



Universidade Estadual de Campinas
Instituto de Computação



Matheus Ferraroni Sanches

Alocação de *RoadSide Units* Ciente de Obstáculos e
Diferentes Modelos de Propagação de Sinal

CAMPINAS
2020

Matheus Ferraroni Sanches

**Alocação de *RoadSide Units* Ciente de Obstáculos e Diferentes
Modelos de Propagação de Sinal**

Dissertação apresentada ao Instituto de
Computação da Universidade Estadual de
Campinas como parte dos requisitos para a
obtenção do título de Mestre em Ciência da
Computação.

Orientador: Prof. Dr. Leandro Aparecido Villas

Este exemplar corresponde à versão final da
Dissertação defendida por Matheus Ferraroni
Sanches e orientada pelo Prof. Dr. Leandro
Aparecido Villas.

CAMPINAS
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Sanches, Matheus Ferraroni, 1996-
Sa55a Alocação de RoadSide Units ciente de obstáculos e diferentes modelos de propagação de sinal / Matheus Ferraroni Sanches. – Campinas, SP : [s.n.], 2020.

Orientador: Leandro Aparecido Villas.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Ciência da computação. 2. Redes de computadores. 3. Otimização combinatória. 4. Algoritmos evolutivos. I. Villas, Leandro Aparecido, 1983-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Allocation of RoadSide Units aware of obstacles and different signal propagation models

Palavras-chave em inglês:

Computer science

Computer networks

Combinatorial optimization

Evolutionary algorithms

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Leandro Aparecido Villas [Orientador]

Roberto Sadao Yokoyama

Lucas Francisco Wanner

Data de defesa: 27-04-2020

Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-6904-7873>

- Currículo Lattes do autor: <http://lattes.cnpq.br/7066133969704063>



Universidade Estadual de Campinas
Instituto de Computação



Matheus Ferraroni Sanches

Alocação de *RoadSide Units* Ciente de Obstáculos e Diferentes Modelos de Propagação de Sinal

Banca Examinadora:

- Prof. Dr. Leandro Aparecido Villas
Universidade Estadual de Campinas
- Prof. Dr. Roberto Sadao Yokoyama
Universidade Federal do ABC
- Prof. Dr. Lucas Francisco Wanner
Universidade Estadual de Campinas

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 27 de abril de 2020

Resumo

Os avanços em tecnologias de informação e comunicação têm oferecido condições para que possam ser criados diversos serviços que melhoram a mobilidade urbana e, consequentemente, a vida das pessoas. Para que esses serviços funcionem da maneira como foram planejados é necessário alcançar requisitos mínimos de latência, disponibilidade e confiança. Com a utilização de *Roadside Units (RSUs)* é possível estabelecer a comunicação entre veículos e infraestrutura, desempenhando papel fundamental para atender tais requisitos. Para que as RSUs possam ter a maior cobertura possível, é necessário que a alocação desses dispositivos seja feita, de modo eficaz, maximizando a área coberta e evitando desperdiçar recursos com coberturas de RSUs sobrepostas, já que o custo dos equipamentos e de implantação são elevados.

Para realizar uma boa alocação de RSUs é preciso que as áreas de cobertura desses dispositivos estejam bem definidas, considerando o impacto causado por obstáculos que estão presentes no cenário. Além disso, definir a área de cobertura utilizando modelos de propagação de sinal ideais também pode resultar em diferenças para o mundo real. Dessa forma, o trabalho atual apresenta o mapeamento da cobertura das RSUs utilizando simuladores, realiza alocação com um algoritmo genético e apresenta um caso de uso. Para auxiliar trabalhos futuros e outros pesquisadores foi desenvolvida uma ferramenta capaz de definir a área de cobertura de um dispositivo, considerando diversos parâmetros da comunicação e do cenário. Uma nova alocação das RSUs também foi feita utilizando a ferramenta proposta, considerando diferentes modelos de propagação de sinal. O impacto nas aplicações é de um aumento de aproximadamente 15% na precisão da visualização da rede, que se trata de uma métrica para representar o quão bem uma central consegue receber conhecimento e representar a situação atual, o que pode ter uma grande influência no desempenho da aplicação.

Abstract

Advances in information and communication technology grant the conditions to create various services that improve the urban mobility and, consequently, increase the quality of life of the people. In order for these services to function properly, it is necessary to achieve minimum requirements for latency, availability and reliability. The use of Roadside Units (RSUs) allows communication between vehicles and infrastructure, playing a fundamental role in meeting these requirements. In order for RSUs have the biggest coverage area, it is necessary that the deployment of these devices is done, effectively, maximizing the covered area and avoiding wasting resources with overlapping coverage areas of RSUs, since the cost of the equipments and implantation are high.

To achieve a good allocation of the RSUs, it is necessary that the coverage of these devices are well defined, considering the impact caused by obstacles that are present in the scenario. In addition, defining the coverage using ideal signal propagation models can also result in differences to the real world. In this way, this work presents the mapping of the coverage area of RSUs using simulators, performs allocation. To assist future work and other researchers, a tool was developed capable of defining the communication area of a device, considering several communication and scenario parameters. A new RSU allocation was also made using the proposed tool, considering different models of signal propagation. The impact on applications is an increase of approximately 15% in the precision of the visualization of the network, which is a metric used to represent how good a central can receive knowledge and represent the actual situation, which can have a great impact on the performance of the application.

Lista de Figuras

2.1	Algoritmo Genético	18
2.2	Tipos de comunicação em VANETs	19
2.3	Perda de sinal	21
2.4	Propagação do sinal	24
2.5	Comparação entre o impacto de obstáculos e o modelo FreeSpace	25
2.6	OpenStreetMaps	26
2.7	Simulator of Urban MObility	27
2.8	VEINS	28
4.1	Área de comunicação com diferentes RSSI	35
4.2	Cenário utilizado	36
4.3	Exemplo de resultado obtido com 8 RSUS	39
4.4	Comparação entre porcentagens de cobertura e sobreposição	41
4.5	Funcionamento do calculo da cobertura	41
4.6	Cenário com sistema de re-roteamento inteligente	43
4.7	Comparação entre ORC e CHIMERA - Tempo.	44
4.8	Comparação entre ORC e CHIMERA - Precisão.	45
4.9	Arquitetura desenvolvida	47
4.10	Classes de propagação de sinal	50
4.11	Exemplo de resultado obtido	52
4.12	BER	52
4.13	Área vs RSSI mínimo	52
4.14	Exemplo de área de comunicação obtido para uma RSU	54
4.15	Comparação entre cobertura e sobreposição	57
4.16	Resultado utilizando posições com espaçamento fixo	58
4.17	Resultado utilizando cruzamentos para alocação	58

Lista de Tabelas

3.1	Comparação de trabalhos relacionados	32
4.1	Parâmetros da Execução	35
4.2	Configurações do GA	37
4.3	Configurações disponíveis	46
4.4	Diferenças e evoluções	53
4.5	Parâmetros utilizados no RP-Sim	54
4.6	Configurações do cenário	55
4.7	Configurações do algoritmo genético	56

Lista de Abreviações e Siglas

- **RSU:** *Roadside Unit*
- **RP-Sim:** RadioPropagation Simulator
- **ITS:** Sistemas de Transporte Inteligente
- **VANETs:** Redes Veiculares Ad hoc
- **GA:** Algoritmo Genético
- **MANET:** Redes Móveis Ad-Hoc
- **OBU:** On Board Unit
- **V2V:** Veículo-para-Veículo
- **V2I:** Veículos-Para-Infraestrutura
- **V2H:** Comunicação Híbrida
- **V2X:** Veículo-para-Tudo
- **DSRC:** *Dedicated Short Range Communication*
- **GIS:** Sistema de Informações Geográficos
- **SUMO:** Simulator of Urban Mobility
- **ORC:** Otimizando Alocação de Infraestrutura com Algoritmo Genético
- **BER:** Taxa de erro de bits

Lista de Equações

- **Equação 2.1:** Fórmula do FreeSpace
- **Equação 2.2:** Fórmula do Egli
- **Equação 2.3:** Fórmula do Ericsson
- **Equação 2.4:** Fórmula do SUI
- **Equação 4.1 e 4.2:** Função objetivo

Sumário

1	Introdução	13
1.1	Motivação	15
1.2	Objetivos	15
1.3	Estrutura da Dissertação	16
2	Fundamentação Teórica	17
2.1	Otimização Combinatória	17
2.2	Redes Veiculares (VANETs)	18
2.3	Propagação de Sinal sem Fio	20
2.3.1	FreeSpace	21
2.3.2	Egli	22
2.3.3	Ericsson	22
2.3.4	Stanford University Interim - SUI	22
2.3.5	Impacto dos Obstáculos na Propagação do Sinal	23
2.4	Ferramentas Utilizadas	25
2.4.1	OpenStreetMaps	25
2.4.2	Simulator of Urban Mobility - SUMO	25
2.4.3	OMNeT++	27
2.4.4	VEINS	27
3	Trabalhos Relacionados	29
4	Alocação de RoadSide Units	33
4.1	Otimizando Alocação de Infraestrutura com Algoritmo Genético	33
4.1.1	Identificar Área de Comunicação	34
4.1.2	Descrição do Cenário	35
4.1.3	Algoritmo Genético	36
4.1.4	Avaliação da Alocação	40
4.1.5	Cálculo da Área de Cobertura	40
4.1.6	Caso de Uso: Sistema de Re-roteamento Inteligente	41
4.2	RP-Sim: Radio Propagation Simulator	45
4.2.1	Arquitetura	47
4.2.2	Implementação	49
4.2.3	Funcionalidades	50
4.2.4	Demonstração	51
4.3	Otimização de RSU com Diferentes Modelos de Propagação de Sinal	52
4.3.1	Identificar Área de Comunicação	53
4.3.2	Descrição do Cenário	55
4.3.3	Algoritmo Genético	55

4.3.4	Calculo da Área de Cobertura	56
4.3.5	Avaliação da Alocação	57
5	Considerações Finais e Trabalhos Futuros	59
5.1	Conclusões	59
5.2	Publicações	60
5.2.1	Publicações Como Primeiro Autor	60
5.2.2	Cooperação em Outros Trabalhos	60
5.3	Trabalhos Futuros	60
	Referências Bibliográficas	62
A	Trabalho Publicado no SBPO 2019	68

Capítulo 1

Introdução

Os avanços em tecnologias de informação e comunicação têm permitido que diversas áreas sejam beneficiadas, como a qualidade de vida e mobilidade urbana. As Redes Veiculares Ad hoc (*Vehicular Ad hoc Networks* - *VANETs*) são um exemplo de tecnologia que acompanham esses avanços. Com esse tipo de tecnologia apresentando grandes avanços é possível que aplicações e serviços inovadores sejam desenvolvidos, como, por exemplo, a recomendação de rotas seguras [38], rotas mais rápidas [56], busca por multimídia [40], detecção e gerenciamento de congestionamentos [50, 51], nuvens de processamento veicular [61], ou auxiliar com veículos autônomos [27]. As aplicações que tem como objetivo melhorar a mobilidade urbana são conhecidas como Sistemas de Transporte Inteligentes (ITS) [32].

É importante ressaltar que muitas dessas aplicações têm como base de seu funcionamento a capacidade de se conectar com outros veículos, obter e processar dados [32]. Esses dados podem estar relacionadas aos veículos, como velocidade e direção, ou podem ser de interesse dos motoristas, como vias congestionadas e localização de acidentes de trânsito, que podem estar disponíveis apenas em uma central. A utilização de infraestrutura pode ser feita para auxiliar todos esses estágios, desde facilitar a comunicação com veículos, até disponibilizar maior poder de processamento de dados.

