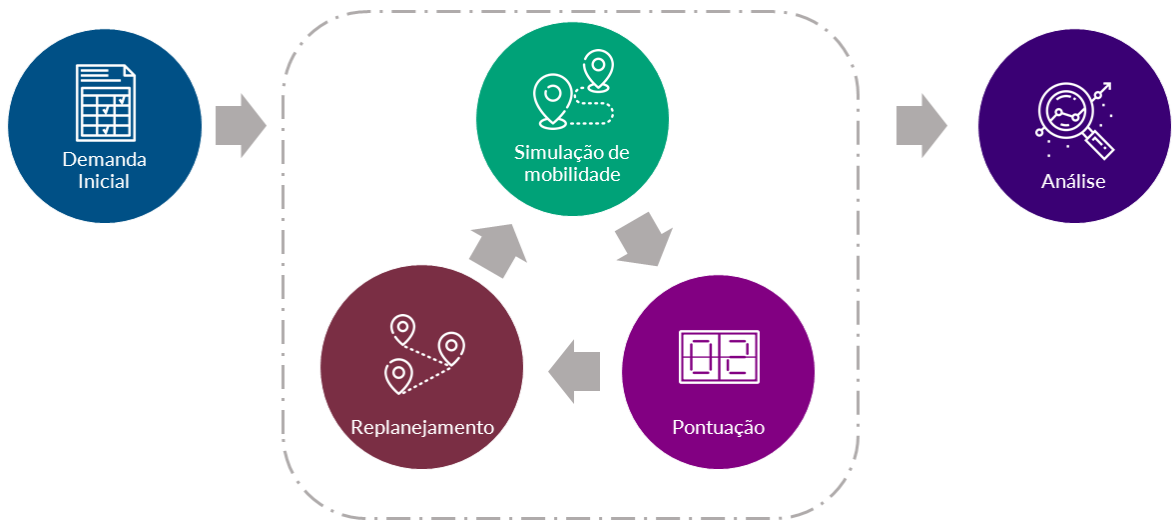


Figura 4: O ciclo de vida da simulação no MATSim. Adaptado de [Horni e Axhausen \(2019\)](#).

tes, podendo ocasionar, por exemplo, engarrafamentos se vários agentes decidem transitar pela mesma estrada.

Após a execução, dá-se início a etapa de *Pontuação*, onde cada plano é avaliado e recebe um determinado número de pontos. A função de pontuação é customizável mas, por padrão, o tempo gasto nas atividades aumenta a pontuação, enquanto o tempo gasto na viagem o diminui. Desse modo, agentes em engarrafamentos perdem pontos, enquanto agentes que têm viagens rápidas e curtas acumulam mais pontos, executando atividades por um maior período. Para a função padrão, a utilidade de um plano (S_{plan}) é calculada como a soma de todas as utilidades de atividade ($S_{act,q}$) mais a soma de todas as (des)utilidades de viagem ($S_{trav,mode(q)}$) com N como o número de atividades, como mostrado na Equação 2.1. A (*trip*) q é a viagem que segue a atividade q . Para a pontuação, a última atividade é mesclada com a primeira para produzir um número igual de viagens e atividades.

$$S_{plan} = \sum_{q=0}^{N-1} S_{act,q} + \sum_{q=0}^{N-1} S_{trav,mode(q)}. \quad (2.1)$$

Como discutido anteriormente, agentes podem ser influenciados por outros, provocando congestionamentos em vias. Portanto, na etapa de *Planejamento*, os agentes podem modificar seus planos visando melhorar as suas respectivas pontuações. Dessa forma, horários de início e fim de atividades podem ser alterados na simulação de mobilidade a fim de evitar situações que levem a pontuações ruins.

Na etapa de *Análise*, as ações ou eventos realizados na simulação são agregados para avaliar qualquer medida na resolução desejada, como compartilhamentos de modo (carro, táxi, bicicleta), total de milhas percorridas pelos agentes, duração média das viagens e

distância por modo e hora. Tais análises podem ser executadas automaticamente no final ou em uma etapa separada qualquer de pós-processamento.

2.6.4 Arquivos de entrada

Para realizar uma simulação, os seguintes arquivos precisam necessariamente serem configurados no MATSim:

- o arquivo de configuração, que descreve o comportamento da simulação;
- o arquivo de rede, que descreve a rede rodoviária da área a ser simulada;
- e o arquivo de população, que descreve as demandas de viagem de acordo com o cenário que se deseja simular.

O MATSim utiliza o arquivo de configuração para a construção de uma ligação entre o simulador e o usuário, controlando como a simulação será feita. Cada parâmetro faz parte de exatamente um grupo de configurações, como, por exemplo, o parâmetro *inputNetworkFile* mostrado no Código 2.1. Esse grupo coleta parâmetros semelhantes ou diferentes para uma parte específica do MATSim. Configurações como sistema de coordenadas, número de iterações e tipos de atividades são definidas neste arquivo.

```
<config>
  <module name="network">
    <param name="inputNetworkFile" value="network.xml" />
  </module>
  <module name="plans">
    <param name="inputPlansFile" value="plans.xml" />
  </module>
  <module name="controler">
    <param name="firstIteration" value="0" />
    <param name="lastIteration" value="0" />
  </module>
  <module name="planCalcScore">
    <parameterset type="activityParams">
      <param name="activityType" value="h" /> <!-- home -->
      <param name="priority" value="1" />
      <param name="typicalDuration" value="12:00:00" />
      <param name="minimalDuration" value="08:00:00" />
    </parameterset>
  </module>
</config>
```

Código Fonte 2.1: Exemplo de arquivo de configuração.

O arquivo *network.html* descreve a rede em que agentes ou veículos se locomovem, como apresentado no Código 2.2. Consiste de nós e *links*, ou vértices e arestas na teoria dos grafos. Cada nó é representado por valores de coordenadas x e y, enquanto *links* têm

um número maior de características. Dentre essas características podem-se citar atributos como *from* e *to* que descrevem a geometria da rede. Além disso, todos os links são unidirecionais. Se uma via puder ser percorrida nas duas direções, dois links devem ser estipulados com os atributos *from* e *to* contrários.

```
<network>
  <nodes>
    <node id="0" x="-200.0" y="200.0" > </node>
    <node id="1" x="0.0" y="200.0" > </node>
    <node id="2" x="200.0" y="200.0" > </node>
    <node id="3" x="400.0" y="400.0" > </node>
  </nodes>
  <links cpperiod="01:00:00" effectivecellsize="7.5" effectivecellwidth="3.75">
    <link id="0_1" from="0" to="1" length="200.0" freespeed="222.2222222222223"
    capacity="3600.0" permlanes="1.0" oneway="1" modes="car" >
    </link>
    <link id="1_2" from="1" to="2" length="200.0" freespeed="222.2222222222223"
    capacity="3600.0" permlanes="1.0" oneway="1" modes="car" >
    </link>
    <link id="2_3" from="2" to="3" length="200.0" freespeed="3.33889816360601"
    capacity="1800.0" permlanes="1.0" oneway="1" modes="car" >
    </link>
  </links>
</network>
```

Código Fonte 2.2: Exemplo de arquivo de *network*.

A demanda de viagens do MATSim é descrita pelos planos diários dos agentes presentes no arquivo de população, como exposto no Código 2.3. Por isso, o arquivo de população contém essencialmente uma lista de planos diários em uma estrutura hierárquica. A população contém uma lista de pessoas, cada pessoa tem uma lista de planos, e cada plano tem uma lista de atividades. Apenas um plano é escolhido. O plano selecionado de cada agente é executado pelo mobsim, e pode ser substituído por outro durante a fase de replanejamento.

```
<population>
  <person id="car_home1_work1_0">
    <plan selected="yes">
      <activity type="home" x="-1188391.227582054" y="114456.09067770051"
      end_time="09:00:00" > </activity>
      <leg mode="car"> </leg>
      <activity type="work" x="4663945.2848932855" y="0.0"
      end_time="17:00:00" >
      </activity>
      <leg mode="car"> </leg>
      <activity type="home" x="-1188391.227582054" y="114456.09067770051"
      end_time="09:00:00" > </activity>
    </plan>
  </person>
</population>
```

Código Fonte 2.3: Exemplo de arquivo de população.

2.6.5 Equilíbrio de *Nash*

O equilíbrio de *Nash* é calculado através da repetição do comportamento de melhor resposta: em todas as iterações, uma fração dos agentes recalcula uma rota ou um destino com base no que seria melhor na iteração anterior, assumindo que o comportamento de todos os outros agentes permaneça inalterado. Após esse replanejamento, os planos resultantes de todos os agentes são executados simultaneamente na simulação de mobilidade e novas medidas de desempenho são computadas. Este processo é repetido várias vezes. Uma vez estabilizado, nenhum agente pode melhorar substancialmente através de um replanejamento de rota ou destino e um equilíbrio aproximado de *Nash* é obtido. Com o destaque para simulações de evacuação, o equilíbrio de *Nash* faz com que tempos mais curtos de evacuação sejam alcançados, onde toda a população tenta encontrar uma rota de evacuação ideal por meio do aprendizado iterativo.

2.7 Via

O Via é um *software* independente que possibilita a visualização e análise de dados do MATSim, bem como conjuntos de dados espaciais e temporais genéricos. Os dados visualizados no Via são estruturados em camadas. Cada camada é responsável por visualizar um conjunto de dados. Assim, ele fornece camadas para visualizar redes, instalações, atividades e viagens de agentes, paradas de transporte público ou apenas pontos genéricos no espaço, como uma coordenada qualquer.

3 Trabalhos relacionados

Nesta seção são apresentados trabalhos relevantes e relacionados a este projeto ordenados cronologicamente. Como fator de comparação foram utilizadas as seguintes métricas: objetivos, simulação, ferramenta utilizada caso tenha sido feita simulação, recomendação de abrigos, se leva em consideração lotação dos abrigos e controle de congestionamentos. A comparação é feita na Tabela 1.

3.1 Trabalhos selecionados

Há várias maneiras de implementar um método de recomendação de rotas. Assim, serão citados trabalhos referentes à métodos de recomendação de rotas com foco em cenários de desastres naturais.

Com o propósito de abordar o uso de algoritmos evolutivos multiobjetivo (MOEA) e o sistema de informações geográficas (GIS) para o planejamento de evacuação, [Saadatseresht e Taleai \(2009\)](#) propuseram uma abordagem em três etapas para o planejamento de evacuação. No primeiro passo, locais seguros adequados para evacuação são selecionados/-projetados. Estes locais foram adotados do plano abrangente para a gestão de desastres de Teerã projetado pela *Japan International Cooperation Agency (JICA)* e a *Tehran Disaster Mitigation and Management Center (TDMMC)*. Na segunda etapa, as áreas seguras candidatas apropriadas são escolhidas para cada bloco de construção. Esta seleção é baseada na proximidade das áreas seguras disponíveis aos blocos. Na terceira etapa, é feita uma distribuição ideal das pessoas considerando-se capacidade de estradas e distância para as áreas seguras. Seguindo essas três etapas, a área segura adequada e o caminho ideal para entrar na área segura são determinados para cada bloco de construção.

No artigo de [Stepanov e Smith \(2008\)](#), o projeto e a análise ótima das rotas de evacuação em redes de transporte são examinados. Uma metodologia para projetar políticas de roteamento ótimas para o planejamento de evacuação de emergência (EEP) é sugerida. Uma formulação do modelo de programa inteiro (IP) foi apresentada para gerar uma atribuição ótima de rota para um EEP estocástico. Este modelo de IP utiliza modelos de fila M/G/c/c de taxa de serviço de decaimento dependente do estado para capturar funções de atrasos de tempo em ligações rodoviárias. As medidas de desempenho para a política de evacuação gerada, como o tempo de liberação, a distância percorrida total e o nível de congestionamento, são avaliadas com o *software* de simulação MGCCSimul. Embora a política derivada com o modelo EEP IP leve a uma maior distância total percorrida, ela minimiza simultaneamente o nível de bloqueio e congestionamento, bem como o tempo de liberação. Em comparação com a política de IP original, a política de IP do EEP re-

sulta em uma redução de 22% no tempo de liberação. O modelo EEP IP sugerido fornece solução analítica rápida e requer apenas uma avaliação com o programa de simulação.

O artigo de [Lämmel e Flötteröd \(2009\)](#) compara duas estratégias de roteamento diferentes em uma simulação com vários agentes de um grande cenário de evacuação no mundo real. A primeira abordagem aproxima o equilíbrio de Nash, onde cada evacuado adota uma estratégia de roteamento ideal individualmente, independentemente do que essa solução impõe aos outros. A segunda abordagem minimiza aproximadamente o tempo total de viagem no sistema, o que exige impor o comportamento cooperativo dos evacuados. Ambas as abordagens são analisadas em termos da dinâmica de evacuação global e em um nível geográfico detalhado. Os resultados deste artigo mostram que soluções de roteamento cooperativo matematicamente motivadas podem ser obtidas com uma sobrecarga computacional aceitável, mesmo em um sistema puramente baseado em simulação. A abordagem de roteamento cooperativo apresentada, que é uma aproximação do "roteamento ótimo do sistema" bem conhecido no campo da atribuição dinâmica de tráfego, gera uma taxa de evacuação substancialmente mais alta do que uma estratégia de roteamento não-cooperativa alternativa.

[Coutinho-Rodrigues e Alçada-Almeida \(2012\)](#) propuseram um problema de otimização multiobjetivo para encontrar caminhos de evacuação e a localização de abrigos para planejamento de evacuação urbana. Os autores consideraram muitos objetivos para otimização, como comprimentos de caminho, riscos de caminho, tempos de evacuação, comprimentos de caminhos de *backup* e número de abrigos. O conjunto de rotas primárias entre o local do desastre e os locais do abrigo foi gerado com uma abordagem bi-objetiva do caminho mais curto considerando os tamanhos dos caminhos e os riscos do caminho. Seu algoritmo para gerar os pares de caminhos primários e de *backup* para cada setor foi implementado no ambiente GIS. O modelo foi testado simulando uma situação de incêndio no centro da cidade (Baixa) de Coimbra, Portugal, uma área com muitas ruas estreitas, poucas áreas abertas e edifícios muito antigos. Dez soluções foram geradas e comparadas usando tabelas, gráficos e mapas codificados por cores em um ambiente GIS. A solução ideal foi determinada minimizando cada objetivo individualmente. Soluções adicionais foram obtidas usando o método de ponderação, minimizando a distância para a solução ideal e usando um procedimento hierárquico.

Com o objetivo de avaliar e otimizar a eficiência de uso espaço-temporal de um plano de evacuação, [Fang Qiuping Li e Shaw \(2013\)](#) propõem uma métrica de avaliação e a utilizam para a otimização de planos de evacuação. Primeiro, foi definida uma métrica de eficiência de uso espaço-tempo com base em trajetórias em segmentos de estradas e interseções. Em segundo lugar, foi proposto um algoritmo híbrido de otimização multiobjetivo de duas camadas para planejar as direções do movimento de conversão de veículos e pedestres em uma rede integrada de rodovias e construção de interiores com a finalidade de tomar decisões sobre planos de evacuação. Foram analisados os resultados otimizados de três

perspectivas diferentes, ou seja, soluções de Pareto, curvas de evacuação e eficiências de uso espaço-tempo em interseções e *links*. Os experimentos computacionais deste estudo demonstram que as interseções em uma rede de evacuação precisam ser atribuídas a direções razoáveis de movimento de rotação de veículos e pedestres. Uma comparação entre as soluções propostas calculadas e uma solução TRANSIMS (Sistema de Simulação de Análise de Transportação) mostra que a abordagem proposta é adequada e eficaz para organizar veículos e pedestres no espaço e no tempo em um processo de evacuação.

No artigo de [Kim e Lee \(2017\)](#), uma rede de evacuação é projetada para desenvolver um plano de evacuação com base em um problema hierárquico de design de rede usado em redes de comunicação, transporte público e projetos de redes de organizações sociais. Análises experimentais foram realizadas para encontrar planos de evacuação eficazes em *Haeundae Beach*, Coréia, usando simulação de transporte de agentes múltiplos (MAT-Sim). Para projetar uma rede de evacuação hierárquica eficaz, um projeto de rede de evacuação hierárquica é desenvolvido usando os conceitos de localização do *hub*, *cluster* e design de rede. Além disso, é considerado o contrafluxo na rede de evacuação, bem como a determinação das linhas do corredor pela regra de distância mínima, regra de fluxo máximo, regra de acesso máximo e regra de tempo de viagem mínimo. Como resultado da execução de cada cenário usando o MATSim, é avaliado um critério para estabelecer a linha do corredor. Uma rota de evacuação ideal é encontrada para minimizar as filas superlotadas de evacuados, quando o tempo total de evacuação da área do tsunami é de aproximadamente 3 horas e 30 minutos.

[Benchekroun e Mouraouah \(2018\)](#) propõem um plano de evacuação de tsunami para a cidade de Tânger-Marrocos para evacuação horizontal (quando os evacuados se deslocam das zonas de risco para áreas mais seguras ou áreas mais altas (colinas) localizadas fora da área de inundação). Este plano foi elaborado considerando a ameaça de tsunami de fonte tsunamigênica localizada na margem da Península Ibérica e usando os mapas de inundação do pior cenário para definir a área de inundação. O plano de evacuação é elaborado por meio da modelagem de rotas de evacuação usando a ferramenta *Closet Facility* implementada no software ArcMap 10.2 do sistema de informações geográficas (GIS) do ESRIS. A modelagem de evacuação foi aplicada para encontrar as rotas mais eficazes, levando em consideração o menor tempo de viagem para evacuar as zonas de risco em caso de tsunami durante a baixa temporada. A modelagem foi alcançada usando dados de entrada detalhados da rede viária, abrigos de construção de evacuação ou áreas de montagem e distribuição espacial da população na zona de risco. Os resultados são apresentados em um plano de evacuação em que as rotas de fuga, locais seguros e pontos de montagem estão claramente marcados. Durante a resolução da análise, foram escolhidos atributos de restrição, incluindo limites de capacidade dos caminhos ou altos graus da encosta, para proibir alguns evacuados de atravessar determinadas estradas, a fim de evitar congestionamentos de pedestres.

Veeraswamy Edwin R. Galea e Smith (2018) estenderam o modelo de evacuação *buildingEXODUS* para modelar evacuações urbanas em larga escala, incluindo a rede de estradas e espaços abertos (por exemplo, parques, espaços verdes e praças) junto com os edifícios. Os resultados da simulação de evacuação foram acoplados aos resultados de um modelo de propagação de incêndios florestais e aplicados ao incêndio florestal de *Swinley* que ocorreu em *Berkshire*, Reino Unido em maio de 2011. Foram avaliados quatro procedimentos de evacuação diferentes nas rotas percorridas pelos pedestres, fornecendo estatísticas importantes sobre a evacuação como tempo para chegar ao local da montagem, a distância percorrida, o congestionamento experimentado pelos agentes e as margens de segurança associadas ao uso de cada rota de evacuação. Uma das principais conclusões deste trabalho é a importância de formular procedimentos de evacuação que identifiquem a população ameaçada, forneça um aviso de evacuação oportuno, identifique rotas apropriadas que mantêm uma distância segura da frente de perigo, maximizando assim as margens de segurança mesmo com o custo de seguir rotas de evacuação mais longas. A simulação de evacuação oferece um meio de atingir esses objetivos.

No trabalho de Lujak e Giordani (2018) foi estudado a agilidade das rotas de evacuação em relação à mudança dinâmica de condições perigosas imprevisíveis em redes espaciais inteligentes. Com isso, eles propuseram duas novas métricas de importância do nó: evacuação entre centralidade de evacuação e centralidade de evacuação, ambas inspiradas na centralidade de evacuação. A primeira métrica representa a fração de rotas de evacuação eficientes entre todos os pares de origem-destino diferentes do nó especificado que passa por esse nó, enquanto a segunda representa a importância do nó especificado para a evacuação, considerando a disponibilidade de caminhos alternativos de evacuação eficientes (rotas) desse nó para saídas seguras. Além disso, dado um conjunto de posições e saídas seguras dos evacuados, eles encontraram rotas de evacuação ágeis mais curtas, onde, nesse trabalho, rota ágil representa a capacidade de redirecionar de forma eficiente e segura dos nós intermediários no caso de quedas imprevisíveis de segurança, maximizando o valor da centralidade da evacuação dos nós intermediários da rota. Além disso, propuseram um algoritmo para esse problema e discutiram sua capacidade de reagir às mudanças nas circunstâncias de segurança ao longo das rotas recomendadas.

Tabela 1: Trabalhos relacionados.

| Trabalhos Relacionados | Objetivos | Simulação? | Sim, qual a ferramenta? | Recomenda abrigos? | Considera lotação dos abrigos? | Controle de congestionamento |
|--|---|------------|-------------------------|--------------------|--------------------------------|------------------------------|
| Stepanov e Smith (2008) | Projetar e analisar algoritmos de atribuição de rotas | Sim | MGCCSimul | Não | Não | Sim |
| Saadatseresht e Taleai (2009) | Abordar o uso de algoritmos para o planejamento de evacuação | Não | - | Não | Não | Não |
| Lämmel e Flötteröd (2009) | Comparar estratégias de roteamento em simulação de grande cenário de evacuação | Sim | MATSim | Não | Não | Sim |
| Coutinho-Rodrigues e Alçada-Almeida (2012) | Identificar caminhos de evacuação e a localização de abrigos | Não | - | Sim | Não | Não |
| Fang Qiuping Li e Shaw (2013) | Fornecer método prático de medir eficácia de planos de evacuação | Sim | FIFO e TRANSIMS | Não | Não | Não |
| Kim e Lee (2017) | Projetar rede de evacuação para desenvolver plano de evacuação com base em problema de design de rede hierárquico | Sim | MATSim | Não | Não | Sim |
| Benchekroum e Mouraouah (2018) | Propor plano de evacuação de tsunami para a cidade de Tânger-Marrocos para evacuação horizontal | Não | - | Sim | Não | Sim |
| Veeraswamy Edwin R. Galea e Smith (2018) | Estender modelo de evacuação <i>buildingEXODUS</i> para modelar evacuações urbanas em larga escala | Sim | <i>Prometheus</i> | Sim | Não | Sim |
| Lujak e Giordani (2018) | Fornecer método prático de medir eficácia de planos de evacuação. | Não | - | Não | Não | Sim |
| Este projeto | Recomendar rotas e abrigos disponíveis, reduzir tempo e minimizar congestionamentos | Sim | MATSim | Sim | Sim | Sim |

A metodologia adotada neste trabalho difere dos trabalhos selecionados, fornecendo a recomendação de rotas para evacuados considerando-se a distância e disponibilidade de vagas em abrigos, a disponibilidade de vias no cenário e a redução de congestionamentos. Além disso, a simulação de evacuação do tráfego de veículos é realizada utilizando o simulador MATSim em um estudo de caso real. Essas contribuições terão então os seus desempenhos analisados e o impacto no tempo de evacuação total avaliado.

4 Proposta

Este trabalho consiste em desenvolver uma metodologia para a recomendação de rotas junto à simulação de um cenário de desastre com o software MATSim. Simultaneamente, para ilustrar a aplicação da metodologia desenvolvida, é feita sua aplicação na simulação da área do desastre *Camp Fire, Butte County (CA)*, que aconteceu em 2018. Em ordem, apresentaremos o cenário de desastre utilizado como estudo de caso, discutiremos o método proposto e, por fim, a simulação é comentada.

4.1 Cenário de desastre

No dia 08 de novembro de 2018 ocorreu o incêndio mais destrutivo na história da Califórnia, conhecido como *Camp Fire*. Este foi o desastre natural mais caro em 2018 no mundo em termos de perdas seguradas. Em janeiro de 2019, o dano total foi estimado em US \$ 16,5 bilhões; um quarto dos danos não estava segurado, [LosAngelesTime \(2019\)](#). Além de perdas financeiras, o incêndio causou a morte de 85 civis e feriu outros 12 civis e 5 bombeiros, [FOX40 \(2019\)](#). Com isso, o *Camp Fire* também é o incêndio florestal mais mortal nos Estados Unidos desde o incêndio de Cloquet em 1918 e está no topo da lista dos incêndios florestais mais mortais do mundo.

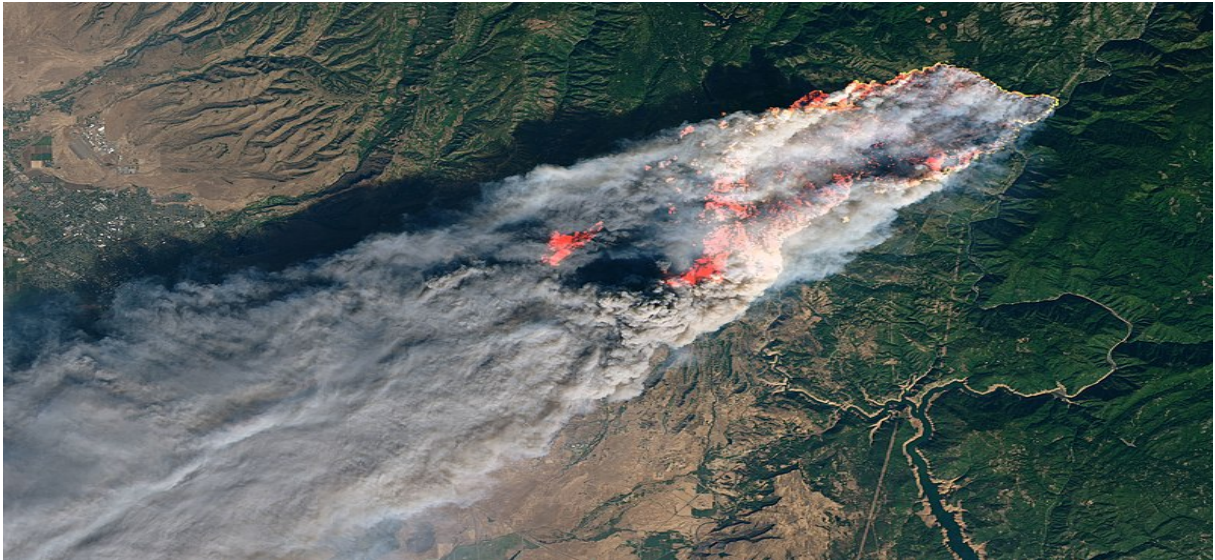


Figura 5: Incêndio *Camp Fire* visto do satélite Landsat 8, em 08 de novembro de 2018. Escala: 8 km por cm. Fonte: [SFGATE \(2018\)](#).



Figura 6: Extensão da fumaça provocada pelo *Camp Fire*. Fonte: [SFGATE \(2018\)](#).

Este incêndio teve uma rápida propagação, que teve como força motriz os ventos extremos que assolaram a região. Com rajadas de até 100 quilômetros por hora, estes ventos empurraram o fogo com uma velocidade rápida para o sul da Califórnia, [WIRED \(2018\)](#). Rajadas fortes também direcionaram faíscas e fumaça para as comunidades a favor do vento. Esse foi um dos problemas para extinguir o *Camp Fire*, que estava situado de modo que os ventos vindos do Nordeste soprassem a fumaça para a área da baía, a 160 quilômetros de distância.