O processamento pode ser realizado nas próprias *Roadside Units* - (RSUs), ou em servidores cloud que as utilizam para transmitir os dados para veículos [64, 52], reduzindo a carga de trabalho que é de responsabilidade destes. As RSUs são dispositivos preparados para VANETs, podendo ter acesso a rede e um alto poder computacional.

Além das limitações relacionadas a conexão e processamento, as aplicações podem ter requisitos mínimos de latência de comunicação, cobertura, confiabilidade e segurança [11]. O não atendimento desses requisitos mínimos pode resultar no mal funcionamento dessas aplicações ou piorar a mobilidade urbana, já que as aplicações podem funcionar incorretamente e com conhecimento incompleto da situação atual do tráfego. Aplicações de roteamento de veículos com conhecimento incompleto podem sugerir rotas congestionadas para os veículos [20].

Observando os requisitos das aplicações em ITS é possível entender a importância de realizar a alocação e utilização coordenadas das RSUs, para que eles sejam atendidos por meio de uma comunicação eficiente entre os veículos e a infraestrutura. Em destaque, a alocação coordenada das RSUs têm papel fundamental para o bom funcionamento

das aplicações que as utilizam. Durante a alocação das RSUs é importante considerar rigorosamente as propriedades de mapeamento da área de comunicação, pois quanto mais os resultados forem próximos ao cenário real, melhores soluções poderão ser encontradas para onde alocar cada uma das RSUs disponíveis.

Para melhorar o mapeamento das áreas de comunicação e a qualidade de sinal no cenário são destacadas duas características: Diferentes modelos de propagação de sinal e obstáculos presentes no cenário [54]. É importante considerar diferentes modelos de propagação de sinal, pois as configurações do cenário podem ser mais adequadas a um modelo específico, como é o caso de variação na potência de transmissão ou cenário urbano/rural. Os obstáculos têm uma grande influência no alcance do sinal e na área onde é possível a comunicação com as RSUs, principalmente em regiões urbanas. Essas regiões têm maior quantidade de casas, prédios e outras construções que podem reduzir a área efetiva de alcance do sinal [54].

Se a alocação das RSUs for feita sem utilizar as áreas com uma qualidade mínima de sinal, que são todas as áreas que recebem o sinal da RSU com uma potência mínima para se manter comunicação, diversos problemas, como a falta de conectividade, podem surgir para as aplicações. A capacidade da rede obter ou enviar dados fica comprometida, afetando as aplicações de maneira negativa. É comum encontrar trabalhos que consideram a área de comunicação de uma RSU sendo uma área com um raio fixo [62, 7]. Considerar um raio fixo em que as RSUs conseguem se comunicar mostra-se incoerente com o mundo real, pois diversos fatores, como distância total, obstáculos e tipo de ambiente, podem causar grande impacto na área de comunicação [25].

Uma alocação não coordenada das RSUs pode levar a situações em que certas regiões apresentem baixa qualidade de sinal. Esse tipo de resultado da alocação pode ocorrer por razão de mal planejamento na disposição das RSUs ou por super estimar a qualidade de sinal que chega em diferentes locais.

Em um primeiro momento, o principal problema acarretado pela alocação incorreta está relacionado com a falta de cobertura, mas as aplicações podem apresentar problemas secundários gerados por causa disso, como conhecimento incompleto da rede e impossibilidade de se comunicar com os nós necessários. Ao considerar uma distância máxima de comunicação, ou não considerar a perda do sinal causada por obstáculos, o estágio de alocação pode assumir que regiões têm uma comunicação ideal, mas, na realidade, o sinal que chega até determinada região é muito fraco para que haja comunicação. Além disso, outro possível problema é o desperdício de recursos, pois pode ser necessário utilizar mais RSUs para cobrir a mesma área, já que as posições selecionadas para a alocação garantem apenas uma cobertura reduzida, ou deixar de cobrir uma região por limitações financeiras, tendo em vista que o custo para implantar uma RSU ainda pode ultrapassar o valor de \$2.000,00 dólares [18]. Com limitações na quantidade de RSUs, utilizar as disponíveis para cobrir a maior área possível se torna um elemento essencial para que a maior quantidade de nós esteja dentro da área de cobertura.

Também foram propostos diferentes modelos de propagação de sinal que tentam ser mais realistas e próximos a propagação encontrada em ambientes urbanos, suburbanos ou rurais [4]. Utilizar esse tipo de informação para calcular a área de comunicação pode trazer ganhos de qualidade aos resultados. Diferentes modelos de propagação podem

calcular valores diferentes para o sinal recebido para distâncias similares, demonstrando a importância de considerar mais de um modelo de propagação.

Apesar da grande importância de se considerar o impacto que as construções têm na perda de sinal, ainda é uma tarefa de alta complexidade computacional definir esse impacto. A utilização de técnicas de *Ray Tracing*[29] podem ser feita em conjunto com informações sobre posição de antenas e obstáculos para calcular o impacto causado pelas construções no cenário. A utilização dessas técnicas, com diferentes modelos de propagação durante a definição da qualidade do sinal em cada região, gera resultados que podem ser utilizados para melhorar a qualidade dos dados utilizados durante o estágio de alocação das RSUs.

1.1 Motivação

A quantidade de veículos tem aumentado rapidamente nos grandes centros urbanos, como, por exemplo, a cidade de São Paulo, que teve um aumento de quase 2.5 milhões de veículos entre 2015 e 2019 [34]. Num futuro onde os veículos estarão conectados, a necessidade de uma infraestrutura adequada para lidar com a grande quantidade de requisições e cobrir regiões que podem ter menos veículos em horários específicos é essencial. Diversas aplicações podem se beneficiar de uma rede capaz de atender aos requisitos para um bom funcionamento. Algumas das aplicações que podem se beneficiar estão ligadas a saúde [8, 33], segurança [48, 41] e controle de congestionamento [50]. O tempo médio no trânsito utilizando transporte públicos em cidades como Brasília, São Paulo, Recife e Rio de Janeiro, já ultrapassa 120 minutos diários, considerando apenas o trajeto feito entre casa e trabalho [45]. A utilização de soluções ITS podem melhorar a mobilidade e eficiência desses cenários. O planejamento das infraestruturas com RSUs para VANETs têm se mostrado vantajosa, por se ter uma rede dedicada capaz de explorar as configurações de rede que favorecem as características das VANETs, como a alta mobilidade dos nós e topologia da rede.

A utilização de infraestrutura é um passo fundamental para alcançar os requisitos que as aplicações dependentes da rede possuem, permitindo que elas consigam funcionar corretamente. O estágio de alocação precisa ser feito de maneira precisa para permitir que tais requisitos sejam atendidos, selecionando os locais onde a cobertura é otimizada para a quantidade de RSUs disponíveis. Durante a alocação é importante que a qualidade do sinal em diferentes locais tenha sido definida corretamente e utilizada para definir quais são as áreas que conseguem se comunicar com a RSU, já que, se definidas incorretamente, pode gerar resultados que não representam a situação encontrada no mundo real. Devido ao preço elevado para alocar uma RSU, é importante utilizar das técnicas disponíveis, como algoritmos evolutivos [44], para aprimorar a utilização desse recurso.

1.2 Objetivos

O desenvolvimento deste trabalho teve como objetivo geral a utilização de algoritmos evolutivos para realizar a alocação de RSUs, considerando diferentes modelos de propagação

de sinal e ciente dos obstáculos.

Para alcançar o objetivo geral foram definidos os seguintes objetivos específicos:

1. *Definir locais viáveis de alocação de RSUs:* Os dados obtidos sobre as vias dos cenários precisam ser processados em busca dos cruzamentos, já que as posições destes serão utilizadas para definir a área de comunicação das RSUs. Os dados obtidos estão desorganizados e, muitas vezes, existem dezenas de localidades que são consideradas cruzamentos em distâncias muito próximas, podendo dificultar o uso. Esses cruzamentos são utilizados em razão de fornecerem posições nas quais a área de comunicação é maior caso uma RSU seja alocada nele. Para que o uso desses dados seja possível é necessário realizar um pré-processamento.
2. *Determinar área de cobertura:* Utilizar as ferramentas e simuladores disponíveis para determinar a área de comunicação das RSUs considerando diferentes modelos de propagação e obstáculos no cenário.
3. *Desenvolver ferramenta para calcular a área de comunicação:* Desenvolver uma ferramenta para auxiliar no mapeamento da área de comunicação considerando diferentes modelos de propagação e ciente dos obstáculos presentes no cenário. Com essa ferramenta a definição das áreas cobertas pelas RSUs se torna mais rápida e fácil.
4. *Alocar RSUs:* Utilizar algoritmo genético para definir o local de alocação de diferentes quantidades de RSUs nos cenários escolhidos. Maximizar a área coberta e utilizar esse parâmetro para definir a qualidade das soluções encontradas.
5. *Diferentes modelos de propagação de sinal:* Realizar a alocação das RSUs com diferentes modelos de propagação de sinal e avaliar a adequação desses modelos.

1.3 Estrutura da Dissertação

Esta dissertação está dividida em 5 Capítulos. O Capítulo 2 apresenta a fundamentação teórica e as ferramentas que foram utilizadas. O Capítulo 3 apresenta e discute os trabalhos relacionados. O capítulo 4 apresenta as abordagens utilizadas para alcançar os objetivos e a ferramenta que foi desenvolvida para auxiliar nesse processo. O Capítulo 5 apresenta as considerações finais e os trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste Capítulo são apresentados os conceitos fundamentais e as ferramentas utilizadas para o desenvolvimento deste trabalho, divididos em 4 seções. A Seção 2.1 apresenta conceitos básicos sobre otimização e apresenta o algoritmo genético, que foi a abordagem utilizada. A Seção 2.2 introduz o tema sobre redes veiculares e discute algumas características dos padrões utilizados. A Seção 2.3 inicia a discussão sobre propagação de sinal sem fio e o impacto causado por construções. A seção 2.4 apresenta as ferramentas que foram utilizadas.

2.1 Otimização Combinatória

A otimização combinatória tem como objetivo encontrar um solução de menor ou maior custo dentre um conjunto de soluções, que apesar de ser finito, pode ser grande demais para avaliar todas as possíveis soluções.

Uma das maneiras de resolver problemas de otimização combinatória é enumerar todas as soluções possíveis e selecionar a melhor. O problema com esse tipo de abordagem, apesar dela sempre encontrar a melhor solução possível, é que, para muitos problemas, o grande número de possibilidades de soluções torna o tempo gasto para calcular as soluções impraticável [19, 5].

Dentro da área de otimização combinatória existem diferentes abordagens para tratar de problemas onde existem soluções demais para que seja calculada a melhor delas. Neste trabalho é utilizada uma classe de algoritmo chamada de Algoritmo Genético (GA), que é uma abordagem heurística [53].

Os GAs são um tipo de algoritmos bio inspirados [37], baseados em heurísticas, que utilizam mecanismos de seleção natural e genética para encontrar soluções [30]. GAs também são algoritmos evolucionários, dessa forma, as soluções encontradas vão melhorando a cada iteração, até chegar no critério de parada, que é o critério usado para determinar até quando o algoritmo deve continuar buscando soluções [37].

A Figura 2.1 apresenta o funcionamento de um GA de forma genérica. É possível ver que o primeiro passo é a geração de uma nova população, formada por indivíduos completamente aleatórios. Após esse primeiro passo é iniciado um ciclo que contempla avaliar o *score*, checar critério de parada e gerar novos indivíduos [37].