Além da destruição e mortes causadas, essa fumaça atingiu uma grande área, como visto nas Figuras 5 e 6. A fumaça resultou em poluição do ar difundido em toda a Área da Baía de São Francisco e Vale Central, o que levou ao fechamento de escolas públicas em cinco municípios da área da baía e dezenas de bairros da área metropolitana de Sacramento em 16 de novembro. A fumaça era visível em lugares distantes como a cidade de Nova York, depois que nuvens de fumaça percorreram uma distância de mais de 3.000 milhas

(aproximadamente 4.828,032 quilômetros). Além disso, este incêndio cobriu uma área de 153.336 acres, como mostrado na Figura 7 e destruiu 18.804 estruturas, com a maior parte dos danos ocorrendo nas primeiras quatro horas.

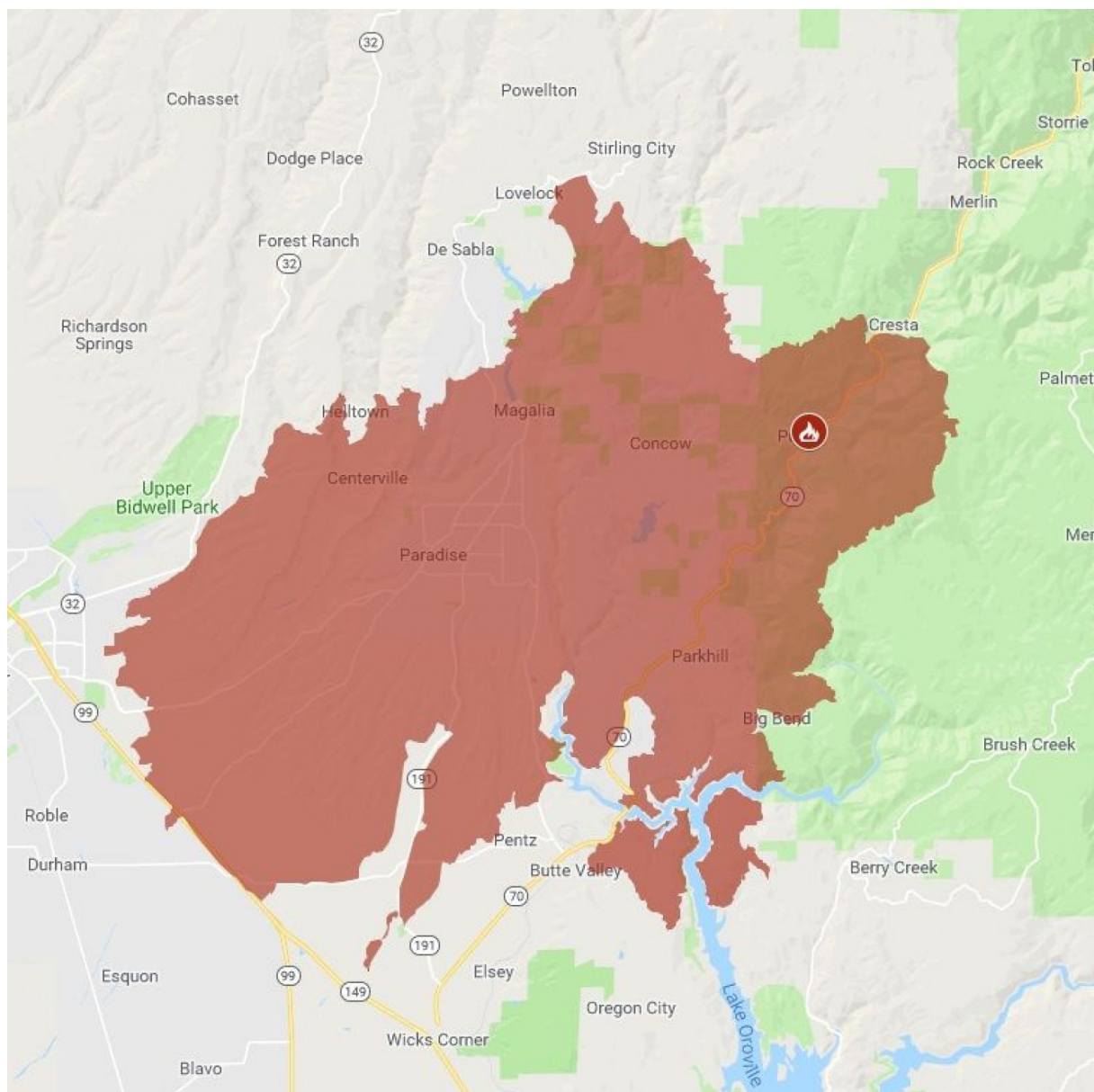


Figura 7: Área queimada pelo incêndio *Camp Fire*. Escala: 2 km por cm. Fonte: [Newsweek](#) (2018).

4.2 Método

A Figura 8 descreve a metodologia seguida para alcançar os resultados obtidos. Nas subseções abaixo é explicado o passo a passo dessa metodologia.

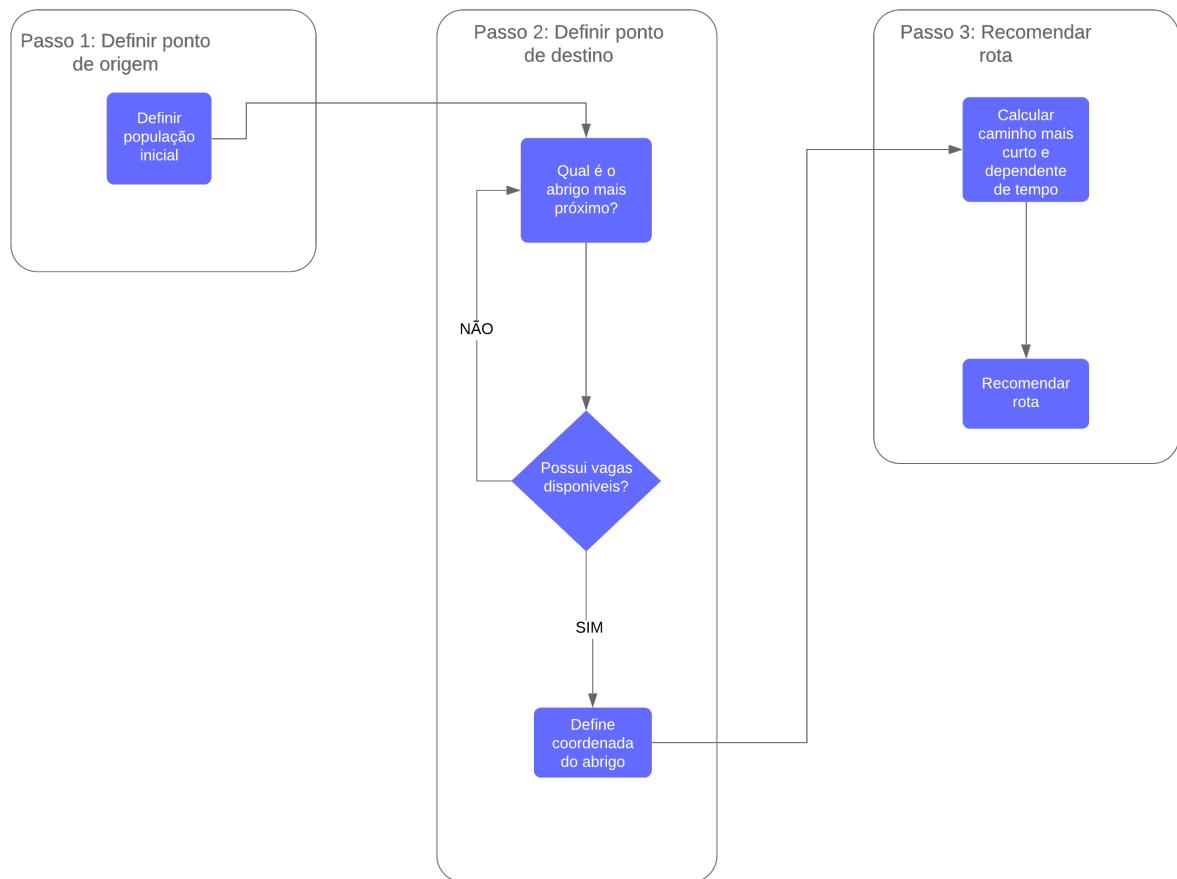


Figura 8: Metodologia de recomendação de rotas.

4.2.1 Definição do ponto de origem

Como discutido anteriormente, o fogo se espalhou rapidamente na região do Condado de Butte. Isso causou grandes danos a cabos de comunicação e torres de celular¹, fazendo com que algumas localidades não tivessem avisos de evacuação. Porém, as cidades de Pulga e Paradise foram as únicas que receberam os avisos, tendo suas ordens de evacuação, via *Twitter*, às 07:23² e 08:00³, respectivamente. Como a área afetada era extensa e o aumento de pessoas evacuando congestionaria as estradas e diminuiria o tráfego, os horários de evacuação foram diferentes. Portanto, houve a necessidade de fazer um cronograma de horas em que os residentes deveriam evacuar. Este cronograma foi então uma contribuição para este trabalho, já que em trabalhos anteriores os evacuados saíam no mesmo horário,

¹ **Ahead of Camp Fire Anniversary, New Details Emerge of Troubled Evacuation**; Frontline. Disponível em <https://to.pbs.org/2XyxAR9>. Acesso em 20 de novembro de 2019.

² **Camp Fire Timeline of Terror: The evacuation of Butte County's Paradise from beginning to end**; ABC7News. Disponível em: <https://abc7ne.ws/20yK1rW>. Acesso em 20 de novembro de 2019.

³ **As California's deadliest wildfire closed in, evacuation orders were slow to arrive**; Los Angeles Times. Disponível em: <https://lat.ms/2s3B1Uf>. Acesso em 20 de novembro de 2019.

como, por exemplo, [Lämmel e Nagel \(2008\)](#), [Bakillah Hubert Klüpfel e GeorgWalenciak \(2012\)](#) e [Benchekroun e Mouraouah \(2018\)](#).

Assim, levando em consideração estes fatores, a área de evacuação foi dividida em quatro setores: o primeiro com Pulga e Starling City; o segundo com a região de Magalia e Concow, o terceiro com a área de Paradise, Centerville, Helltown e Parkhill e o quarto com as localidades próximas a Chico, como mostrado na Figura 9. Em relação a definição do horário de evacuação dessas áreas, foi considerado o tempo que o incêndio levou para chegar em Paradise após a ordem de evacuação de Pulga, ou seja, essas ordens de evacuação foram utilizadas como base para evacuar os outros dois setores. Logo, os horários de evacuação ficaram estabelecidos como 07:23, 07:42, 08:00 e 08:19, em ordem.



Figura 9: Setores do cronograma de evacuação. Adaptado de [Times \(2018\)](#).

Após feito o cronograma, os pontos de origem dos evacuados foram definidos. Esta definição se deu por meio da obtenção de coordenadas aleatórias de cada setor e seus respectivos horários através da geração de um código. Estes pontos foram apresentados na Figura 10 a seguir.

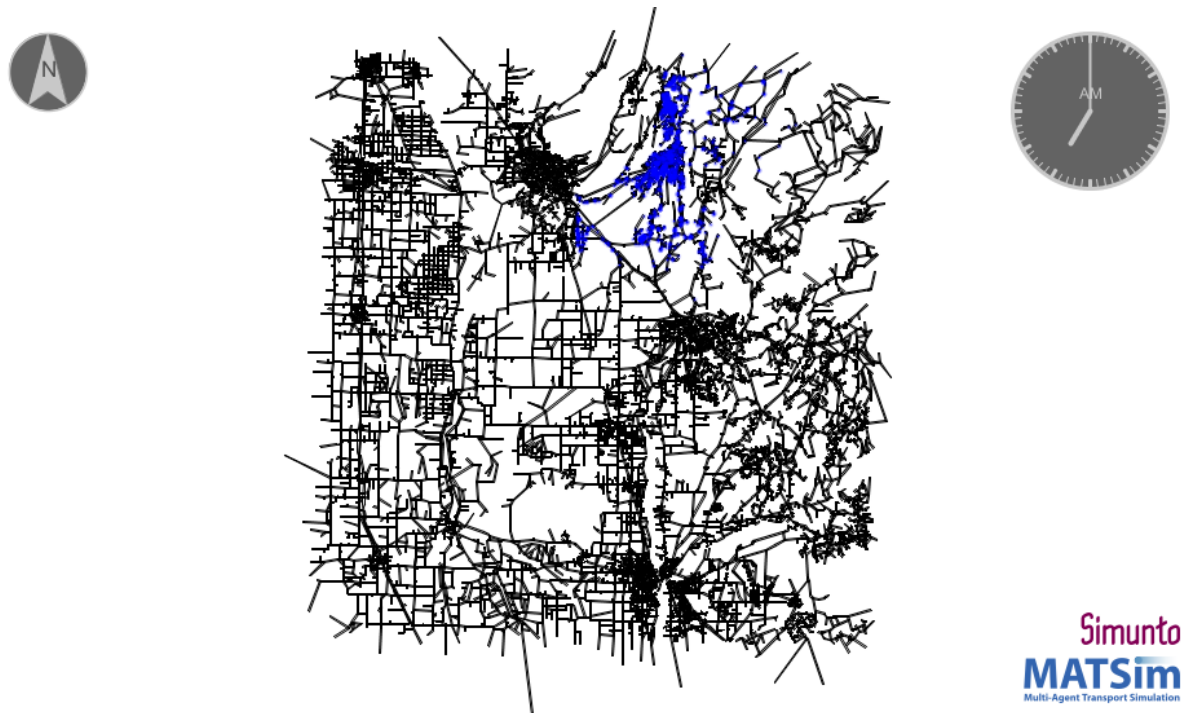


Figura 10: Pontos de origem dos evacuados.

4.2.2 Definição do ponto de destino

Inicialmente, para a definição do ponto de destino, nós realizamos um levantamento dos abrigos disponíveis próximos ao desastre e suas respectivas lotações. Assim, cinco abrigos foram selecionados⁴, como apresentado na Tabela 2. Em seguida, suas coordenadas foram obtidas através do *Google Maps* e transformadas do Sistema Geodésico Mundial (WGS 84), norma usada pelo Sistema de Posicionamento Global (GPS), para o código criado pelo Grupo de Pesquisa Petrolífera Europeia (European Petroleum Survey Group), o código EPSG. Essa transformação se dá devido ao fato do simulador não suportar dados geográficos no sistema WGS 84.

Tabela 2: Abrigos selecionados como destinos de evacuação.

| Abrigos | Localização | Capacidade |
|---------------------------|-------------|------------|
| East Avenue Church | Chico | 250 |
| Silver County Fairgrounds | Chico | 620 |
| Glenn County Fairgrounds | Orland | 450 |
| Yuba Sutter Fairgrounds | Yuba City | 380 |
| Butte County Fairgrounds | Gridley | 100 |

Após isso, os nós mais próximos a essas coordenadas são escolhidos e assim é criada uma lista com os nós referentes as localizações dos abrigos, conforme é mostrado na Figura 11. Subsequentemente, é feito um cálculo utilizando o algoritmo de *Dijkstra* que recebe

⁴ **Camp Fire Resources**; Far Northern Regional Center. Disponível em: <http://bit.ly/2D2qRp9>. Acesso em 20 de novembro de 2019.

como parâmetros a lista de nós e o nó da primeira atividade dos agentes. Com isso, são retornados os menores caminhos entre o ponto de origem do evacuado e os abrigos. Assim, é possível criar uma lista com os abrigos mais próximos ao evacuado em ordem crescente. Depois, é checado se há vagas disponíveis no abrigo mais próximo. Se houver, este abrigo será escolhido como destino pro evacuado, se não, o abrigo seguinte na lista será checado. Essa é uma outra contribuição do nosso trabalho, já que normalmente é recomendado somente o abrigo mais próximo sem considerar a lotação, como visto nos trabalhos de [Agafonova e Myasnikova \(2017\)](#) e [Zhu Kun Xie e Yang \(2018\)](#), por exemplo.



Figura 11: Localização de abrigos no mapa da área simulada.

4.2.3 Recomendação de rota

No MATSim, as rotas dos agentes são especificadas por sequências de *links* e são geradas por um algoritmo de caminho mais curto que requer uma função de custo. Esta função de custo pode ser personalizada. No cálculo da rota deste método, que acontece durante a simulação, o tempo de viagem do *link* foi utilizado como o custo e o algoritmo A* foi aplicado como roteador. Na construção dessa rota, foi considerado também o horário de partida da primeira atividade dos agentes e a disponibilidade de vias, já que no cenário real diversas rotas foram fechadas devido as chamadas varrem perigosamente perto das rodovias. Para analisar essa disponibilidade, um algoritmo foi desenvolvido para gerar um arquivo de eventos de mudança de rede, fazendo com que fosse viável a remoção de vias indisponíveis em tempo de execução. O arquivo usado para fazer essa remoção é mostrado no Apêndice C. Essa remoção foi feita de forma dependente do tempo, ou seja,

foram definidos horários em que essas rotas deveriam ser removidas de qualquer sugestão. Com o cronograma de evacuação como base, foram estabelecidos horários diferentes para cada setor, cada um tendo uma hora para evacuar a área. Dessa forma, dependendo do horário, essas rotas não foram utilizadas na recomendação.

Além disso, como forma de recomendar melhores rotas, o equilíbrio de *Nash*, usado nativamente no simulador, foi aplicado. Ele calcula esse equilíbrio repetindo o comportamento de melhor resposta: em todas as iterações, uma fração dos agentes recalcula uma rota ou um destino com base no que seria melhor na iteração anterior, assumindo que o comportamento de todos os outros agentes permaneça inalterado. Após esse replanejamento, os planos resultantes de todos os agentes são executados simultaneamente na simulação de mobilidade e novas medidas de desempenho são computadas. Este processo é repetido várias vezes, de acordo com o número de iterações definidas no cenário, neste caso, 5 iterações. Uma vez estabilizado, nenhum agente pode melhorar substancialmente através de um replanejamento de rota ou destino e um equilíbrio aproximado de *Nash* é obtido.

Nesse módulo, outro fator que poderia influenciar na recomendação de rotas foi a minimização de congestionamentos na rede rodoviária. Esta minimização foi feita através da detecção de congestionamentos e, em seguida, a execução do algoritmo de roteamento novamente para recomendar rotas alternativas. Ou seja, rotas congestionadas se tornaram menos atraentes, promovendo a busca por outras rotas. Com isso, decisões de viagem individuais dos agentes são alteradas durante as iterações. Esta implementação faz também com que a simulação passe do Equilíbrio de *Nash* para o Sistema Ideal, onde rotas mais rápidas são recomendadas, mas não são estáveis (rotas com maior risco de impactar negativamente no tempo de evacuação). Depois que o descongestionamento foi minimizado, os usuários de transporte não têm mais nenhum incentivo para se comportar de maneira ideal e retornam ao comportamento normal. Em consequência, o comportamento do deslocamento volta do Sistema Ideal para o equilíbrio de *Nash*. Os algoritmos usados para realização desta tarefa são mostrados nos Apêndices D e E.

4.3 Simulação

Após a descrição da metodologia, nesta seção, descrevemos como ocorre a criação dos arquivos necessários à simulação. Todos os códigos estão disponíveis em um repositório⁵ no *GitHub*.

⁵ Link do repositório. <https://github.com/OrranaLhaynher/matsim-example-project>

4.3.1 Criação da rede de simulação

Como discutido anteriormente, um dos arquivos principais para a realização de uma simulação no MATSim é o arquivo de *network*. O método mais comum para a sua geração é através da exportação de dados do OpenStreetMap (OSM). Existem diversos serviços da *Web* que possibilitam a extração de dados para uma área de interesse, tais como: *GeoFabrik*, *BBBike* e *Overpass Turbo*. Além destes, existem outros *softwares* livres, como o *Quantum GIS* e *JOSM*, que têm *plugins* que são utilizados para este fim. Contudo, a forma mais comum é através do próprio site do *OpenStreetMap*, onde é possível buscar pelo mapa ou região de interesse e exportá-lo, conforme ilustrado na Figura 12. Com a exportação, um arquivo é gerado. Este arquivo é utilizado para salvar informações de mapas de ruas na forma de "nós"(pontos), "caminhos"(conexões) e "relações"(propriedades de ruas e objetos, como *tags*).

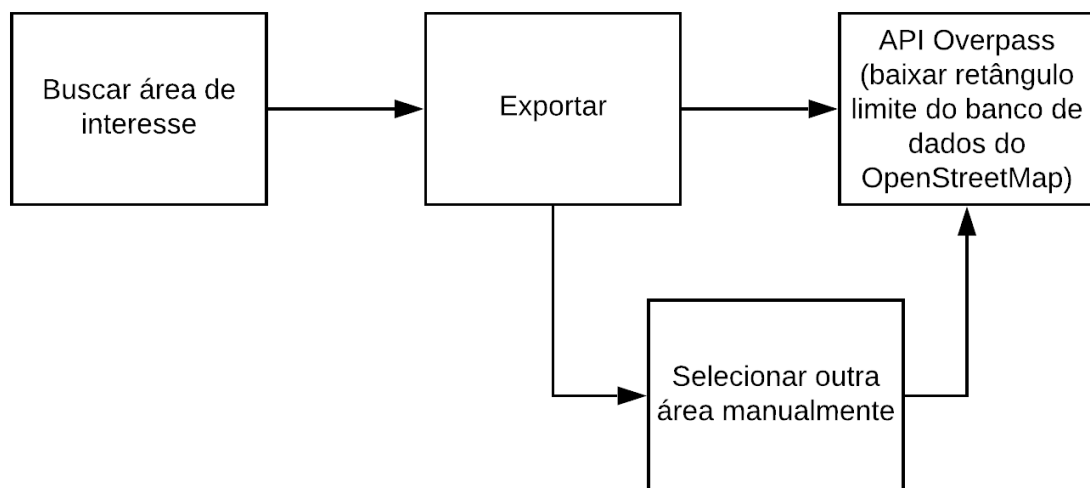


Figura 12: Fluxo de extração de dados do *OpenStreetMap*.

Para a criação do cenário deste trabalho, foram exportados cinco arquivos: um arquivo com toda a rede rodoviária utilizada na simulação e outros quatro, correspondentes aos setores comentados anteriormente. Após isso, foi desenvolvido um código para a conversão de dados OSM em arquivos compatíveis com o simulador MATSim, resultando na rede mostrada na Figura 13. O algoritmo é responsável por criar também os arquivos SHP que podem ser utilizados para a visualização do cenário abordado, conforme apresentado no Apêndice A.

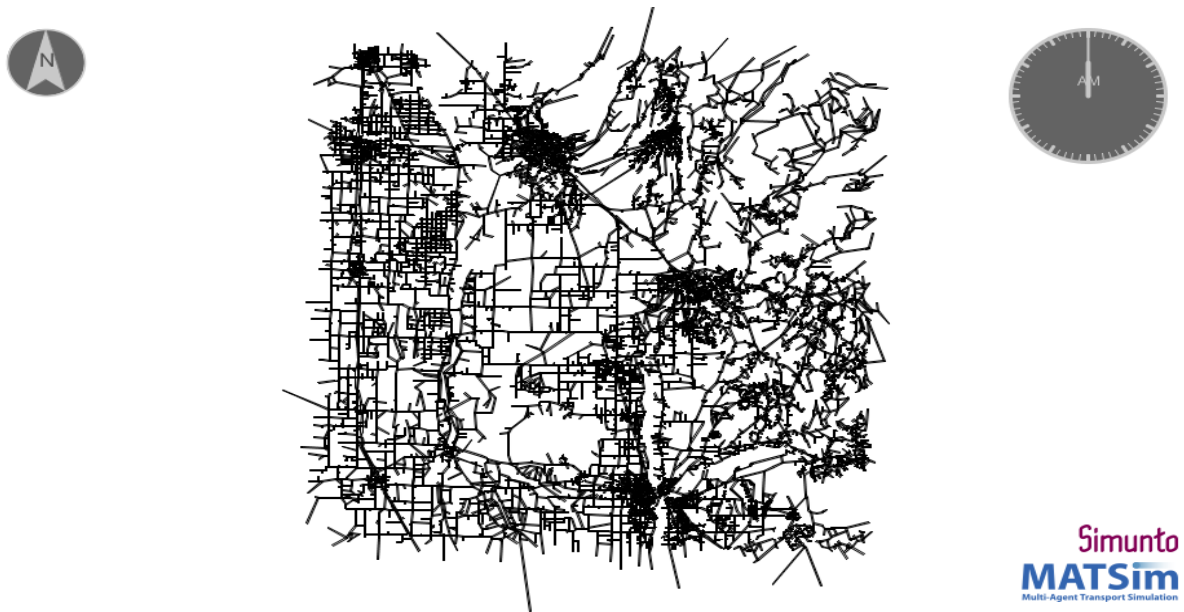


Figura 13: Rede rodoviária MATSim após a conversão.

4.3.2 Criação da população MATSim

Para que o ciclo MATSim seja realizado é necessária uma demanda inicial, como mostrado na Figura 4. Existem comumente duas possibilidades para a geração dessa demanda: a forma prática e a exigente. A forma mais prática é usar um censo populacional completo na criação da demanda, enquanto que a exigente é gerar uma população sintética, que pode ser baseada em pesquisas de amostra ou estrutura.

Neste estudo de caso foi criada uma população sintética de 1800 agentes que foi dividida em 4 setores da área de evacuação, como comentado anteriormente. Então, cada agente teve uma atividade *home* (casa) formulada utilizando os dados de latitude e longitude definidos conforme as coordenadas do setor em que este foi designado, como mostrado na Figura 10. A atividade *shelter* (abrigo) foi definida a partir da seleção de qual abrigo seria enviado o evacuado. O algoritmo contendo toda a implementação deste tópico é apresentado no Apêndice B.

4.3.3 Criação do arquivo de configuração

A criação do arquivo de configuração consistiu na alteração dos parâmetros de cada módulo do arquivo padrão gerado diretamente pelo simulador. Nesse estudo de caso, optou-se por modificar apenas 7 dos 25 módulos que compõem o arquivo de configuração padrão. No arquivo *config* utilizado como dado de entrada no software, só é preciso registrar os módulos que sofreram alteração. O MATSim assume os parâmetros padrão para os módulos que não são registrados no arquivo de configuração. Os tópicos a seguir

detalham as configurações adotadas para a simulação deste estudo de caso. O arquivo de configuração usado neste trabalho é apresentado no Apêndice F.