O *score* é uma métrica calculada para cada elemento e indica o quão bem uma determinada solução resolve o problema enfrentado. Uma das funções do *score* é selecionar qual a solução final após o término das execuções, que são interrompidas no passo de checagem do critério de parada. O critério de parada pode ser definido de diferentes maneiras, já que diferentes problemas podem ser abordados com critérios de parada diferentes. Os jeitos mais comuns de definir o critério de parada são: definir um número máximo de iterações ou definir um valor a ser atingido pelo *score* [37].

Os processos de seleção, *crossover* e inserção de elementos mutantes, são responsáveis por evoluir as soluções encontradas determinado momento. A seleção é responsável por ordenar os elementos pelo *score* e definir os grupos de elite e descarte. O grupo de elite é utilizado durante o *crossover*, gerando elementos que são baseados nos elementos elite e elementos mutantes ou elementos aleatórios.

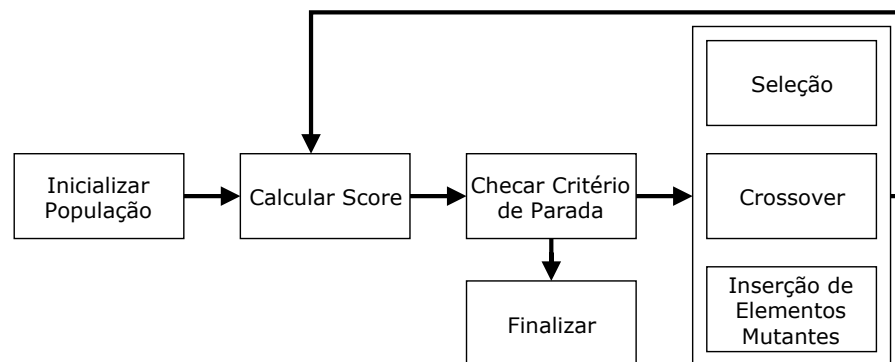


Figura 2.1: Algoritmo Genético

Uma vez que o critério de parada é atingido, os elementos da última geração são ordenados, utilizando o *score*, e o primeiro elemento é selecionado como solução para o problema. As soluções que são encontradas utilizando esse tipo de abordagem podem não ser soluções ótimas, mas geralmente são boas o suficiente para diversas áreas e problemas.

2.2 Redes Veiculares (VANETs)

As VANETs têm apresentado grandes avanços, principalmente por permitir o desenvolvimento de aplicações que podem revolucionar a maneira como vivemos [38, 21]. As VANETs são um tipo de Redes Móveis Ad-Hoc (Mobile Ad-Hoc Networks - MANETs) onde veículos e equipamentos ao longo das vias são equipados com tecnologia de comunicação sem fio, processamento e armazenamento [6].

Para que este tipo de rede funcione, os veículos precisam ter uma *On Board Unit* (OBU), que é um dispositivo capaz de trocar mensagens com outras OBUs e com RSUs. RSUs são dispositivos estáticos que estão posicionados ao longo das vias [3].

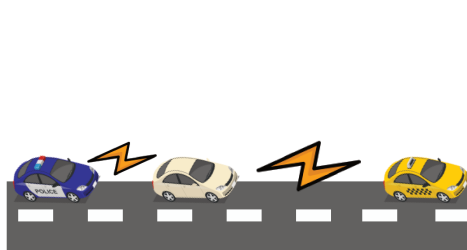
As aplicações em VANETs podem ser classificadas em três categorias: segurança; entretenimento e assistência ao motorista [3]. Aplicações relacionadas a segurança podem trabalhar em diversos aspectos, como alertar veículos próximos sobre acidentes [60], até mesmo evitar rotas que são consideradas perigosas [38]. A transferência de vídeos

utilizando VANETs pode ser considerada uma das aplicações desenvolvidas para entretenimento [35]. Em aplicações ligadas a assistência ao motorista, podemos citar, encontrar melhores rotas [2].

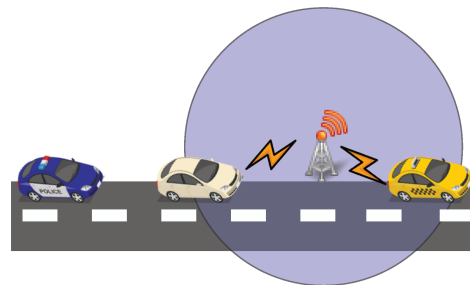
Os 4 principais tipos de comunicação em VANETs são descritos a seguir:

- **Veículo-para-Veículo (V2V):** Na comunicação V2V, apenas a comunicação ad-hoc acontece, com os veículos se comunicando exclusivamente entre si [3, 17].
- **Veículos-Para-Infraestrutura (V2I):** A comunicação V2I é o modo estruturado das VANETs, no qual os veículos conseguem se comunicar apenas com a infraestrutura disponível [3, 17].
- **Comunicação Híbrida (V2H):** A combinação da comunicação V2V e V2I, na qual veículos e infraestrutura podem se comunicar [3, 17].
- **Veículo-para-Tudo (V2X):** Esse é um tipo de comunicação mais recente, na qual os veículos conseguem se comunicar com outros dispositivos ao seu redor, desde celulares dos pedestres até infraestrutura [15].

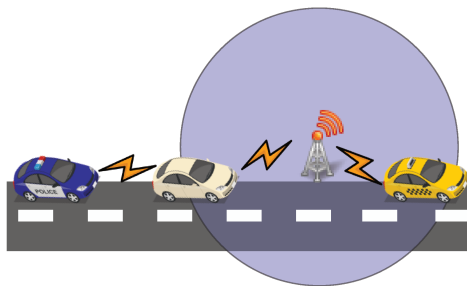
A Figura 2.2 apresenta um exemplo de comunicação dos quatro tipos de comunicação dentro de VANETs.



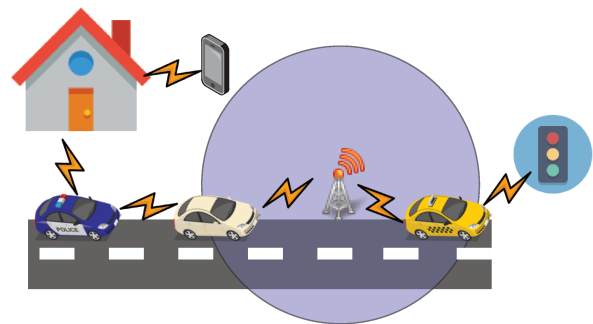
(a) Comunicação V2V



(b) Comunicação V2I



(c) Comunicação V2H



(d) Comunicação V2X

Figura 2.2: Tipos de comunicação em VANETs

As VANETs têm algumas características que já são bem definidas, relacionadas à mobilidade, energia, densidade e poder computacional. Essas características estão apresentadas abaixo:

- **Mobilidade:** A mobilidade dos nós em VANETs está ligada diretamente com a topologia do cenário, já que os veículos precisam estar sobre as ruas para se movimentarem. Além disso, a alta velocidade dos nós gera cenários desafiadores para as aplicações que utilizam VANETs, já que os nós podem estar em contato por pouco tempo um com o outro [3, 58].
- **Energia:** Atualmente a energia não é um fator limitante, como em outros tipos de MANETs, pois o veículo pode fornecer energia para os equipamentos de comunicação, processamento e armazenamento [36].
- **Densidade:** A densidade de veículos pode variar de acordo com a região e com o horário. Horários de picos, como final de tarde, podem tornar vias congestionadas e grande quantidade de veículos próximos [3, 58].
- **Poder Computacional:** Como em redes VANETs os nós são veículos sem restrições energéticas, eles podem estar equipados com diferentes tipos de sensores, como, por exemplo, GPS, câmeras e equipamentos de alto poder de processamento [17].

As VANETs trabalham dentro da frequência 5.9 GHz para comunicação *Dedicated Short Range Communication* (DSRC), com uma taxa de dados de 3 a 27MB/s [17]. Essas características, associadas a alta mobilidade e densidade de veículos, podem impor limites espaciais e temporais para a troca de mensagens entre os dispositivos na rede.

2.3 Propagação de Sinal sem Fio

A perda de sinal em redes sem fio é resultado de diferentes fatores, tais como distância ou absorção atmosférica, os quais podem ter influência na área de comunicação de um dispositivo. Para auxiliar nos cálculos da propagação de sinal foram propostos diferentes modelos ao longo do tempo, mas considerar apenas a propagação fornecida por esses modelos tem se mostrado ineficiente para aplicações que consideram os dados gerados por eles ou aplicações baseadas em diferentes configurações de antenas, já que considerar apenas estes modelos pode levar a uma determinação errada da área de comunicação.

Um fator que pode exercer uma grande influência na qualidade do sinal são as construções, tais como prédios e casas, entre os dispositivos que estão se comunicando, o que tenta ser simulado de maneira genérica por alguns modelos de propagação que tem variáveis relacionadas ao tipo de ambiente. Essas aproximações são valores que podem ser definidos durante a parametrização dos modelos e que aproximam o resultado da degradação obtida em diferentes cenários, como urbanos e rurais, de forma genérica.

O problema de se considerar apenas a degradação calculada por esses modelos é que, dispositivos que estão a mesma distância um do outro, sofrem a mesma perda, mesmo que em algum desses dispositivos existam diversos obstáculos entre eles. A degradação calculada nos modelos será a mesma, independentemente do sinal se propagar por um campo

vazio ou por ruas com vários obstáculos. Basicamente, existem dois tipos de modelos: Modelos que assumem que a perda de sinal será sempre igual para uma determinada distância e modelos que são estocásticos.

Os modelos estocásticos consideram um valor aleatório para influenciar o sinal, mas a potência que chega até o receptor tende a se aproximar de um determinado valor fixo.

Existem diferentes modelos para calcular a perda de sinal sofrida por redes sem fio. Alguns destes modelos são mais próximos dos ideais teóricos (*e.g.*, Free-space), enquanto outros são modelos empíricos que tentam se aproximar da degradação sofrida por meio de múltiplas medições de sinal em determinados ambientes, como, por exemplo, o cenário urbano (*e.g.*, Egli). Este trabalho explorou as diferenças e o uso dos seguintes modelos: FreeSpace, Egli, Ericsson e SUI. O modelo *TwoRayGround* não foi considerado neste trabalho, pois não funciona bem para as distâncias avaliadas [55].

A Figura 2.3 apresenta a potência de sinal recebido em diferentes distâncias, considerando diferentes modelos de perda de sinal. Neste caso, o transmissor tem uma potência de 30db, sendo que nenhum outro ganho ou perda foi considerado. A altura do transmissor é de 30 metros e a altura do receptor é de 1,5 metros. É possível notar um comportamento de rápido declínio na qualidade de sinal de até 250 metros nos modelos comparados. A linha contínua vermelha representa a sensibilidade de recepção de -101 dBw, que é um valor comum em diferentes dispositivos, o que significa que valores inferiores a ela não conseguem se comunicar.

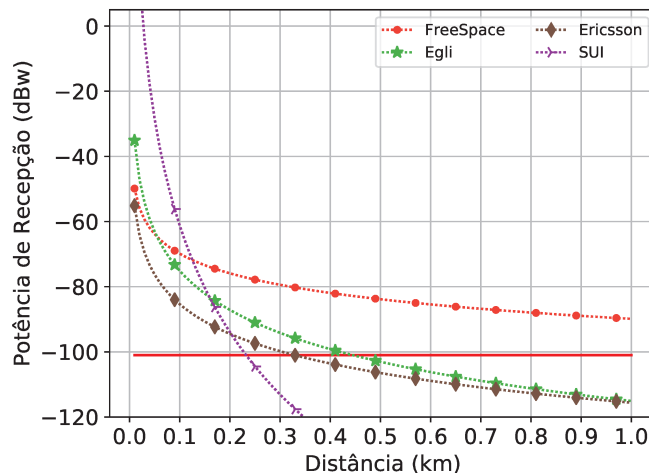


Figura 2.3: Perda de sinal

2.3.1 FreeSpace

A atenuação de espaço livre proposta em [26] é aceita pelo Setor de Normatização das Telecomunicações ITU (ITU-T) como a perda fundamental teórica de uma onda radioelétrica¹. Este modelo idealiza a propagação de uma onda eletromagnética, que resulta em pouca precisão em um cenário real. No entanto, sua baixa complexidade permite computar rapidamente a perda mínima de sinal. Com base na recomendação ITU P.525,

¹<https://www.itu.int/rec/R-REC-P.525/en>

a perda do espaço livre, considerando antenas isotrópica, está representada na equação 2.1.

$$L_{bf}[dB] = -32,45 + 20 \cdot \log_{10}(F_{[Mhz]}) + 20 \cdot \log_{10}(d_{[Km]}) \quad (2.1)$$

onde:

d : distância

f : frequência

2.3.2 Egli

Um dos primeiros modelos empíricos foi proposto em [24], baseado em medidas de perda de sinal nas cidades de Nova Iorque e Nova Jersey. Esse modelo considera a perda de sinal por árvores, construções e irregularidades do terreno de forma genérica. A fórmula simplificada, considerando a altura da antena de recepção em 1.5 metros, está representada na equação 2.2 [63].

$$L_{Egli}[dB] = 139.1 - 20 \cdot \log_{10}(hTx_{[m]}) + 40 \cdot \log_{10}(d_{[Km]}) \quad (2.2)$$

onde:

hTx : altura da antena transmissora

2.3.3 Ericsson

O modelo Ericsson [14] é uma versão melhorada do modelo Hata [47], que permite modificar os parâmetros de acordo com o tipo de cenário (*i.e.*, urbano, suburbano e rural). A perda de sinal com esse modelo é calculada com a equação 2.3.

$$L_e = A_0 + A_1 \cdot \log_{10}(d) + A_2 \cdot \log_{10}(hTx) + A_3 \cdot \log_{10}(hTx) \cdot \log_{10}(d) - 3.2 \cdot ((\log_{10}(11.75 \cdot TRx))^2 + g(f)) \quad (2.3)$$

onde:

$$g(f) = 44.49 \cdot \log_{10}(f) - 4.78 \cdot (\log_{10}(f))^2$$

hTx = Altura da antena de transmissão

TRx = Altura da antena de recepção

d = Distancia em kilometros

Como esse trabalho focou em cenários urbanos, os valores de A_0 , A_1 , A_2 e A_3 foram respectivamente 36.2, 30.2, 12.0 e 0.1 [24].

2.3.4 Stanford University Interim - SUI

O modelo de propagação SUI é um modelo proposto pela Universidade de Stanford [1]. O modelo SUI abrange três categorias de terrenos. A categoria 1 tem a maior perda, adequado para ambientes montanhosos com alta densidades de árvores. A categoria 2 é adequada para terrenos planos ou áreas montanhosas com pouca vegetação. A categoria

3 apresenta baixa perda de sinal, útil para terreno plano. O cálculo para esse modelo é apresentado na equação 2.4.

$$L_{sui} = A + 10\gamma \log_{10}(d_{[m]}/100) + X_f + X_h + s \quad (2.4)$$

onde:

$$\gamma = a - b * hTx + (c/hTx)$$

$a = 4.0$, valor definido de acordo com o terreno.

$b = 0.0065$, valor definido de acordo com o terreno.

$c = 17.1$, valor definido de acordo com o terreno.

hTx = Altura da antena de transmissão

onde s está entre 8.2 e 10.6 de acordo com a quantidade de árvores no cenário. O valor de X_f é o fator de correção para a frequência de operação e o valor de X_h é o fator de correção para a altura da antena.

2.3.5 Impacto dos Obstáculos na Propagação do Sinal

Na Figura 2.4 é possível visualizar a diferença de áreas de comunicação quando se comparam regiões com características distintas. As Figuras 2.4a e 2.4b são regiões urbanas com diversas construções e as Figuras 2.4c e 2.4d são regiões rurais com extenso espaço aberto. Além disso, também é possível ver a diferença que a propagação de sinal tem quando são considerados os obstáculos nas Figuras 2.4b e 2.4d.

A diferença da propagação de sinal reflete diretamente na área de comunicação, que é definida como regiões com potência de sinal maior que -74db. Quando se utiliza o modelo FreeSpace de propagação de sinal e os obstáculos são considerados, a área de comunicação é de 294000m² para áreas rurais e 17400m² para áreas urbanas, uma diferença de 94% na área total.

O modelo de propagação de sinal escolhido para representar o impacto dos obstáculos é o mesmo que é utilizado pelo Veins², que é um simulador de rede veicular de código aberto. O modelo é resultado de uma série de medições empíricas que foram feitas com o objetivo de avaliar o impacto causado por obstáculos urbanos em VANETs [54]. A fórmula utilizada para calcular a degradação causada pelas construções é descrita na equação 2.5.

$$D_{obst} = distObst * 0.63 + N_{cortes} * 2.3 \quad (2.5)$$

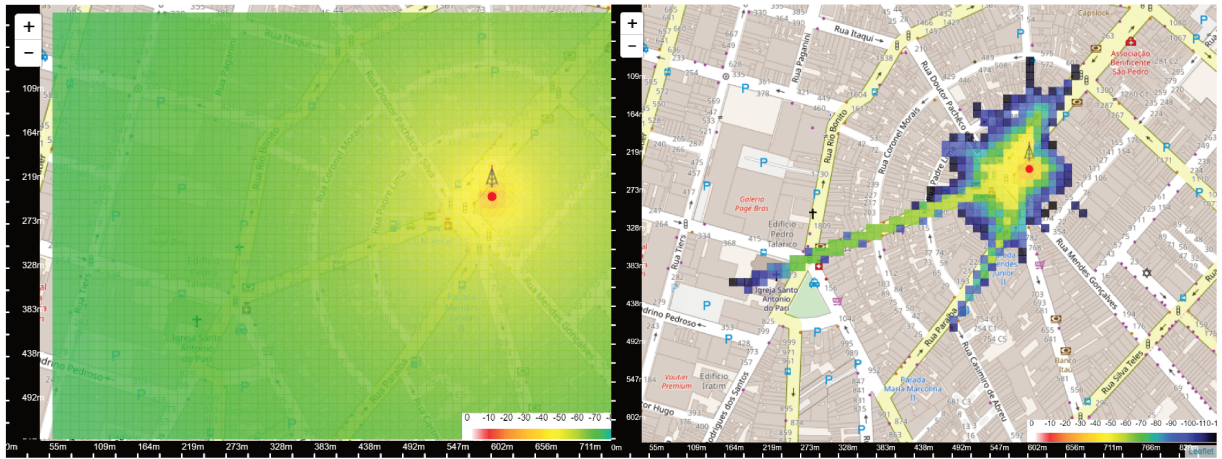
D_{obst} = Degradação causada por obstáculos

$distObst$ = Distância que o sinal percorreu dentro de obstáculos

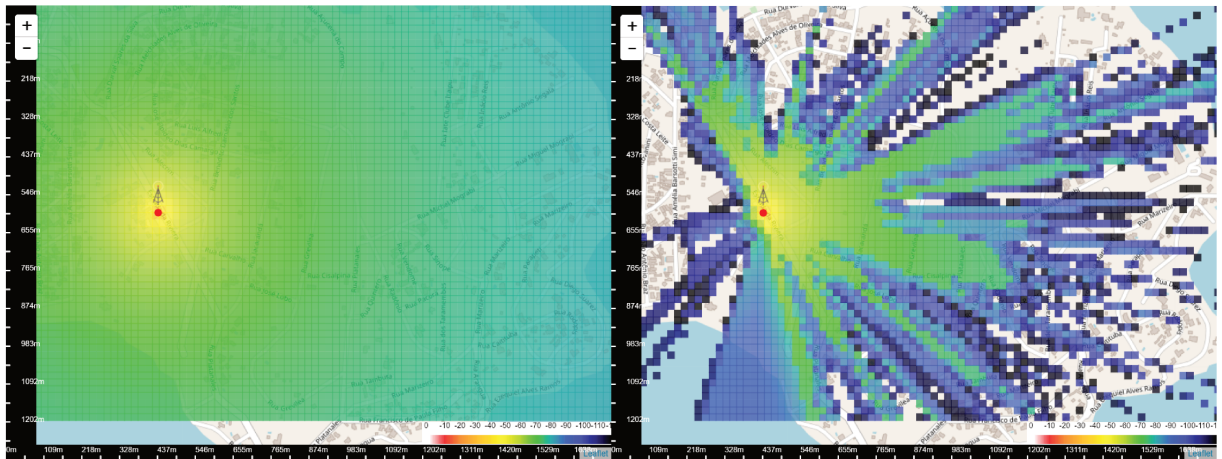
N_{cortes} = Número de vezes que o sinal atravessou algum obstáculo

A perda de sinal causada por obstáculos é uma grande limitadora da área de comunicação de um dispositivo, principalmente dentro de centros urbanos. A Figura 2.5 mostra o impacto causado por obstáculos em dois gráficos distintos. A Figura 2.5a mostra a degradação sofrida quando se considera a perda por obstáculos e o modelo FreeSpace atuando

²<https://veins.car2x.org/>



(a) Região urbana sem considerar obstáculos (b) Região urbana considerando obstáculos



(c) Região rural sem considerar obstáculos (d) Região rural considerando obstáculos

Figura 2.4: Propagação do sinal

em conjunto. Nessa Figura, a cada 25 metros que são percorridos dentro de obstáculo, é considerado que o número de cortes, ou seja, número de vezes que o sinal atravessou um obstáculo, aumenta em 1. É possível ver que em poucos metros dentro de obstáculos a degradação é a mesma que centenas de metros ao ar livre. Com o objetivo de facilitar a visualização foi adicionado um plano que representa onde a degradação atinge 100dB.

A Figura 2.5b apresenta os mesmos modelos utilizados na Figura 2.5a, mas agora separados em duas linhas em um gráfico 2D. Nessa figura é mais fácil ver que o impacto causado por obstáculos tem um impacto muito maior e mais rápido do que o modelo FreeSpace. Foi destacado uma linha contínua onde a degradação atinge 100dB, no eixo Y do gráfico.

É possível notar que a degradação causada por obstáculos reduz rapidamente a potência do sinal que está sendo transmitido. Ao associar isso com outros modelos, como FreeSpace ou SUI, torna-se fácil visualizar que a área de comunicação terá seu tamanho reduzido. Apesar de diversos fatores influenciarem nas métricas de comunicação, aplicações que não consideram esses dados podem ter seu resultado divergido mais rapidamente no mundo real.