1. Modulo *Global* - define o sistema de coordenadas.
2. Modulo *Network* - define o diretório do arquivo de rede.
3. Modulo *Plans* - define o diretório do arquivo de população.
4. Modulo *Controller* - o principal parâmetro definido nesse módulo é o número de iterações feitas na simulação, nesse caso adotado como 5. Outro parâmetro definido é a frequência com que os arquivos de saída são gerados (definido como 1 para este estudo) e o formato do *snapshot*.
5. Modulo *Qsim* - é definido o número máximo de horas do dia contidas em cada simulação. Isso é importante para evitar que o programa faça iterações infinitas caso haja algum erro de configuração.
6. Modulo *PlanCalcScore* - define os parâmetros da função de utilidade usada para calcular os scores de cada plano e atividade. Nesse estudo, foram mantidas as configurações gerais de pontuação do MATSim, e introduzidas apenas as características de cada tipo de atividade considerada.
7. Modulo *Strategy* - estabelece o que muda para os agentes entre cada iteração simulada. Um parâmetro importante nesse modulo é a quantidade de planos que pode haver na memória do agente.

5 Resultados

Neste capítulo, são apresentados os resultados da simulação feita no software MATSim utilizando os arquivos gerados pela metodologia descrita no capítulo anterior.

5.1 Visualização dos eventos de simulação

Após a importação do arquivo de configuração do Apêndice F no *MATSim*, foram criados arquivos de *output*, possibilitando a visualização no *Via*. Três simulações foram planejadas para avaliar a metodologia de recomendação de rotas: (1) utilizando o cronograma de evacuação e as configurações nativas do simulador, (2) utilizando as modificações propostas sem o módulo de descongestionamentos e (3) com as modificações propostas. Além disso, arquivos de cada iteração foram gerados para um melhor entendimento do que acontece durante a simulação.

No primeiro cenário, foram simulados 1800 agentes, que tinham seus pontos de destinos escolhidos de forma randômica entre os cinco abrigos disponíveis. A área do desastre não foi removida durante a execução, fazendo com que rotas naquela área fosse recomendadas. Por isso, mesmo após o tempo seguro de utilização dessas rotas, agentes continuavam evacuando por elas. Notou-se também que os congestionamentos experienciados nessa simulação estavam concentrados ou na área de evacuação ou próximo a ela. Isso acontece devido a distribuição aleatória por abrigos.

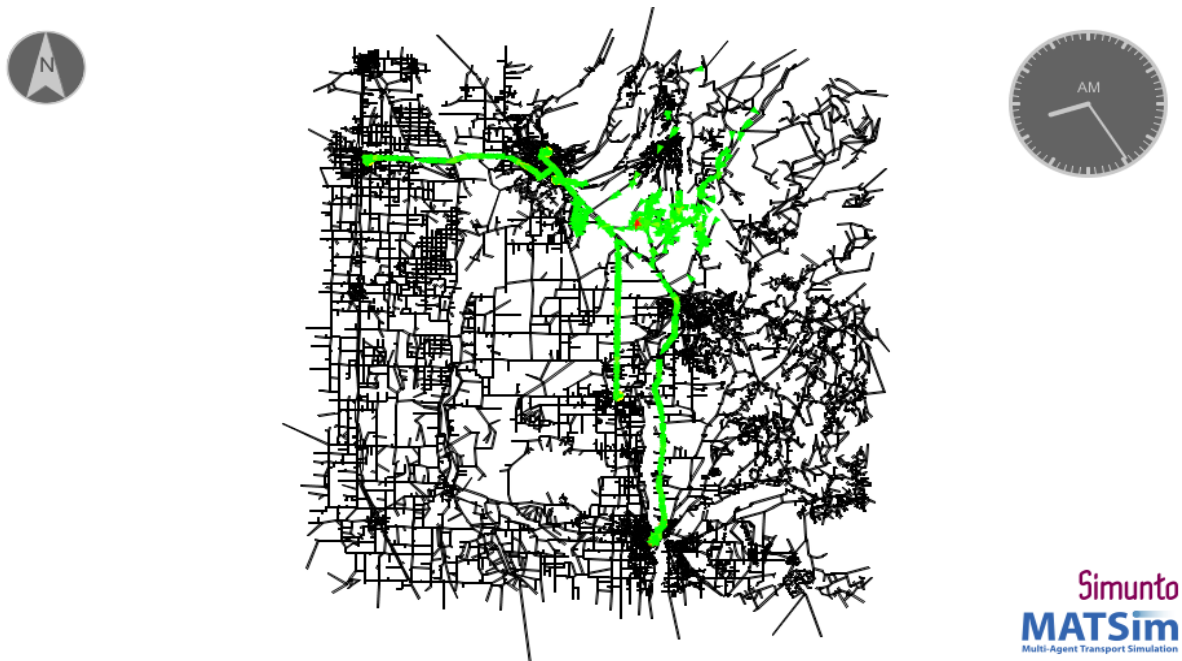


Figura 14: Instante em que toda a população entra na rede rodoviária na simulação 01.

No segundo cenário, foram simulados 1800 agentes, que tinham como pontos de destinos recomendados o abrigo mais próximo e com vagas disponíveis. A área do desastre foi removida durante a execução, fazendo com que rotas naquela área não fossem recomendadas após o horário definido. Desse modo, agentes não entram na região de perigo e acabam tendo rotas recomendadas mais seguras para evacuar. Com relação a congestionamentos, essa simulação foi a que teve o maior número deles. Porém, a maioria dos congestionamentos encontravam-se nas áreas em que os abrigos estavam localizados. Isso ocorre em virtude da rapidez com que os evacuados chegaram aos seus pontos de destino.

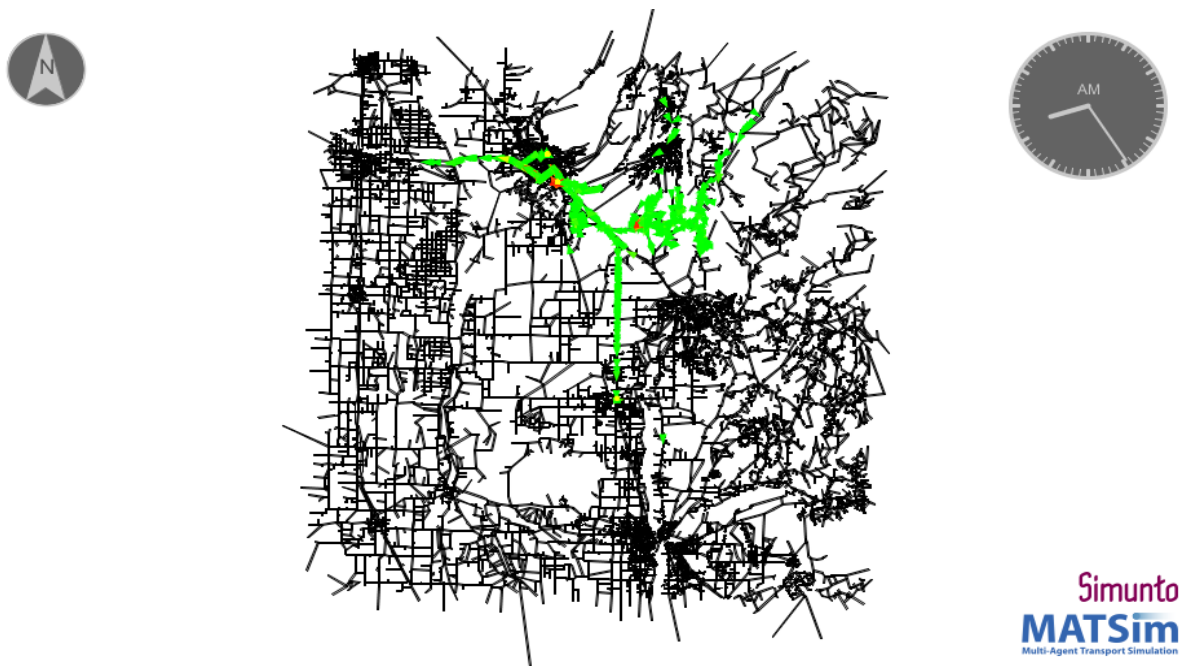


Figura 15: Instante em que toda a população entra na rede rodoviária na simulação 02.

No terceiro cenário, os 1800 agentes, que tinham como pontos de destinos recomendados o abrigo mais próximo e com vagas disponíveis, possuiu outro fator no momento da recomendação da rota: a minimização de congestionamento. A área do desastre também foi removida durante esta execução, fazendo com que rotas nessa área não fossem recomendadas após o horário definido. No tocante dos congestionamentos, essa simulação foi a que teve o melhor desempenho. O número de congestionamentos foi menor que o da segunda simulação e maior do que o a primeira, como mostrado na Tabela 6. Porém, como dito anteriormente, os congestionamentos da primeira simulação estavam principalmente na área de desastre, enquanto que os congestionamentos desta simulação teve um número maior na área dos abrigos.

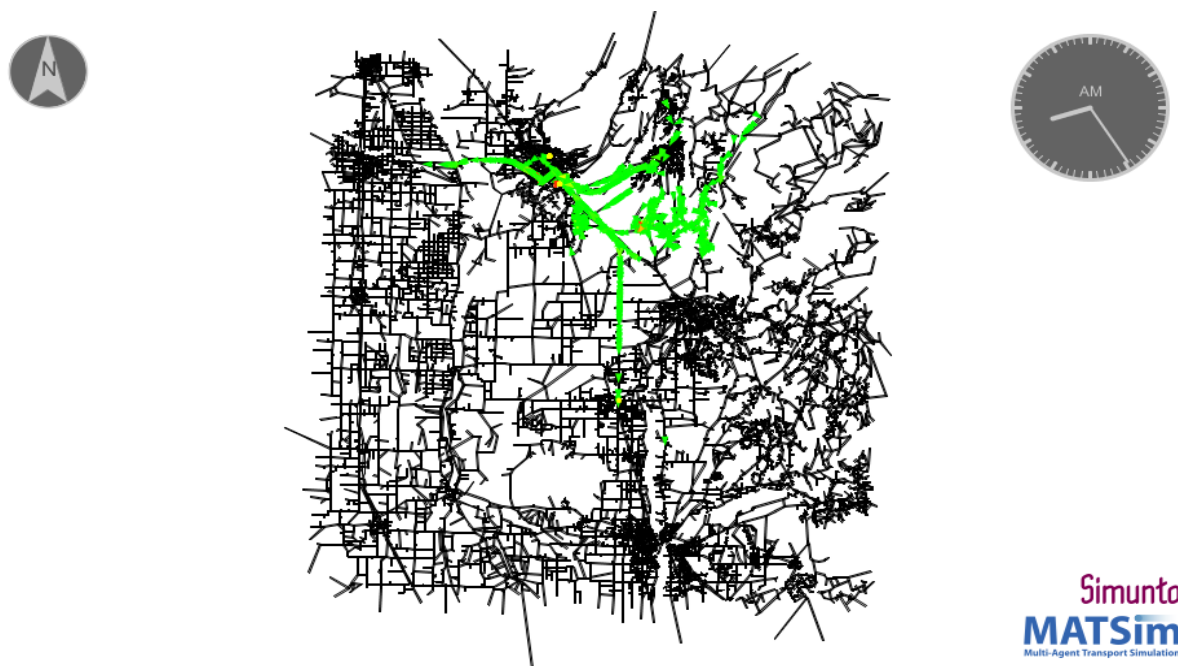


Figura 16: Instante em que toda a população entra na rede rodoviária na simulação 03.

5.2 Análise

Uma análise das simulações foi feita para comprovar que as contribuições impactaram positivamente no tempo de evacuação individual e total do cenário. Na subseção 5.2.1 será mostrada como ficou a distribuição de evacuados nos abrigos. Enquanto que na subseção 5.2.2, foram apresentados uma amostra dos tempo de evacuação dos agentes em um dos abrigos escolhidos. Por fim, na subseção 5.2.3, o tempo de evacuação total foi exibido.

5.2.1 Distribuição de evacuados pelos abrigos

Na simulação com o cenário modelado de forma nativa, os evacuados tiveram o seu ponto de destino definido de forma aleatória, como mostrado na Figura 17. Mesmo com os horários de evacuação iguais, residentes eram distribuídos pelos cinco abrigos. Isso fez com que alguns tivessem um tempo de evacuação razoável, enquanto outros demoraram horas para chegar no seu destino. Na Figura 18, três evacuados que saíram da mesma área estavam sendo encaminhados para três abrigos diferentes. Seus tempos de evacuação em minutos foram respectivamente 01:07:82, 57:25 e 45:40.

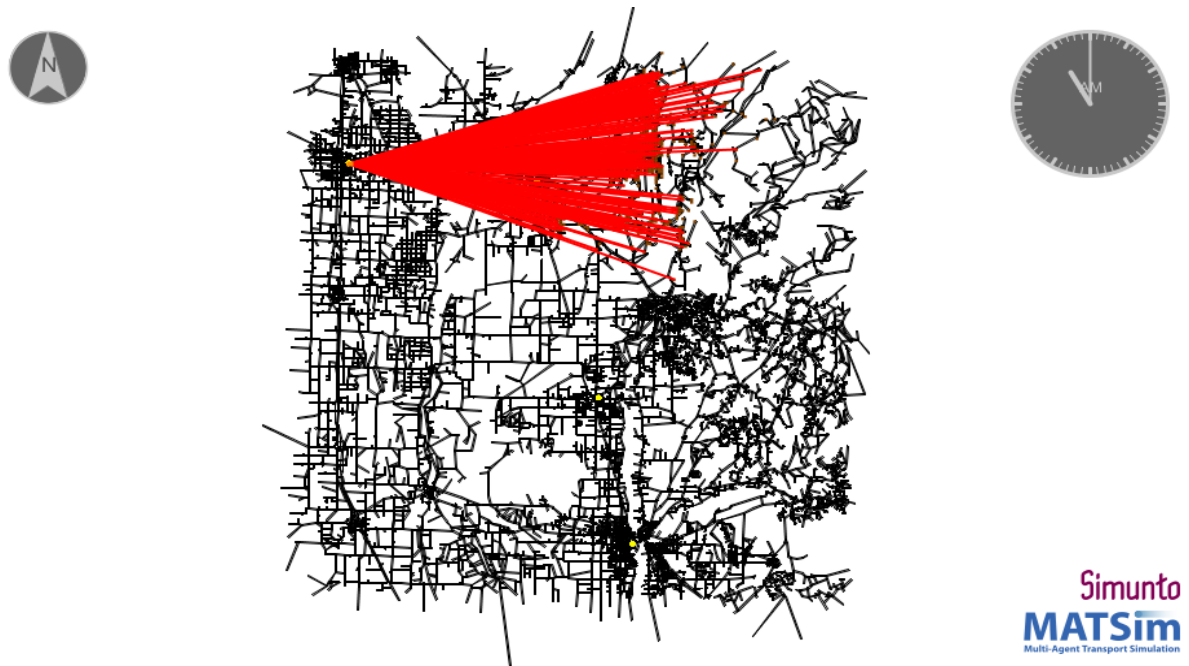


Figura 17: Distribuição dos agentes no abrigo *Glenn County Fairgrounds* na simulação nativa do MATSim.