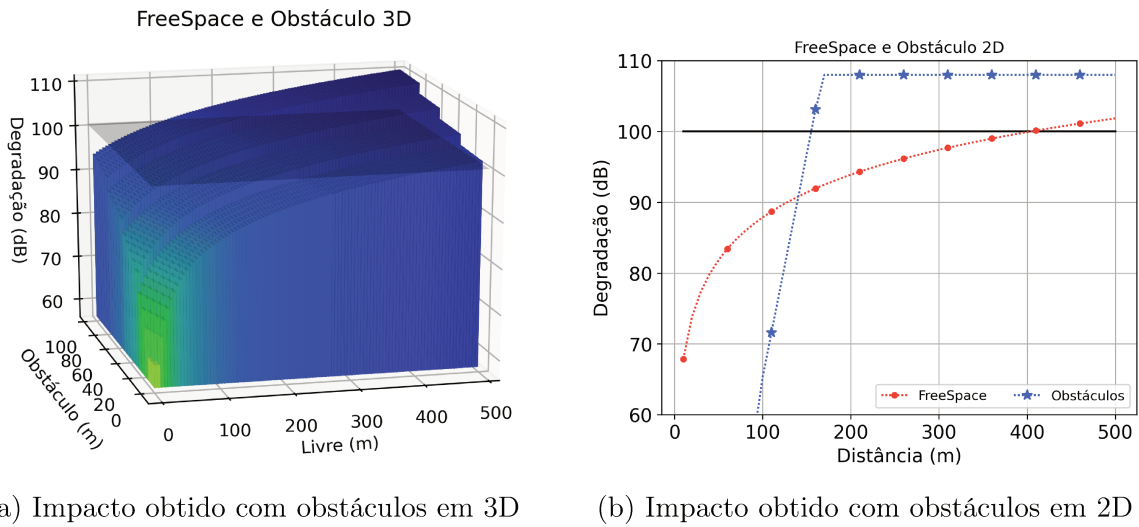


Figura 2.5: Comparação entre o impacto de obstáculos e o modelo FreeSpace

2.4 Ferramentas Utilizadas

As ferramentas que foram utilizadas durante o desenvolvimento deste trabalho estão descritas nas Seções 2.4.1, 2.4.2, 2.4.3 e 2.4.4.

2.4.1 OpenStreetMaps

O OpenStreetMaps³ é uma ferramenta de mapas online, classificada como Sistema de Informações Geográficas (GIS). Os dados que são empregados por essa ferramenta são obtidos de forma colaborativa, onde diversas pessoas podem sugerir edições [16]. A comunidade de usuários, ao encontrar incoerência com a realidade, pode fazer suas próprias edições, dentre alterar nome ou formato de ruas até marcar estabelecimentos e construções que ainda não foram inseridos.

Essa ferramenta também oferece a opção de exportar dados sobre as vias e construções, que são os dados utilizados durante este trabalho. Para exportar esses dados é possível utilizar uma ferramenta de seleção de área de interesse e definir quais dados devem ser exportados.

A Figura 2.6 apresenta uma visão geral do OpenStreetMaps onde é possível ter uma ideia da quantidade de dados disponíveis.

2.4.2 Simulator of Urban Mobility - SUMO

A plataforma de simulação SUMO⁴ consiste em um pacote com vários programas que permitem a simulação do tráfego em cenários gerados artificialmente ou baseados no mundo real. Também estão presentes ferramentas relacionadas à visualização; encontrar rotas; cálculo de emissões de gases e outras funcionalidades. Os dados exportados do

³© OpenStreetMap contributors - <https://www.openstreetmap.org>

⁴dlr.de/ts/sumo

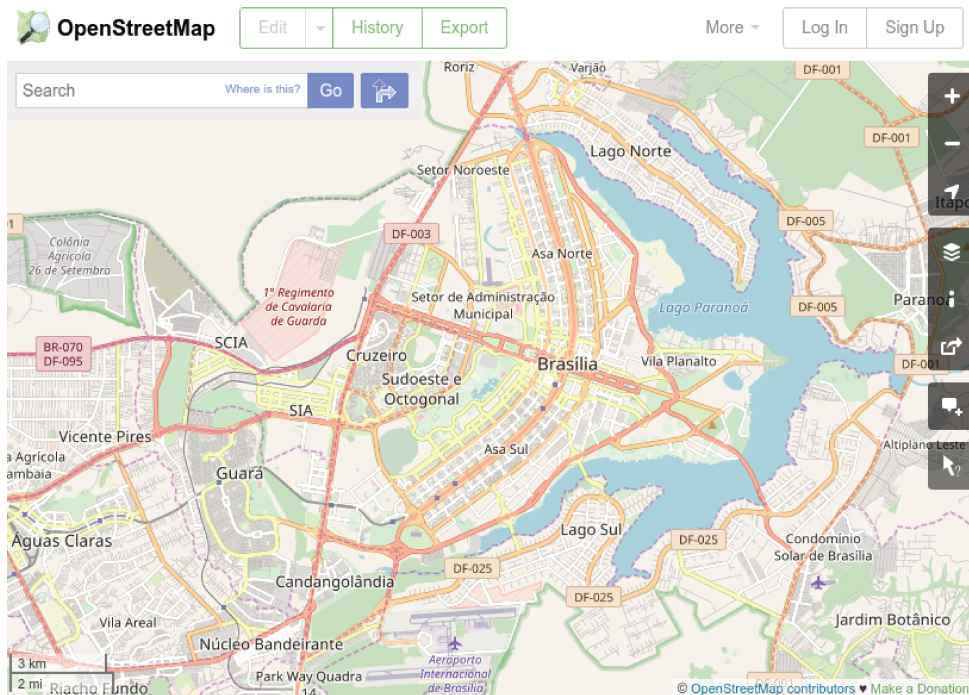


Figura 2.6: OpenStreetMaps

OpenStreetMaps podem ser convertidos, usando uma ferramenta presente no pacote do SUMO, para que seja possível utilizá-los em conjunto. Dessa forma, as simulações podem ser mais próximas do mundo real, já que, agora, vários dados são baseados no mundo real. Também é possível gerar cenários artificialmente, completamente aleatórios ou seguindo algum tipo de padrão, como um padrão em GRID, que são ruas retangulares ou quadradas, reduzindo a acurácia da simulação com o mundo real.

Com o SUMO é possível testar diversas soluções e ter uma prévia dos resultados de forma mais simples, rápida e barata, do que implementá-las no mundo real. São comuns soluções envolvendo algoritmos de roteamento de veículos, controle de semáforos e realizar a avaliação de mudanças na infraestrutura. Também estão disponíveis diversos relatórios dentro do SUMO, como distância percorrida e emissão de gases dos veículos.

Durante as simulações é possível interagir com elementos como semáforos e veículos, mostrando a granularidade das interações que são possíveis. Assim, surge a possibilidade de que os desenvolvedores foquem mais em suas aplicações e percam menos tempo criando esses elementos.

A Figura 2.7 apresenta a interface principal do SUMO. É possível notar que parte de um cenário real foi carregado, evidenciando as vias de uma rodovia e um bairro residencial. Este cenário foi exportado do OpenStreetMaps e convertido usando uma das ferramentas que acompanha o SUMO, o *netconvert*. Após essa conversão é possível carregá-lo dentro do SUMO e utilizar um arquivo de rotas para iniciar a simulação.

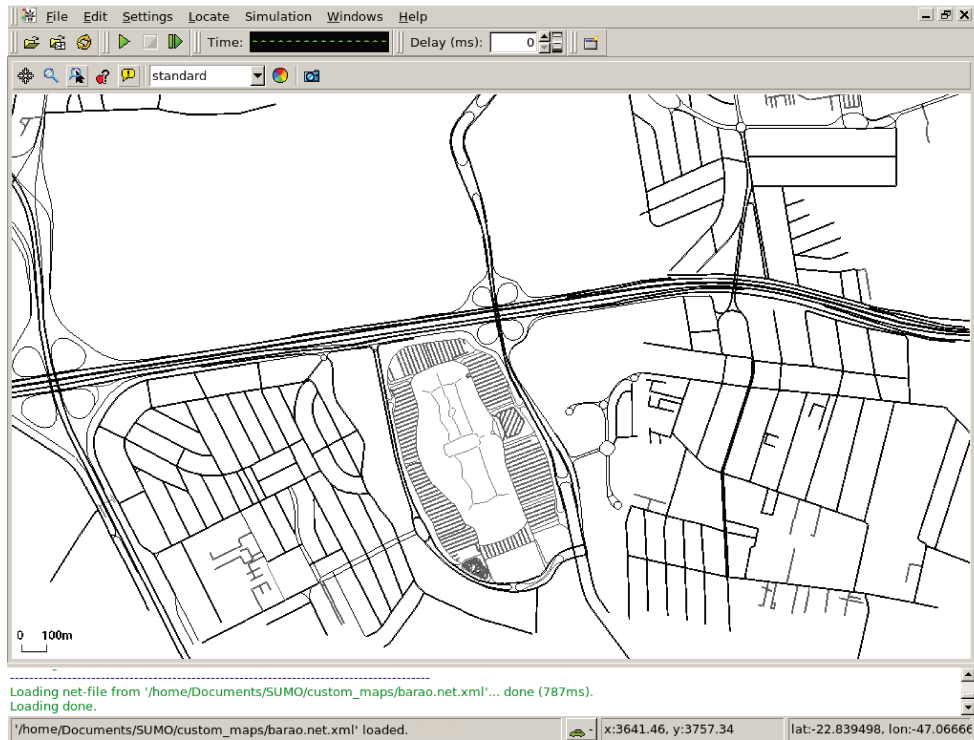


Figura 2.7: Simulator of Urban MObility

2.4.3 OMNeT++

O OMNeT++⁵ é um simulador de eventos discretos, de uso livre para fins não lucrativos, que pode ser acoplado ao SUMO. Este simulador é feito em C++ e tem como objetivo a simulação de eventos discretos para trabalhar com comunicação de redes, multiprocessadores, sistemas distribuídos ou sistemas em paralelo [59]. O OMNeT++ está sendo usado em diversas áreas distintas e está sendo amplamente adotado em áreas de simulação de redes sem fio ou ad-hoc [59].

Algumas das principais características presentes no OMNeT++ são: capacidade de simulações em larga escala, facilidade no desenvolvimento e encontrar problemas, modularidade e fácil integração com SUMO [59] e Veins. Essas características tornam o desenvolvimento do modelo e análise dos resultados uma tarefa mais rápida.

2.4.4 VEINS

O VEINS⁶ é um framework de código aberto que simula redes veiculares. O VEINS roda em conjunto com o simulador de eventos discretos OMNeT++ e com o simulador de tráfego SUMO para permitir que a comunicação no padrão 802.11p aconteça.

É possível, utilizando o VEINS, fazer alterações durante a execução de uma simulação para interagir com os pacotes que são enviados ou recebidos na rede pelos dispositivos, como RSUs e veículos. É possível considerar modelos de propagação de sinal pré definidos, como *FreeSpace* ou *Simple Obstacle Shadowing*, para saber quando a comunicação pode

⁵<https://www.omnetpp.org/>

⁶<https://veins.car2x.org/>

ser feita e qual sua qualidade.

Do mesmo jeito que o OMNeT++, o VEINS foi projetado para ser escalável com diferentes tamanhos de simulação. Para permitir maior velocidade, pode-se utilizar o VEINS em clusters. É possível utilizar dados sobre construções, exportados do OpenStreetMaps, para melhorar a precisão da propagação de sinal no VEINS, quando o modelo utilizado considera esse tipo de dado. A Figura 2.8 mostra a interação simplificada do VEINS com o OMNeT e com o SUMO.

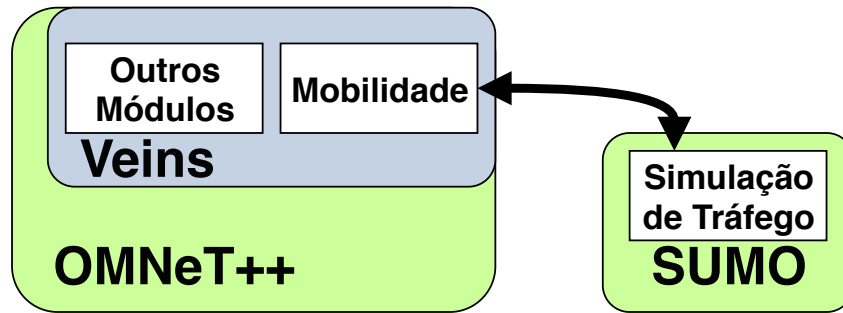


Figura 2.8: VEINS

Com a utilização do VEINS é possível obter métricas relacionadas a comunicação dentro do cenário, tais como pacotes transmitidos ou perdidos.