Figura 18: Três evacuados na mesma área sendo enviados para três abrigos diferentes.

Em contrapartida, na simulação com o cenário modelado com as contribuições, os evacuados da mesma área foram encaminhados de seus pontos de origem para o abrigo mais próximo, como mostrado na Figura 19. Com isso o tempo de evacuação diminuiu, já que eles têm uma distância menor a percorrer. Na Figura 20, três evacuados que saíram da mesma área foram encaminhados para dois abrigos de acordo com a distância e número

de vagas disponíveis. Seus tempos de evacuação em minutos foram respectivamente 31:07, 45:15 e 56:35.



Figura 19: Distribuição dos agentes no abrigo *Glenn County Fairgrounds* na simulação da metodologia do trabalho.



Figura 20: Três evacuados sendo enviados para os dois abrigos mais próximos.

5.2.2 Tempo de evacuação individual

Para fins de análise do tempo de evacuação dos agentes, foram exportados três arquivos CSV com os dados de todos os evacuados que tiveram como ponto de destino o abrigo Butte County Fairgrounds. Com esses arquivos, pode-se perceber uma diminuição no tempo de evacuação. Como amostra, os dez primeiros evacuados tiveram seus tempos de evacuação comparados.

Na Tabela 3, o maior tempo de evacuação foi de 59:55 minutos e o menor, de 22:03 minutos. Já na Tabela 4, o maior tempo de evacuação foi de 48:18 minutos e o menor, de 38:18 minutos. Por último, na Tabela 5, o maior tempo de evacuação foi de 01:04:35 hora e o menor, de 38:05 minutos. Com isso, pode se constatar que: (1) por definir aleatoriamente os pontos de destinos, a primeira metodologia faz com que aja uma discrepância entre os valores máximo e mínimo de evacuação; (2) a metodologia utilizada na segunda simulação consegue alcançar um equilíbrio no tempo de evacuação de todos os agentes por distribuir evacuados nos abrigos mais próximos a esses, já que as distâncias entre os abrigos e a área de desastre são parecidas; e (3) por fazer a transição do equilíbrio de *Nash* para o sistema ótimo, a metodologia utilizada na terceira simulação causa recomendações de rotas instáveis (rápidas, mas que podem não ser as melhores), provocando assim um aumento no tempo de evacuação.

Tabela 3: Dados de dez evacuados da simulação nativa.

| Id do agente | Horário de saída | Horário de chegada | Duração em horas |
|--------------|------------------|--------------------|------------------|
| 973 | 08:00:00 | 08:55:02 | 00:55:02 |
| 924 | 08:00:00 | 08:59:15 | 00:59:15 |
| 881 | 07:27:00 | 08:09:55 | 00:42:55 |
| 972 | 08:00:00 | 08:57:12 | 00:57:12 |
| 959 | 08:00:00 | 08:30:03 | 00:30:03 |
| 923 | 08:00:00 | 08:41:17 | 00:41:17 |
| 880 | 07:27:00 | 07:49:03 | 00:22:03 |
| 971 | 08:00:00 | 08:59:55 | 00:59:55 |
| 975 | 08:00:00 | 08:58:13 | 00:58:13 |
| 922 | 08:00:00 | 08:39:17 | 00:39:17 |

Tabela 4: Dados de dez evacuados da segunda simulação.

| Id do agente | Horário de saída | Horário de chegada | Duração em horas |
|--------------|------------------|--------------------|------------------|
| 973 | 08:00:00 | 08:38:28 | 00:38:28 |
| 924 | 08:00:00 | 08:45:03 | 00:45:03 |
| 881 | 08:00:00 | 08:40:05 | 00:40:05 |
| 972 | 08:00:00 | 08:39:27 | 00:39:27 |
| 959 | 08:00:00 | 08:39:10 | 00:39:10 |
| 923 | 08:00:00 | 08:46:40 | 00:46:40 |
| 880 | 08:00:00 | 08:48:18 | 00:48:18 |
| 971 | 08:00:00 | 08:38:18 | 00:38:18 |
| 975 | 08:00:00 | 08:43:12 | 00:43:12 |
| 922 | 08:00:00 | 08:39:37 | 00:39:37 |

Tabela 5: Dados de dez evacuados da simulação da metodologia completa.

| Id do agente | Horário de saída | Horário de chegada | Duração em horas |
|--------------|------------------|--------------------|------------------|
| 973 | 08:00:00 | 08:38:17 | 00:38:17 |
| 924 | 08:00:00 | 09:00:23 | 01:00:23 |
| 881 | 07:27:00 | 08:06:57 | 00:39:57 |
| 972 | 08:00:00 | 08:39:18 | 00:39:18 |
| 959 | 08:00:00 | 09:04:35 | 01:04:35 |
| 923 | 08:00:00 | 08:43:12 | 00:43:12 |
| 880 | 07:27:00 | 08:15:37 | 00:48:37 |
| 971 | 08:00:00 | 08:38:05 | 00:38:05 |
| 975 | 08:00:00 | 08:41:53 | 00:41:53 |
| 922 | 08:00:00 | 08:38:58 | 00:38:58 |

5.2.3 Tempo de evacuação total

Na Tabela 6, o tempo de execução, o número de congestionamentos na rede, a duração destes congestionamentos e o tempo de evacuação total de cada simulação foi apresentado. Como pode-se observar, as simulações 02 e 03 tiveram tempos de evacuação melhores do que a simulação 01. Isso se dá principalmente pelo planejamento adequado do método de evacuação. Além disso, na simulação 01, com a recomendação aleatória de residentes pelos abrigos, agentes demoram para chegar nos destinos, o que afeta o tempo de evacuação do cenário como um todo. Através da Tabela 6, nota-se também que o número de congestionamentos nas simulações 02 e 03 é maior do que na simulação 01. Esse aumento de congestionamentos acontece devido a diminuição de rotas disponíveis na área de desastre durante essas simulações, já que com o tempo essas rotas vão sendo removidas da recomendação. Por isso, há um número menor de rotas e conseqüentemente um número

maior de congestionamentos. Contudo, a maioria dos congestionamentos nessas simulações ocorre fora da área de risco, principalmente próximo aos abrigos. Esta particularidade dá-se devido a rapidez com que evacuados chegam aos seus destinos. Enquanto que na primeira simulação os congestionamentos concentram-se na área de desastre.

Tabela 6: Tempo de evacuação total e congestionamentos.

| Simulações | Tempo de Execução | Quantidade de Congestionamentos | Duração dos Congestionamentos em minutos | Tempo de Evacuação Total |
|--------------|-------------------|---------------------------------|---|--------------------------|
| Simulação 01 | 00:00:58 | 4 | 21:16, 30:06, 24:53, 07:45 | 02:50:57 |
| Simulação 02 | 00:01:01 | 9 | 24:09, 30:53, 13:08, 31:14, 54:41, 17:02, 17:24, 39:50, 21:19 | 02:28:46 |
| Simulação 03 | 00:02:52 | 5 | 15:29, 15:53, 32:09, 41:51 e 13:57 | 02:42:19 |

6 Conclusão

O planejamento e simulação de desastres através de métodos de evacuação traz benefícios a pesquisadores e desenvolvedores na área, pois permite a avaliação de políticas de evacuação antes de serem colocadas em prática em um ambiente real, facilitando assim a criação de estratégias eficazes. Neste trabalho, foi aplicada uma metodologia de recomendação de rotas e, conseqüentemente, a simulação desta no cenário do desastre natural *Camp Fire*. O objetivo foi recomendar rotas utilizando distribuição de evacuados em abrigos próximos com disponibilidade de vagas, remoção de vias indisponíveis e a minimização de congestionamentos.

Como forma de analisar o desempenho das contribuições feitas, foram utilizados os tempos de evacuação das três simulações realizadas. Com isso, resultados satisfatórios quanto aos gargalos previstos e os tempo de evacuação, tanto individual quanto global, foram obtidos. Assim, informações úteis quanto ao desempenho da metodologia foram obtidas através da comparação com uma metodologia simples de evacuação. Como resultado principal, o tempo de evacuação total diminuiu em 08:38 minutos com a adoção do método completo e 22:11 minutos com a adoção do método sem o módulo de congestionamento, possibilitando assim o uso deste na criação de políticas de evacuação.

Com o desenvolvimento de trabalhos futuros, essa metodologia será aplicada a simulação de um cenário com números diferentes de evacuados e vagas em abrigos. Além disso, a melhoria do modelo de cronograma de evacuação deve ser realizado, com isso a análise do impacto de diferentes cronogramas no tempo de evacuação e no número de congestionamentos da simulação poderá ser avaliado.

7 Publicações

Capítulo aceito - Mateus P. Garcia, Orrana L. V. de Sousa, Ana C. de A. Alves, Deborah M.V. Magalhães. Processamento de Dados do *Twitter* para Monitoramento e Recomendação de Rotas em Cenários de Desastres Naturais. **X SINFO - Simpósio de Sistemas de Informação**, Picos. Piauí. 2019.

Referências

- AGAFONOVA, A.; MYASNIKOVA, V. Efficiency comparison of the routing algorithms used in centralized traffic management systems. *ScienceDirect*, 2017. Citado na página 37.
- BAELDUNG. *Dijkstra Shortest Path Algorithm in Java*. 2019. Disponível em: <<https://www.baeldung.com/java-dijkstra>>. Citado 2 vezes nas páginas 8 e 19.
- BAKILLAH HUBERT KLüPFEL, G. L. M.; GEORGWALENCIAK. Multi-agent transport simulation for regional evacuation processes. 2012. Citado na página 35.
- BENCHEKROUN, S.; MOURAOUAH, A. E. Tsunami evacuation plan for the city of tangier-morocco based on gis tools. *Journal of Tsunami Society International*, 2018. Citado 3 vezes nas páginas 28, 30 e 35.
- COUTINHO-RODRIGUES, L. T. J.; ALçADA-ALMEIDA, L. Solving a location-routing problem with a multiobjective approach: the design of urban evacuation plans. *Journal of Transport Geography*, 2012. Citado 2 vezes nas páginas 27 e 30.
- FANG QIUPING LI, Q. L. L. D. H. Z.; SHAW, S.-L. A space–time efficiency model for optimizing intra-intersection vehicle–pedestrian evacuation movements. *Transportation Research Part C*, 2013. Citado 2 vezes nas páginas 27 e 30.
- FOX40. *List of Missing in Camp Fire Down to 1*. 2019. Disponível em: <<http://bit.ly/2Oqrka1>>. Citado na página 31.
- HAKLAY, M.; WEBER, P. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 2008. Citado na página 18.
- HORNI, K. N. A.; AXHAUSEN, K. W. *The Multi-Agent Transport Simulation MATSim*. [S.l.: s.n.], 2019. 10–22 p. Citado 4 vezes nas páginas 8, 20, 21 e 22.
- KIM, S. L. J.; LEE, S. An evacuation route choice model based on multi-agent simulation in order to prepare tsunami disasters. *Transportmetrica B: Transport Dynamics*, 2017. Citado 2 vezes nas páginas 28 e 30.
- LOSANGELESTIME. *California’s Camp fire was the costliest global disaster last year, insurance report shows*. 2019. Disponível em: <<https://lat.ms/345ddxl>>. Citado na página 31.
- LUJAK, M.; GIORDANI, S. Centrality measures for evacuation: Finding agile evacuation routes. *Future Generation Computer Systems*, 2018. Citado 2 vezes nas páginas 29 e 30.
- LÄMMEL, G.; FLÖTTERÖD, G. Towards system optimum: Finding optimal routing strategies in time dependent networks for large-scale evacuation problems. 2009. Citado 2 vezes nas páginas 27 e 30.
- LÄMMEL, M. R. G.; NAGEL, K. Bottlenecks and congestion in evacuation scenarios: A microscopic evacuation simulation for large-scale disasters. 2008. Citado na página 35.
- MATSIM. 2019. Disponível em: <<https://matsim.org/>>. Citado na página 15.

- MATTAR, M. R. M. Deslocamentos ambientais: O caso dos pequenos países insulares. 2011. Citado na página 14.
- NEWSWEEK. *What Fires Are Still Burning In California? Update, Map, Containment Lateste*. 2018. Disponível em: <<http://bit.ly/33Vxh58>>. Citado 2 vezes nas páginas 8 e 33.
- ONUBR. *Relatório da ONU alerta para aumento dramático das perdas econômicas provocadas por desastres*. 2018. Disponível em: <<http://bit.ly/2XqCDD5>>. Citado na página 14.
- OPENSTREETMAP. 2019. Disponível em: <<https://www.openstreetmap.org/about>>. Citado na página 18.
- SAADATSERESHT, A. M. M.; TALEAI, M. Evacuation planning using multiobjective evolutionary optimization approach. *European Journal of Operational Research*, 2009. Citado 2 vezes nas páginas 26 e 30.
- SFGATE. *Satellite image shows incredible fiery heat of Camp Fire*. 2018. Disponível em: <<http://bit.ly/32Mv0be>>. Citado 3 vezes nas páginas 8, 31 e 32.
- SHAHABI, K.; WILSON, J. P. Casper: Intelligent capacity-aware evacuation routing. *Computers, Environment and Urban Systems*, 2014. Citado na página 17.
- STEPANOV, A.; SMITH, J. M. Multi-objective evacuation routing in transportation networks. *European Journal of Operational Research*, 2008. Citado 2 vezes nas páginas 26 e 30.
- TIMES, L. A. *As California's deadliest wildfire closed in, evacuation orders were slow to arrive*. 2018. Disponível em: <<https://lat.ms/2s3B1Uf>>. Citado 2 vezes nas páginas 8 e 35.
- VEERASWAMY EDWIN R. GALEA, L. F. P. J. L. S. H. R. J. G. A.; SMITH, T. E. The simulation of urban-scale evacuation scenarios with application to the swinley forest fire. *Safety Science*, 2018. Citado 2 vezes nas páginas 29 e 30.
- WIRED. *These Wind Patterns Explain Why California's Wildfires Are So Bad*. 2018. Disponível em: <<http://bit.ly/2DdgGhL>>. Citado na página 32.
- ZENG, W.; CHURCH, R. L. *Finding shortest paths on real road networks: the case for A**. Zenodo, 2009. Disponível em: <<https://doi.org/10.1080/13658810801949850>>. Citado na página 19.
- ZHU KUN XIE, K. O. Y.; YANG, H. Hurricane evacuation modeling using behavior models and scenario-driven agent-based simulations. *ScienceDirect*, 2018. Citado na página 37.

Apêndices

APÊNDICE A – *Script* para a criação da rede MATSim

```

1  class NetworkGeneratorCalifornia {
3
4      public static final String CALIFORNIACoord = "EPSG:3311";
5
6      public static void main(String [] args) throws SAXException {
7          String osm = "nomedoarquivo.osm";
8
9          Scenario sc = ScenarioUtils.createScenario(ConfigUtils.createConfig()) ;
10         Network net = sc.getNetwork();
11
12         CoordinateTransformation ct = TransformationFactory.getCoordinateTransformation(
13             TransformationFactory.WGS84, CALIFORNIACoord);
14
15         //constrói um novo leitor de openstreetmap
16         OsmNetworkReader onr = new OsmNetworkReader(net,ct);
17         onr.setHighwayDefaults(1, "null", 1, 60/3.6, 1, 600);
18         onr.setHighwayDefaults(1, "service", 1, 60/3.6, 1, 600);
19         onr.parse(osm); //começa a conversão de .osm para matsim
20
21         new NetworkCleaner().run(net); //remove links isolados ou não conectados
22         String filename = "nomedoarquivo.xml.gz";
23         new NetworkWriter(net).write(filename);
24
25         // Cria um ESRI shape file da rede MATSim
26
27         FeatureGeneratorBuilderImpl builder = new FeatureGeneratorBuilderImpl(net,
28             CALIFORNIACoord);
29         builder.setWidthCoefficient(0.01);
30         builder.setFeatureGeneratorPrototype(PolygonFeatureGenerator.class);
31         builder.setWidthCalculatorPrototype(CapacityBasedWidthCalculator.class);
32         new Links2ESRIShape(net, "nomedoarquivo.shp", builder).write();
33     }
34 }

```

Código Fonte A.1: Criando a rede MATSim.

APÊNDICE B – *Script* para a criação da população MATSim

```

1 public class NearShelterLotation{
2
3     private static int ID = 1;
4     private static final String Cali = "EPSG:3311";
5     private static final String exampleDirectory = "";
6     private static final Logger log = Logger.getLogger(NearShelterLotation.class);
7
8     public static void main(String[] args) throws IOException {
9
10        final String NETWORKFILE = "nomedoarquivo.xml";
11        CoordinateTransformation ct = TransformationFactory.getCoordinateTransformation(
12            TransformationFactory.WGS84, Cali);
13
14        // input files
15        String zonesFile1 = exampleDirectory + "area1.shp";
16        String zonesFile2 = exampleDirectory + "area2.shp";
17        String zonesFile3 = exampleDirectory + "area3.shp";
18        String zonesFile4 = exampleDirectory + "area4.shp";
19
20        Scenario scenario = ScenarioUtils.createScenario(ConfigUtils.createConfig());
21        new MatsimNetworkReader(scenario.getNetwork()).readFile(NETWORKFILE);
22        Network network = scenario.getNetwork();
23
24        // reads the shape file in
25        SimpleFeatureSource area1 = ShapeFileReader.readDataFile(zonesFile1);
26        SimpleFeatureSource area2 = ShapeFileReader.readDataFile(zonesFile2);
27        SimpleFeatureSource area3 = ShapeFileReader.readDataFile(zonesFile3);
28        SimpleFeatureSource area4 = ShapeFileReader.readDataFile(zonesFile4);
29
30        SimpleFeature home1 = null;
31        SimpleFeature home2 = null;
32        SimpleFeature home3 = null;
33        SimpleFeature home4 = null;
34
35        // Iterator to iterate over the features from the shape file
36        SimpleFeatureIterator it1 = area1.getFeatures().features();
37        SimpleFeatureIterator it2 = area2.getFeatures().features();
38        SimpleFeatureIterator it3 = area3.getFeatures().features();
39        SimpleFeatureIterator it4 = area4.getFeatures().features();
40
41        ArrayList<SimpleFeature> t1 = new ArrayList<SimpleFeature>();
42        ArrayList<SimpleFeature> t2 = new ArrayList<SimpleFeature>();
43        ArrayList<SimpleFeature> t3 = new ArrayList<SimpleFeature>();
44        ArrayList<SimpleFeature> t4 = new ArrayList<SimpleFeature>();
45
46        while (it1.hasNext()) {
47            home1 = it1.next();
48            Geometry gm = (Geometry) home1.getDefaultGeometry();

```

```
49     home1.setDefaultGeometry(gm);
50     t1.add(home1);
51 }
52
53 while (it2.hasNext()) {
54     home2 = it2.next();
55     Geometry gm2 = (Geometry) home2.getDefaultGeometry();
56     home2.setDefaultGeometry(gm2);
57     t2.add(home2);
58 }
59
60 while (it3.hasNext()) {
61     home3 = it3.next();
62     Geometry gm3 = (Geometry) home3.getDefaultGeometry();
63     home3.setDefaultGeometry(gm3);
64     t3.add(home3);
65 }
66
67 while (it4.hasNext()) {
68     home4 = it4.next();
69     Geometry gm4 = (Geometry) home4.getDefaultGeometry();
70     home4.setDefaultGeometry(gm4);
71     t4.add(home4);
72 }
73
74 it1.close();
75 it2.close();
76 it3.close();
77 it4.close();
78
79 createPersons(scenario, t1, t2, t3, t4, (int) 1800, ct);
80 createActivities(scenario, ct, network); // this method creates the remaining activities
81
82 String popFilename = "nomedoarquivo.xml.gz";
83 // and finally the population will be written to a xml file
84 new PopulationWriter(scenario.getPopulation(), scenario.getNetwork()).write(popFilename);
85 log.info("population written to: " + popFilename);
86
87 }
88
89 private static void createPersons(Scenario scenario, ArrayList<SimpleFeature> t1,
90 ArrayList<SimpleFeature> t2, ArrayList<SimpleFeature> t3, ArrayList<SimpleFeature> t4,
91 int number, CoordinateTransformation ct) {
92
93     Population pop = scenario.getPopulation();
94     PopulationFactory pb = pop.getFactory();
95
96     for (number=0; number<450; number++) {
97         // 37:00 minutos de diferença entre o fim das atividades home
98
99         Person pers = pb.createPerson(Id.create(ID++, Person.class));
100        pop.addPerson(pers);
101        Plan plan = pb.createPlan();
102        Coord c = getCoordInGeometry(t1);
103        Activity act = pb.createActivityFromCoord("home", new Coord(c.getX(), c.getY()));
104        act.setEndTime(7.383333333333333*3600); //07:23 7.383333333333333
105        plan.addActivity(act);
```

```

    pers.addPlan(plan);
107 }
    for (number= 450; number<900; number++) {
109     Person pers = pb.createPerson(Id.create(ID++, Person.class));
        pop.addPerson(pers);
111     Plan plan = pb.createPlan();
        Coord c = getCoordInGeometry(t2);
113     Activity act = pb.createActivityFromCoord("home", new Coord(c.getX(), c.getY()));
        plan.addActivity(act);
115     act.setEndTime(8*3600); //08:00
        pers.addPlan(plan);
117 }
    for (number= 900; number<1350; number++) {
119     Person pers = pb.createPerson(Id.create(ID++, Person.class));
        pop.addPerson(pers);
121     Plan plan = pb.createPlan();
        Coord c = getCoordInGeometry(t3);
123     Activity act = pb.createActivityFromCoord("home", new Coord(c.getX(), c.getY()));
        plan.addActivity(act);
125     act.setEndTime(8.616666667*3600); //08:37
        pers.addPlan(plan);
127 }
    for (number= 1350; number<1800; number++) {
129     Person pers = pb.createPerson(Id.create(ID++, Person.class));
        pop.addPerson(pers);
131     Plan plan = pb.createPlan();
        Coord c = getCoordInGeometry(t4);
133     Activity act = pb.createActivityFromCoord("home", new Coord(c.getX(), c.getY()));
        plan.addActivity(act);
135     act.setEndTime(9.233333333333333*3600); //09:14 9.233333333333333
        pers.addPlan(plan);
137 }
}
139
private static void createActivities(Scenario scenario, CoordinateTransformation ct,
141 Network network) {

143     Population pop = scenario.getPopulation();
    PopulationFactory pb = pop.getFactory(); // the population builder creates all we need
145     Coord shelter = new Coord();
    List<Node> coord = new ArrayList<>();
147     Node[] y = getShelter(ct, network);
    coord = getShelterLotation(y, network);
149

    // this loop iterates over all persons
151     for (Person pers : pop.getPersons().values()) {
        // each person has exactly one plan, that has been created in createPersons(...)
153     Plan plan = pers.getPlans().get(0);

155     List<Integer> valuesList = getNearShelterPointInFeature(ct, scenario, network,
        (Activity) plan.getPlanElements().get(0));
157     int key = valuesList.get(0);
        int key1 = valuesList.get(1);
159     int key2 = valuesList.get(2);
        int key3 = valuesList.get(3);
161     int key4 = valuesList.get(4);

```

```

163 Leg leg = pb.createLeg(TransportMode.car);
    plan.addLeg(leg); // there needs to be a log between two activities
165
    if (coord.contains(y[key])) {
167         coord.remove(y[key]);
            shelter = y[key].getCoord();
169     }else if ((!coord.contains(y[key])) && (coord.contains(y[key1]))){
            coord.remove(y[key1]);
171         shelter = y[key1].getCoord();
    }else if ((!coord.contains(y[key])) && (!coord.contains(y[key1])) && (coord.contains(
173     y[key2]))) {
            coord.remove(y[key2]);
175         shelter = y[key2].getCoord();
    }else if ((!coord.contains(y[key])) && (!coord.contains(y[key1])) && (!coord.contains(
177     y[key2])) && (coord.contains(y[key3]))) {
            coord.remove(y[key3]);
179         shelter = y[key3].getCoord();
    }else if ((!coord.contains(y[key])) && (!coord.contains(y[key1])) && (!coord.contains(
181     y[key2])) && (!coord.contains(y[key3])) && (coord.contains(y[key4]))) {
            coord.remove(y[key4]);
183         shelter = y[key4].getCoord();
    }
185
    // shelter activity on a random shelter among the shelter set
187     Activity shelt = pb.createActivityFromCoord("shelter", new Coord((shelter.getX()),
        (shelter.getY())));
189     double startTime = 10 * 3600;
        shelt.setStartTime(startTime);
191     plan.addActivity(shelt);
    }
193
}
195
public static Coord getCoordInGeometry(ArrayList<SimpleFeature> t) {
197     Iterator<SimpleFeature> iter = t.iterator();
        Collections.shuffle(t);
199
        RandomPointsBuilder randomPointsBuilder = new RandomPointsBuilder(new GeometryFactory());
201     randomPointsBuilder.setNumPoints(1);
        randomPointsBuilder.setExtent((Geometry) iter.next().getDefaultGeometry());
203     Coordinate coordinate = randomPointsBuilder.getGeometry().getCoordinates()[0];
        return MGC.coordinate2Coord(coordinate);
205 }
207
public static List<Integer> getNearShelterPointInFeature(CoordinateTransformation ct,
209     Scenario scenario, Network network, Activity homeAct) {
        Map<Double, Integer> travelLength = new LinkedHashMap<Double, Integer>();
211     Map<Integer, Path> path = new LinkedHashMap<Integer, Path>();
        TravelTime travelTimes = null;
213     TravelDisutility travelCosts = null;
        Node[] y = getShelter(ct, network);
215     MatsimClassDijkstra least = new MatsimClassDijkstra(network, travelCosts, travelTimes);
217
        Coord x = homeAct.getCoord();
        Node node = NetworkUtils.getNearestNode(network, x);
219