Capítulo 3

Trabalhos Relacionados

Neste capítulo são apresentados trabalhos científicos que abordam o problema de alocação de RSUs. Um processo que muitos trabalhos utilizam é transformar o cenário em grafo, convertendo os cruzamentos em vértices e as ruas em arestas. Esse processo facilita a utilização de alguns algoritmos e por vezes pode permitir um processamento mais rápido. Durante esse processo abre-se mão de precisão nos cálculos de cobertura, já que um vértice ou uma aresta pode estar coberto ou não. Com a utilização de polígonos, para representar as áreas cobertas e o cenário, pode-se mapear com maior precisão as regiões cobertas.

O trabalho proposto pelos autores Xue, Yang e Dong [62] aborda diretamente o problema de cobertura máxima limitado pela conectividade utilizando RSUs. Os autores mapeiam as ruas em segmentos que representam *clusters* de conectividade, onde *clusters* adjacentes conseguem comunicar-se. Dessa maneira, as RSUs conseguem comunicar-se enviando mensagem através desses *clusters* nas vias. O trabalho tem como objetivo maximizar a conectividade entre RSUs e veículos, repetindo restrições relacionados custo de implantação e a necessidade de cobertura mínima de cada link. Para resolver este problema é utilizada uma técnica de *linear scalarization*. Durante o estágio de simulação, as RSUs tiveram suas áreas de comunicação definidas com alcance máximo de 100 a 500 metros e foi utilizado o modelo de propagação de sinal *TwoRayGround*.

Definir a distância máxima que as RSUs podem se comunicar gera diversos outros problemas, já que essas áreas muitas vezes são impraticáveis no mundo real. Apesar do trabalho ter sido desenvolvido para regiões urbanas, obstáculos como regiões arborizadas e construções podem exercer um forte impacto na distância máxima de comunicação, resultando em uma distância pequena para regiões com mais obstáculos e um grande alcance para regiões com linha de visão. Esse trabalho utilizou o modelo de propagação de sinal *TwoRayGround*[55], que, nas distâncias avaliadas, é normalmente convertido no modelo *FreeSpace*. O cenário é convertido para um grafo para que seja possível trabalhar com ele.

A abordagem aplicada pelos autores Barrachina, Garrido, Fogue, Martinez, Cano, Calafate e Manzoni [7] utiliza informações referentes à densidade dos veículos para definir onde alocar as RSUs, tentando reduzir o tempo que um aviso de emergência leva para chegar ao seu destino e a quantidade de saltos necessários para tanto. A política D-RSU, que foi definida no trabalho, determina que mais RSUs sejam alocadas em regiões com menos veículos, garantindo melhor conectividade nessas áreas. O trabalho assume que, em

regiões mais densas, os veículos podem se beneficiar da comunicação V2V mais facilmente. A área de comunicação de uma RSU foi definida como distância máxima de 400 metros e foi justificada pelo trabalho por utilizar a frequência de 2.4 GHz para a comunicação V2I. Esse trabalho também faz a avaliação de uma alocação onde as RSUs foram distribuídas de maneira uniforme, ou seja, os dispositivos foram alocados de uma maneira fixa entre elas e que é possível a comunicação entre RSUs.

O trabalho desenvolvido por Barrachina também limita a área de comunicação máxima, assumindo que tudo que está até 400 metros consegue se comunicar sem problemas, independente de obstáculos e outras influências. A alocação olha exclusivamente para a quantidade de veículos nas regiões, podendo permitir que alguma região fique sem RSU por ter mais veículos. Esse tipo de abordagem abre caminho para problemas de cobertura em horários com menos veículos.

O trabalho feito por Seif, Taoufik e Sadok [10] também aborda diretamente o problema de cobertura, tentando minimizar a quantidade de RSUs para reduzir o custo de alocação e maximizar a cobertura. Para isso, o trabalho utiliza informações históricas sobre a mobilidade dos veículos para definir padrões de mobilidade, para definir onde alocar as RSUs e para calcular a cobertura obtida. As vias e cruzamentos do cenário são mapeados como um hipergrafo, que é usado para definir os padrões de mobilidade e cobertura, alocando uma RSU em um determinado vértice. Uma vez que as RSUs tenham sido alocadas, é utilizado SUMO e OMNeT para simular a solução proposta e determinar a cobertura.

A utilização de um hipergrafo resulta nos mesmos problemas que são encontrados quando um grafo é utilizado. A identificação das vias mais utilizadas é feita para determinar onde é necessário alocar uma RSU ou não, mas esse tipo de abordagem pode selecionar ruas que não são ideais, já que horários de pico podem influenciar nessa avaliação.

A utilização de algoritmos evolucionários também foi feita pelos autores Moura, Cabral, Sales e Aquino no trabalho [46], que utiliza um algoritmo genético para realizar a alocação das RSUs, modelando o problema como um problema de cobertura com limite temporal. O cenário utilizado foi mapeado como grafo e foi feito um passo de pré-processamento utilizando a métrica de centralidade *betweenness* e métodos de detecção de comunidade. A detecção de comunidades é utilizada para reduzir o espaço de busca e a métrica de centralidade *betweenness* é utilizada para ordenar as comunidades. Após esses estágios, o algoritmo genético é iniciado para determinar onde as RSUs devem ser alocadas. A cobertura é determinada pela quantidade de veículos que têm cobertura no cenário utilizado. A utilização de um grafo auxiliou na execução da métrica de centralidade e possibilitou que as soluções pudessem ser encontradas rapidamente, mas abrindo mão de precisão por conta das duas abordagens utilizadas.

O trabalho desenvolvido por Cavalcante, Aquino, Pappa e Loureiro [13] também modela o problema enfrentado como um problema de cobertura com limite temporal e utiliza um algoritmo genético para solucioná-lo. São alocadas k RSUs com raio de comunicação R e, para realizar os cálculos, o cenário é representado como um grafo em que os dispositivos podem ser alocados nos vértices. Os problemas de limitar a distância máxima de comunicação das RSUs e de modelar o problema como um grafo também estão presentes.

Apesar de não realizar a alocação diretamente, um modelo de propagação de sinal foi desenvolvido de forma empírica pelos autores Sommer, Eckhoff, German e Dressler para representar a atenuação causada por obstáculos urbanos, como prédios e outras construções, no trabalho [54]. Esse modelo é utilizado em simuladores, como o Veins, para definir o impacto que os obstáculos têm na comunicação entre os dispositivos. Esse modelo foi mapeado utilizando VANETs.

Os autores Cao, Cui, Zhang, Jiang e Wang [12] lidam com o problema de alocação de RSUs utilizando um modelo matemático e um algoritmo *branch and bound*. As posições candidatas para alocar uma RSU nesse trabalho foram definidas a partir da topologia da rede e da mobilidade dos veículos. Foi utilizado um conjunto de dados com mais de 780 milhões de registros, contendo dados referente a táxis da cidade escolhida. O problema foi abordado como uma otimização multi objetivo, reduzindo o atrasado de comunicação entre veículos e RSU e os custos de implantação. O trabalho utiliza o algoritmo K vizinhos mais próximos para determinar quais posições candidatas podem ser removidas, reduzindo o espaço de busca por soluções.

O cenário foi modelado como um grafo onde os vértices são as intersecções e as arestas são as ruas, sendo que o peso da aresta é o comprimento da rua. Esse trabalho considera apenas o *delay* entre a comunicação dos dispositivos, para isso, leva em consideração a distância da RSU, assumindo que ela possui uma distância de comunicação máxima fixa.

Os autores Guerna e Bitam [31] apresentam uma abordagem evolutiva para realizar a alocação das RSUs, em que o objetivo é posicionar estes dispositivos em intersecções com alta prioridade dos veículos. Este trabalho tenta maximizar a cobertura das RSUs enquanto minimiza a área de sobreposição e quantidade de dispositivos utilizado. A área de comunicação das RSUs é fixada a uma distância máxima de 250 metros. Para determinar a área coberta por uma RSU é checado quais intersecções estão no máximo 250 metros de um determinado dispositivo. Esse tipo de cálculo de área coberta pode piorar ainda mais o problema de precisão da área coberta pelas RSUs, já que as vias podem ter sua cobertura prejudicada em relação ao mundo real.

Apesar do tema de alocação de RSUs ser relevante há vários anos, muitos trabalhos ainda usam abordagens que podem ser problemáticas em diversas situações. A Tabela 3.1 apresenta os trabalhos que foram discutidos nessa Seção, sendo possível observar que, mesmo com a evolução da área, os problemas abordados nem sempre são considerados durante a alocação e alguns trabalhos lidam com eles de forma pontual. As métricas avaliadas na tabela 3.1 estão discutidas abaixo:

- **Mapeamento de cobertura:** Refere-se à definir previamente a área em que os dispositivos conseguem se comunicar com as RSUs, considerando informações como modelo de propagação e obstáculos. Trabalhos que consideram um raio fixo recebem 'Não' nessa métrica. O Não atendimento dessa métrica pode refletir em resultados que não condizem com o cenário real, encontrando soluções que são impraticáveis.
- **Obstáculos considerados:** Refere-se à trabalhos que consideram os obstáculos para definir a área de comunicação e simulam essa comunicação após a alocação. Desconsiderar os obstáculos pode causar grandes diferenças, principalmente em cenários urbanos, entre a comunicação estimada e a comunicação obtida no mundo

real.

- **Modelos de Propagação:** Refere-se à trabalhos que consideram múltiplos modelos de propagação de sinal para calcular a comunicação entre dispositivos e definir a área de comunicação. Adequar os cálculos da propagação de sinal auxiliar a aproximar a propagação calculada pelos simuladores da propagação obtida no mundo real, já que muitas vezes utilizar modelos que são ideais teóricos pode ser insuficiente.
- **Cobertura não binária:** Refere-se à trabalhos que conseguem considerar as vias que estão parcialmente cobertas e não apenas as que estão 0% ou 100% cobertas. As áreas são avaliadas deste modo podem acabar gerando resultados que superestimam ou subestimam a cobertura das RSUs, por isso modelos que tem a capacidade de interpretar esse dado é tão importante.
- **Cobertura global:** Refere-se à trabalhos que realizam a cobertura global do cenário, não apenas de uma região ou de um horário específico. Modelos que conseguem bons resultados locais muitas vezes são insuficientes para o cenário global, já que solucionar o problema com soluções específicas pode ter um resultado médio inferior do que uma solução global.

	[54] 2011	[13] 2012	[7] 2013	[62] 2016	[46] 2018	[12] 2018	[31] 2018	[10] 2020	Solução Proposta
Mapeamento de cobertura	Não	Não	Não	Não	Não	Não	Não	Não	Sim
Obstáculos considerados	Sim	Não	Não	Não	Não	Não	Não	Não	Sim
Modelos de Propagação	Não	Não	Não	Não	Não	Não	Não	Não	Sim
Cobertura não binária	Não	Não	Não	Não	Não	Não	Não	Não	Sim
Cobertura global	Não	Sim	Não	Não	Sim	Sim	Sim	Não	Sim

Tabela 3.1: Comparação de trabalhos relacionados

Capítulo 4

Alocação de RoadSide Units

Neste capítulo serão apresentadas as abordagens utilizadas para alcançar os objetivos pré-definidos e, também, a metodologia e os resultados obtidos em cada abordagem. Além disso, será apresentada a ferramenta desenvolvida para auxiliar este trabalho e pesquisas futuras.

A seção 4.1 apresenta uma alocação de RSUs em que a área de cobertura de cada uma delas foi mapeada utilizando simuladores e a alocação foi feita com um algoritmo genético. Os pontos críticos e dificuldades dessa abordagem levantaram a necessidade de uma ferramenta, que seja capaz de auxiliar o processo de mapeamento da área de cobertura.

A seção 4.2 apresenta a ferramenta que foi desenvolvida com o objetivo de auxiliar o cálculo e visualização da propagação do sinal sem fio. É apresentado, também, os modelos de propagação, modulação e diversas das configurações que já estão disponíveis para serem utilizadas.

E por fim, a seção 4.3 apresenta uma alocação de RSUs em que as áreas de cobertura foram calculadas utilizando a ferramenta proposta. Uma vez que as áreas de cobertura estão calculadas é feito a alocação com um algoritmo genético, onde os resultados obtidos são apresentados.

4.1 Otimizando Alocação de Infraestrutura com Algoritmo Genético

Otimizando Alocação de Infraestrutura com Algoritmo Genético (ORC) é apresentado neste capítulo. A utilização de simuladores para definir a área de comunicação e o funcionamento do GA, também serão apresentados e discutidos. Por fim, os resultados obtidos são apresentados na seção 4.1.6.

Nessa seção, o Algoritmo Genético (GA) que foi utilizado durante a alocação das RSUs, é uma adaptação de uma versão já validada desse tipo de algoritmo. A principal diferença está em como o genoma é montado, utilizado e alterado para parar de conter posições de latitude e longitude separadamente e passar a conter objetos com posições que pré definidas. Em sua versão original, apresentada no Anexo A, o GA é utilizado para realizar a alocação de bases do corpo de bombeiros na cidade de Chicago, usando como

área que podia ser alcançada a partir de uma sede em um limite de tempo. O objetivo dessa alocação é maximizar a área coberta pelo corpo de bombeiros. É possível observar a versatilidade do algoritmo desenvolvido, podendo encontrar soluções para diferentes cenários onde exista um problema sobre maximização da área coberta.

Na versão atual a sede do corpo de bombeiros foi trocada por uma RSU e a área que pode ser alcançada pelo corpo de bombeiros a partir destas sedes foi transformada em áreas com sinal mínimo para se comunicar com as RSUs do cenário.

Suposições: todas as interseções das vias foram definidas como possíveis candidatas para alocar uma RSU, selecionando, assim, posições que podem ter uma linha de visão maior com mais locais. O sinal sofre influência por obstáculos urbanos, como prédios, casas, construções, entre outros. O modelo *simple obstacle shadowing* [54] foi usado em conjunto com o modelo *FreeSpace* para calcular a propagação do sinal nos simuladores. Esses modelos foram escolhidos por já serem adotados em diversos outros trabalhos da literatura. A qualidade da comunicação é medida usando o indicador de intensidade do sinal recebido (RSSI).

4.1.1 Identificar Área de Comunicação

Como já foi discutido anteriormente, a região em que os veículos conseguem se comunicar com um determinado dispositivo é afetada por diversos fatores, por exemplo, distância e construções presentes no cenário. É importante levar em consideração esses fatores para determinar a área de comunicação, para aproximar os resultados obtidos do que é encontrado no mundo real.

Para identificar a área de comunicação de cada RSU, é feito uma simulação utilizando as ferramentas descritas na seção 2.4. Nessa simulação um veículo percorre todas as vias do cenário a 6km/h, transmitindo no canal de controle com a frequência de 10 Hz. Além do veículo percorrendo as vias, uma RSU está posicionada em cada local candidato, já definido nas suposições.

As RSUs têm a função de receber os *beacons* enviados pelos veículos e salvar a posição de onde o sinal foi enviado e o RSSI recebido. Depois que a simulação é finalizada, todas as posições com RSSI menores que -94 db são descartadas, regiões com uma qualidade mínima de sinal maior que -94 db.

Esses locais são transformados em um polígono usando o algoritmo *concave hull*, resultando em um polígono que cobre todas as posições em que algum veículo foi capaz de se comunicar com a RSU. Cada RSU possui o seu próprio polígono que representa sua área de comunicação. Como esse polígono pode ter problemas para cobrir uma rua inteira, uma vez que a simulação retorna as posições extremamente precisas por ser uma simulação computacional, é criado um segundo polígono, que representa uma versão inflada do primeiro, onde o segundo contém completamente o primeiro com uma margem de N metros. Após testes, foi definido que o valor de 10 metros para N é adequado ao cenário escolhido. Com esse novo polígono, que é uma versão maior do primeiro, é possível ter margem suficiente para cobrir todas as faixas das vias que o polígono original cobria.

A Figura 4.1 mostra os resultados neste processo, na qual é possível ver a diferença da área de comunicação quando aplica-se o processo descrito variando-se apenas o RSSI

utilizado. É possível notar que, quanto menor o valor do RSSI, maior a área de comunicação. Também é possível notar que em direções com mais construções e obstáculos, a qualidade do sinal diminui mais rapidamente do que em regiões com maior linha de visão, o que influencia no formato e área desse polígono.

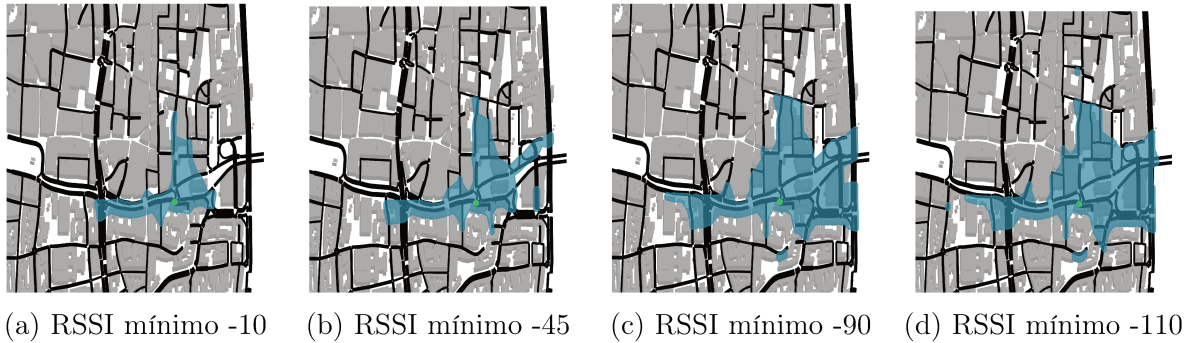


Figura 4.1: Área de comunicação com diferentes RSSI

4.1.2 Descrição do Cenário

Nessa seção são apresentadas as informações acerca do cenário utilizado para realizar a avaliação da proposta. As versões das ferramentas utilizadas também são apresentadas.

Para avaliar a otimização proposta em um cenário realista, um subcenário de TAPAS-Cologne¹ foi utilizado. TAPASCologne é um projeto de código aberto que fornece um conjunto de dados em larga escala com alto realismo para simulação de veículos urbanos com base no SUMO. Ele usa um mapa realista da cidade de Colônia, Alemanha, obtido através do OpenStreetMap. A Tabela 4.1 apresenta os parâmetros e valores utilizados durante as avaliações.

Tabela 4.1: Parâmetros da Execução

Parâmetro	Valor
Área utilizada	4.6 km ²
Cenário	TAPASCologne
Área	(12200,13000) (14500,15000)
Simulador de mobilidade	SUMO 0.32.0
Simulador de rede veicular	Veins 4.7.1
Simulador de evento discreto	OMNeT++ 5.3
Potência de transmissão	0.98 mW
Sensibilidade do receptor	-110.0 db
Rssi mínimo considerado	-94.0 db

A Figura 4.2 mostra o cenário usado durante esse processo. As cruzes com cor laranja representam os possíveis locais para alocar RSU, sendo que os locais com alta densidade de RSUs geralmente são locais com muitas interseções. As linhas pretas representam as vias e os polígonos cinza são construções dessa região. O cenário selecionado tem 2,3 km de largura e 2,0 km de altura, sendo localizado na parte central da cidade.

¹<https://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne>

A área de comunicação que foi apresentada na Seção 4.1.1 é calculada para todos os locais candidatos, ou seja, para cada uma das cruzes que estão presentes na Figura 4.2.

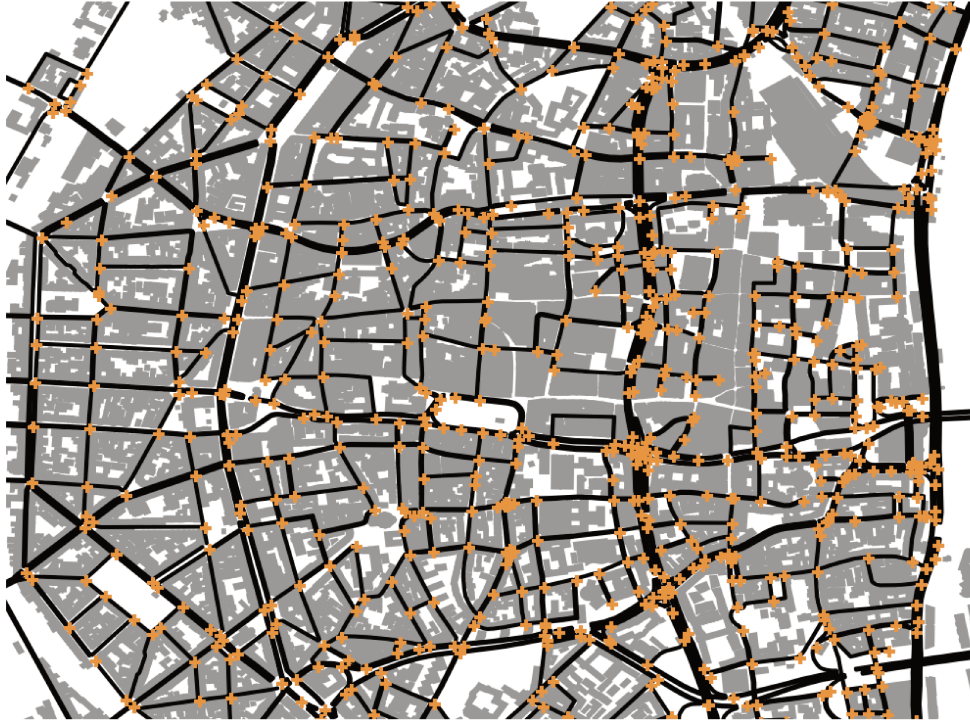


Figura 4.2: Cenário utilizado

4.1.3 Algoritmo Genético

Uma vez que a área de comunicação com cada RSU está definida, é possível selecionar quais locais candidatos devem receber um dispositivo para maximizar a cobertura. Algoritmos evolutivos já se mostraram eficientes para esse objetivo, por isso, esse trabalho adotou a utilização de um algoritmo genético adaptado para solucionar o problema de selecionar a posição para alocar K RSUs maximizando a cobertura.

As adaptações que foram feitas para o funcionamento do algoritmo genético com os dados e objetivos desse trabalho estão descritos a seguir:

- **Genoma:** O genoma nesta versão tem tamanho fixo e é constituído por uma lista de valores do tipo float, que tem associado a cada valor um objeto com outras informações.
 - Mutação adaptada para o genoma do tipo float.
 - Herança adaptada para o genoma do tipo float.
 - Crossover adaptado para o genoma do tipo float.
- **Recursos disponíveis:** A quantidade de recursos, agora, passa a ser fixa para cada execução, permitindo que a solução apenas tente maximizar a área coberta com os recursos disponíveis e a geração de uma curva de pareto.