```

```
221     for (int i = 0; i < y.length; i++) {
222         Node node1 = y[i];
223         path.put(i, least.calcLeastCostPath(node, node1, 0.0, null, null));
224     }
225
226     for (int i = 0; i < path.size(); i++) {
227         travelLength.put(getLinkTravelDisutility(path.get(i).links, 0.0, null, null), i);
228     }
229
230     final Map<Double, Integer> sortedByCount = travelLength.entrySet().stream().sorted(
231     Map.Entry.comparingByKey()).collect(Collectors.toMap(Map.Entry::
232     getKey, Map.Entry::getValue, (e1, e2) -> e1, LinkedHashMap::new));
233     List<Integer> valuesList = new ArrayList<Integer>(sortedByCount.values());
234
235     return valuesList;
236 }
237
238 public static double getLinkTravelDisutility(List<Link> link, double time, Person person,
239 Vehicle vehicle) {
240     double length = 0;
241     for (Link link1 : link) {
242         length = link1.getLength() + length;
243     }
244     return length;
245 }
246
247 public static Node[] getShelter (CoordinateTransformation ct, Network network){
248
249     Coord shelterYSF = new Coord ((double) -121.610496, (double) 39.130820);
250     Coord coordYSF = ct.transform(shelterYSF);
251     Node YSF = NetworkUtils.getNearestNode(network, coordYSF);
252
253     Coord shelterBCF = new Coord ((double) -121.684840, (double) 39.366423);
254     Coord coordBCF = ct.transform(shelterBCF);
255     Node BCF = NetworkUtils.getNearestNode(network, coordBCF);
256
257     Coord shelterEAC = new Coord ((double) -121.830866, (double) 39.761694);
258     Coord coordEAC = ct.transform(shelterEAC);
259     Node EAC = NetworkUtils.getNearestNode(network, coordEAC);
260
261     Coord shelterGCF = new Coord ((double) -122.181001, (double) 39.742266);
262     Coord coordGCF = ct.transform(shelterGCF);
263     Node GCF = NetworkUtils.getNearestNode(network, coordGCF);
264
265     Coord shelterSDF = new Coord ((double) -121.812576, (double) 39.717312);
266     Coord coordSDF = ct.transform(shelterSDF);
267     Node SDF = NetworkUtils.getNearestNode(network, coordSDF);
268
269     Node[] y = new Node[5];
270
271     y[0] = EAC;
272     y[1] = GCF;
273     y[2] = SDF;
274     y[3] = YSF;
275     y[4] = BCF;
```

```
277     return y;
279 }
281 public static List<Node> getShelterLotation(Node[] list, Network network) {
283     List<Node> places = new ArrayList<>();
285     for (int i=0; i<620; i++) {
287         places.add(list[2]);
289     }
291     for (int i=0; i<250; i++) {
293         places.add(list[0]);
295     }
297     for (int i=0; i<100; i++) {
299         places.add(list[4]);
301     }
303     for (int i=0; i<450; i++) {
305         places.add(list[1]);
307     }
309     for (int i=0; i<380; i++) {
311         places.add(list[3]);
313     }
315     return places;
317 }
319 }
```

Código Fonte B.1: Criando a população *MATSim*.

APÊNDICE C – *Script* para a criação do arquivo *networkChangeEvents*

```

public class EventsTeste {
2
    private static final String exampleDirectory = "";
4
    final static String NETWORKFILE = exampleDirectory + "nomedoarquivo.xml";
    public static final String xmlFilePath = exampleDirectory + "nomedoarquivo.xml";
6
    public static void main(String[] args) {
8
        Scenario scenario = ScenarioUtils.createScenario(ConfigUtils.createConfig());
10
        new MatsimNetworkReader(scenario.getNetwork()).readFile(NETWORKFILE);
12
        try {
14
            DocumentBuilderFactory documentFactory = DocumentBuilderFactory.newInstance();
16
            DocumentBuilder documentBuilder = documentFactory.newDocumentBuilder();
18
            Document document = documentBuilder.newDocument();
20
            // root element
            Element root = document.createElement("networkChangeEvents");
22
            document.appendChild(root);
24
            Attr attrN = document.createAttribute("xmlns");
            attrN.setValue("http://www.matsim.org/files/dtd");
26
            root.setAttributeNode(attrN);
            Attr attrC = document.createAttribute("xmlns:xsi");
28
            attrC.setValue("http://www.w3.org/2001/XMLSchema-instance");
            root.setAttributeNode(attrC);
30
            Attr attrE = document.createAttribute("xsi:schemaLocation");
            attrE.setValue("http://www.matsim.org/files/dtd http://www.matsim.org/files/dtd/
32
            networkChangeEvents.xsd");
            root.setAttributeNode(attrE);
34
            // events element
36
            Element events = document.createElement("networkChangeEvent");
38
            root.appendChild(events);
40
            // set an attribute to staff element
            Attr attr = document.createAttribute("startTime");
42
            attr.setValue("09:00:00");
            events.setAttributeNode(attr);
44
            for ( Link link1 : scenario.getNetwork().getLinks().values() ) {
46
                Element link = document.createElement("link");
                Attr attr1 = document.createAttribute("refId");
48
                attr1.setValue(link1.getId().toString());

```

```
50         link.setAttributeNode(attr1);
51         events.appendChild(link);
52     }
53
54     Element link2 = document.createElement("flowCapacity");
55     Attr attr1 = document.createAttribute("type");
56     Attr attr2 = document.createAttribute("value");
57     attr1.setValue("absolute");
58     attr2.setValue("0");
59     link2.setAttributeNode(attr1);
60     link2.setAttributeNode(attr2);
61     events.appendChild(link2);
62
63     Element link3 = document.createElement("freespeed");
64     Attr attr3 = document.createAttribute("type");
65     Attr attr4 = document.createAttribute("value");
66     attr3.setValue("absolute");
67     attr4.setValue("0.0");
68     link3.setAttributeNode(attr3);
69     link3.setAttributeNode(attr4);
70     events.appendChild(link3);
71
72     // create the xml file
73     //transform the DOM Object to an XML File
74     TransformerFactory transformerFactory = TransformerFactory.newInstance();
75     Transformer transformer = transformerFactory.newTransformer();
76     transformer.setOutputProperty(OutputKeys.INDENT, "yes");
77     DOMSource domSource = new DOMSource(document);
78     StreamResult streamResult = new StreamResult(new File(xmlFilePath));
79
80     transformer.transform(domSource, streamResult);
81
82     System.out.println("Done creating XML File");
83
84     } catch (ParserConfigurationException pce) {
85         pce.printStackTrace();
86     } catch (TransformerException tfe) {
87         tfe.printStackTrace();
88     }
89 }
90 }
```

Código Fonte C.1: Script para criar *networkChangeEvents*.

APÊNDICE D – *Script* para a minimização de congestionamentos

```

1  public class DecongestionModule extends AbstractModule {
2
3      private final DecongestionConfigGroup decongestionConfigGroup;
4
5      public DecongestionModule(Scenario scenario) {
6          this.decongestionConfigGroup = (DecongestionConfigGroup) scenario.getConfig().
7          getModules().get(DecongestionConfigGroup.GROUP_NAME);
8      }
9
10     @Override
11     public void install() {
12
13         if (decongestionConfigGroup.isEnableDecongestionPricing()) {
14             System.out.println(decongestionConfigGroup.isEnableDecongestionPricing());
15             switch( decongestionConfigGroup.getDecongestionApproach() ) {
16                 case BangBang:
17                     this.bind(DecongestionTollingBangBang.class).asEagerSingleton();
18                     this.bind(DecongestionTollSetting.class).to(DecongestionTollingBangBang.
19                     class);
20                     break;
21                 case PID:
22                     this.bind(DecongestionTollingPID.class).asEagerSingleton();
23                     this.bind(DecongestionTollSetting.class).to(DecongestionTollingPID.class);
24                     this.addEventHandlerBinding().to(DecongestionTollingPID.class);
25                     break;
26                 case P_MC:
27                     this.bind(DecongestionTollingP_MCP.class).asEagerSingleton();
28                     this.bind(DecongestionTollSetting.class).to(DecongestionTollingP_MCP.class);
29                     this.addEventHandlerBinding().to(DecongestionTollingP_MCP.class);
30                     break;
31                 default:
32                     throw new RuntimeException("not implemented" );
33             }
34         }
35     }
36
37     this.bind(DecongestionInfo.class).asEagerSingleton();
38
39     this.bind(IntervalBasedTollingAll.class).asEagerSingleton();
40     this.bind(IntervalBasedTolling.class).to(IntervalBasedTollingAll.class);
41     this.addEventHandlerBinding().to(IntervalBasedTollingAll.class);
42
43     this.bind(DelayAnalysis.class).asEagerSingleton();
44     this.addEventHandlerBinding().to(DelayAnalysis.class);
45
46     this.addEventHandlerBinding().to(PersonVehicleTracker.class).asEagerSingleton();
47
48     this.addControllerListenerBinding().to(DecongestionControllerListener.class);

```

```
49 |     }  
51 | }
```

Código Fonte D.1: *Script* para minimizar congestionamentos.

APÊNDICE E – *Script* para simular descongestionamento

```

public class DecongestionRunExampleFromConfig {
2
    private static final Logger log = Logger.getLogger(DecongestionRunExampleFromConfig.class);
4
    private static String configFile;
    private static final String outputDirectory = "" ;
6
    public static void main(String[] args) throws IOException {
8
        if (args.length > 0) {
            log.info("Starting simulation run with the following arguments:");
10
            configFile = args[0];
12
            log.info("config file: "+ configFile);
14
        } else {
            configFile = "config.xml";
16
        }
18
        DecongestionRunExampleFromConfig main = new DecongestionRunExampleFromConfig();
        main.run();
20
    }
22
    private void run() throws IOException {
24
        Config config = ConfigUtils.loadConfig(configFile, new DecongestionConfigGroup());
26
        final Scenario scenario = ScenarioUtils.loadScenario(config);
        Controller controller = new Controller(scenario);
28
        controller.addOverridingModule(new DecongestionModule(scenario));
30
        final TollTimeDistanceTravelDisutilityFactory travelDisutilityFactory = new
32
        TollTimeDistanceTravelDisutilityFactory();
        travelDisutilityFactory.setSigma(0.);
34
        controller.addOverridingModule(new AbstractModule(){
36
            @Override
            public void install() {
38
                this.bindCarTravelDisutilityFactory().toInstance( travelDisutilityFactory );
40
            }
        });
42
        controller.getConfig().controller().setOutputDirectory(outputDirectory);
        controller.getConfig().controller().setOverwriteFileSetting(OverwriteFileSetting.
44
        overwriteExistingFiles);
        controller.run();
46
    }
}

```

Código Fonte E.1: *Script* para simular descongestionamentos no cenário.

APÊNDICE F – Arquivo de configuração utilizado na simulação

```

<?xml version="1.0" ?>
<!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">
<config>

  <module name="global">
    <param name="randomSeed" value="4711" />
    <param name="coordinateSystem" value="EPSG:3311" />
  </module>

  <module name="network">
    <param name="inputNetworkFile" value="network.xml.gz" />
    <param name="timeVariantNetwork" value="true" />
    <param name="inputChangeEventsFile" value="networkChangeEvents.xml" />
  </module>

  <module name="plans">
    <param name="inputPlansFile" value="population1.xml.gz" />
  </module>

  <module name="controler">
    <param name="outputDirectory" value="./output" />
    <param name="firstIteration" value="0" />
    <param name="lastIteration" value="5" />
    <param name="writeEventsInterval" value="1" />
  </module>

  <module name="qsim">
    <param name="startTime" value="00:00:00" />
    <param name="endTime" value="23:00:00" />
    <param name="snapshotperiod" value="06:00:00"/>
  </module>

  <module name="planCalcScore">
    <param name="learningRate" value="1.0" />
    <param name="BrainExpBeta" value="2.0" />

    <param name="lateArrival" value="-18" />
    <param name="earlyDeparture" value="-0" />
    <param name="performing" value="+6" />
    <param name="traveling" value="-6" />
    <param name="waiting" value="-0" />

    <param name="activityType_0" value="home" />
    <param name="activityPriority_0" value="1" />
    <param name="activityTypicalDuration_0" value="07:00:00" />
    <param name="activityMinimalDuration_0" value="06:00:00" />
    <param name="activityOpeningTime_0" value="00:00:00"/>
    <param name="activityClosingTime_0" value="09:14:00" />
  </module>

```

```
<param name="activityType_1" value="shelter" />
<param name="activityPriority_1" value="1" />
<param name="activityTypicalDuration_1" value="16:00:00" />
<param name="activityMinimalDuration_1" value="06:00:00" />
<param name="activityOpeningTime_1" value="07:23:00" />
<param name="activityLatestStartTime_1" value="09:14:00" />
<param name="activityEarliestEndTime_1" value="" />
<param name="activityClosingTime_1" value="" />
</module>

<module name="strategy">
  <param name="maxAgentPlanMemorySize" value="5" />

  <param name="ModuleProbability_1" value="0.9" />
  <param name="Module_1" value="BestScore" />

  <param name="ModuleProbability_2" value="0.1" />
  <param name="Module_2" value="ReRoute" />
</module>
</config>
```

Código Fonte F.1: *Script* de configuração.



**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
() Dissertação
() Monografia
() Artigo

Eu, Anana Lhaynher Zelas de Sousa,
autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de
02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar,
gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação
Metodologia para Recomendação de Rotas de
Evacuação em Cenário de Desastre Natural
de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título
de divulgação da produção científica gerada pela Universidade.

Picos-PI 29 de janeiro de 2020.

Anana L. Z. de Sousa
Assinatura