Um genoma do tipo float permite que ele seja ordenado com facilidade, enquanto tem, em cada valor, outras informações associadas. Uma vez ordenado, os primeiro K valores podem ser selecionados e as informações associadas a eles utilizadas para definir onde alocar as RSUs.

Para iniciar o GA, são fornecidos como entrada a quantidade de RSUs que estão sendo alocadas, as áreas que os dispositivos conseguem se comunicar com cada RSU e a área onde a cobertura deve ser maximizada. A configuração usada pelo GA pode ser vista na Tabela 4.2.

Tabela 4.2: Configurações do GA

Parâmetro	Valor
Seleção	Baseada em Rank
Codificação	Float
Tamanho da população	200
Inserção de Mutação	Novos Elementos
Limite de iteração	300
Quantidade de RSUs Alocadas	[1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
Quantidade de melhores elementos	10%
Quantidade de elementos aleatórios	20%
Quantidade de elementos herdados	70%
Porcentagem de genoma herdado	80%/20%

O genoma é decodificado para definir quais locais serão selecionados para alocar uma RSU. Esse processo é realizado da seguinte maneira: um local é atribuído a cada valor do genoma e o genoma é ordenado com base nesse valor. O local atribuído aos primeiros elementos são usados para calcular a cobertura. A quantidade de locais selecionados varia de acordo com a entrada fornecida no início do processo.

O Algoritmo 1 mostra como o algoritmo genético foi implementado em sua versão atual e como o *score* é calculado, que é a métrica usada para classificar as soluções encontradas durante sua execução.

A primeira geração criada é composta por um conjunto de elementos com um genoma completamente aleatório. Assim que essa geração está completa, o loop condicional principal pode ser iniciado. O critério de parada foi definido como 300 gerações, porque isso foi suficiente para as soluções convergirem para um bom resultado para este experimento.

As linhas 10 a 20 descrevem como o *score* é calculado para cada elemento. A área que precisa ser coberta é definida na Linha 10 e tem seu tamanho em metros quadrados salvos na linha 11. A linha 14 é responsável pelo cálculo da área de interseção da área coberta pelas RSUs, que é a sobreposição.

O loop condicional na Linha 15 calcula a diferença entre a área que precisa ser coberta e as áreas cobertas pelas RSUs desse elemento. Isso resultará em uma área menor, contendo apenas as áreas que ainda não foram cobertas por nenhuma RSU. Na Linha 19 é calculado o valor normalizado de quanto é coberto, selecionando este conjunto de RSUs.

Depois que todos os elementos têm seu score calculado, uma nova geração é formada na Linha 21. Se o critério de parada for atingido, o melhor elemento definido até agora é retornado na Linha 22. Isso encerra a execução do GA.

A criação de uma nova geração é uma etapa importante para o GA, pois é onde acontece a evolução e busca por melhores resultados. Neste trabalho, essa criação é

Algoritmo 1: GA e calculo do score

```

1  Entrada
2  areaParaCobrir  $\leftarrow$  Polígono contendo o formato das ruas no cenário
3  areasDeComunicacao  $\leftarrow$  Área de comunicação das RSUs
4  totalRSU  $\leftarrow$  Quantidade total de RSUs
5  Saída
6  Melhor solução encontrada
7  início
8      geração  $\leftarrow$  criarPrimeiraGeracao(totalRSU)
9      enquanto não (criterioParado.satisfeito) faça
10         para elemento in geração faça
11             areaParaCobrirAtual  $\leftarrow$  areaParaCobrir.copy()
12             areaTotalInicial  $\leftarrow$  areaParaCobrirAtual.areaInM2
13             poligonos  $\leftarrow$  getPoligonos(elemento.genoma, areasDeComunicacao)
14             n  $\leftarrow$  poligonos.size
15             sobreposição  $\leftarrow \sum_{i=1}^n \sum_{j=i+1}^n \text{sobreposio}(\text{poligonos}_i, \text{poligonos}_j)$ 
16             enquanto poligonos  $\neq \emptyset$  faça
17                 areaParaCobrirAtual  $\leftarrow$  areaParaCobrirAtual-poligonos[0]
18                 poligonos.popFirst()
19             areaFinal  $\leftarrow$  areaParaCobrirAtual.areaEmM2
20             elemento.score  $\leftarrow$  areaFinal/areaTotalInicial
21             elemento.sobreposição  $\leftarrow$  sobreposição
22         geração  $\leftarrow$  criarNovaGeracao(geração)
23     melhorSolucaoEncontrada  $\leftarrow$  getMelhorElemento(geração)

```

responsável por inserir a variabilidade genética nas soluções, usando elementos aleatórios. Os melhores elementos são replicados para a próxima geração, garantindo que as melhores soluções encontradas até o momento permaneçam até o final do processo. O processo de geração da nova leva de elementos é descrito no Algoritmo 2.

Os parâmetros necessários são: (i) os elementos da geração atual; (ii) a porcentagem de quanto manter dos melhores elementos e (iii) o quanto inserir de elementos completamente aleatórios. A linha 9 ordena os elementos da geração, para garantir que os melhores elementos estejam no início do vetor. O loop na linha 12 adiciona os melhores elementos a um vetor separado. O loop na Linha 17 é responsável por inserir a quantidade correta de elementos aleatórios.

Depois que os melhores elementos são separados e os elementos aleatórios são criados e inseridos, a nova geração pode ser iniciada, mesclando os melhores elementos até o momento e os elementos aleatórios, na linha 20. O loop na linha 22 copia para o vetor "subLasts" todos os elementos da geração atual que não estão entre os melhores, com os elementos aleatórios, que é importante para a próxima etapa.

Após a criação dos vetores anteriores, o loop na linha 25 seleciona um elemento dos melhores elementos da geração atual e um elemento dos "subLats", contendo um elemento completamente aleatório ou um elemento que não esta entre os melhores. Esses dois elementos terão seus genomas fundidos para criar um novo elemento, com a maior parte do genoma vindo do elemento que foi definido como sendo um dos melhores até o momento. O novo elemento criado nesta parte será adicionado na nova geração na linha 29, repetindo esse processo até que as gerações tenham o mesmo tamanho. Ao final, na linha 30, a geração atual receberá a geração que foi criada durante essas etapas.

Quando o GA alcança o critério de parada, o resultado pode ser decodificado a partir do genoma usando as etapas descritas anteriormente, ordenando o genoma. Uma amostra do resultado é apresentada na Figura 4.3, onde os edifícios não são exibidos, mas foram

Algoritmo 2: Criação de nova geração

```

1  Entrada
2   $geracaoAtual \leftarrow$  A geração do último loop
3   $bestElementsReplica \leftarrow 0.1$ 
4   $randomElementsInsertion \leftarrow 0.2$ 
5  Saída
6  Nova geração de elementos
7  início
8     $novaGeracao \leftarrow []$ ,  $subBests \leftarrow []$ ,  $subAleatorios \leftarrow []$ 
9     $geracaoAtual.ordenaPorScore()$ 
10    $tamanho \leftarrow geracaoAtual.tamanho * bestElementsReplica$ 
11    $i \leftarrow 0$ 
12   enquanto  $i < tamanho$  faça
13      $subBests.add(geracaoAtual[i])$ 
14      $i \leftarrow i + 1$ 
15    $tamanho \leftarrow geracaoAtual.tamanho * randomElementsInsertion$ 
16    $i \leftarrow 0$ 
17   enquanto  $i < tamanho$  faça
18      $subAleatorios.add(new RandomElement())$ 
19      $i \leftarrow i + 1$ 
20    $novaGeracao \leftarrow subBests \cup subAleatorios$ 
21    $i \leftarrow geracaoAtual.tamanho * bestElementsReplica$ 
22   enquanto  $i < geracaoAtual.tamanho$  faça
23      $subLasts.add(geracaoAtual[i])$ 
24    $subLasts \leftarrow subAleatorios \cup subLasts$ 
25   enquanto  $novaGeracao.tamanho < geracaoAtual.tamanho$  faça
26      $element1 \leftarrow subAleatorios.random()$ 
27      $element2 \leftarrow subLasts.random()$ 
28      $element \leftarrow new Element(element1, element2)$ 
29      $novaGeracao.add(element)$ 
30    $geracaoAtual \leftarrow novaGeracao$ 

```

usados para alcançar esses resultados. As linhas pretas representam as vias do cenário, o polígono azul representa a área coberta pelas RSUs selecionadas e os círculos verdes são os locais onde uma RSU foi alocada.



Figura 4.3: Exemplo de resultado obtido com 8 RSUS

4.1.4 Avaliação da Alocação

O GA foi usado para alocar as RSUs, variando o número de RSUs em cada execução. A Figura 4.4 mostra o resultado comparando a área de cobertura e informações relativas a sobreposição em cada uma dessas execuções. A linha vertical exibe o intervalo de confiança para cada resultado, que foi calculado com 95% de confiança para 33 execuções.

As áreas em que os dispositivos conseguem se comunicar com alguma RSU aumentam em proporção logarítmica, o que ocorre devido à uma seleção de menor quantidade de RSUs é possível selecionar locais com maior impacto na cobertura total e a área de sobreposição tende a ser menor, já que menos RSUs estão sendo alocadas. Conforme há o aumento da quantidade de RSUs sendo alocadas, a sobreposição pode aumentar e os locais candidatos com maior impacto na cobertura já foram selecionados. Esse comportamento indica que o GA já selecionou bons locais candidatos para alocar as RSUs desde o princípio.

É importante ressaltar que o conjunto de locais selecionados para uma determinada quantidade k RSUs sendo alocadas, não necessariamente faz parte do conjunto selecionado para alocações com mais RSUs, ou seja, os locais candidatos selecionados para alocar um conjunto de 4 RSUs podem não fazer parte dos locais candidatos ao alocar 8 RSUs. Isso acontece porque com quantidades diferentes sendo alocadas diferentes disposições podem ser mais vantajosas do que repetir as que já foram encontradas anteriormente.

A sobreposição apresentada na Figura 4.4 é a média da sobreposição para cada RSU na solução encontrada. Com poucas RSUs no cenário, a sobreposição é próxima de 0, o que significa que as RSUs selecionadas estão longe o suficiente para não cobrir a mesma área mais de uma vez. Assim que a seleção começa a ser feita com mais RSUs, esse valor tende a ficar estável, perto de 10% para os conjuntos avaliados.

Também é possível ver na Figura 4.4 a cobertura média por RSU. A área total coberta pode ser obtida multiplicando a média coberta pela quantidade de RSU. O comportamento encontrado suporta as informações apresentadas até agora, que utilizando menos RSUs nos cenários os locais com maior cobertura são escolhidos primeiro.

4.1.5 Cálculo da Área de Cobertura

Para realizar o cálculo da área de cobertura foi selecionada uma abordagem que utiliza a diferença de polígonos. Dessa forma é possível definir a área que deve ser coberta como um polígono e a área coberta pelas RSUs como outro polígono. Com essas informações é possível calcular a diferença entre esses polígonos, sendo que o polígono restante é a área que ainda não foi coberta por nenhuma RSU.

Como nesse estágio do trabalho os simuladores são utilizados para determinar a área de cobertura das RSUs, a área que deve ser coberta foi definida como vias da área de interesse. Para que isso fosse possível, as vias do cenário foram mapeadas e convertidas em um polígono com o mesmo formato e tamanho. A Figura 4.5 apresenta um exemplo do funcionamento desse cálculo.

No estágio A, as ruas, formas pretas e amarelas, foram mapeadas em polígonos. No estágio B, os polígonos da área de comunicação das RSUs foram adicionadas, representadas pelos círculos azuis. E, por último, foi calculada a diferença entre eles, sobrando apenas a área sem cobertura ainda.

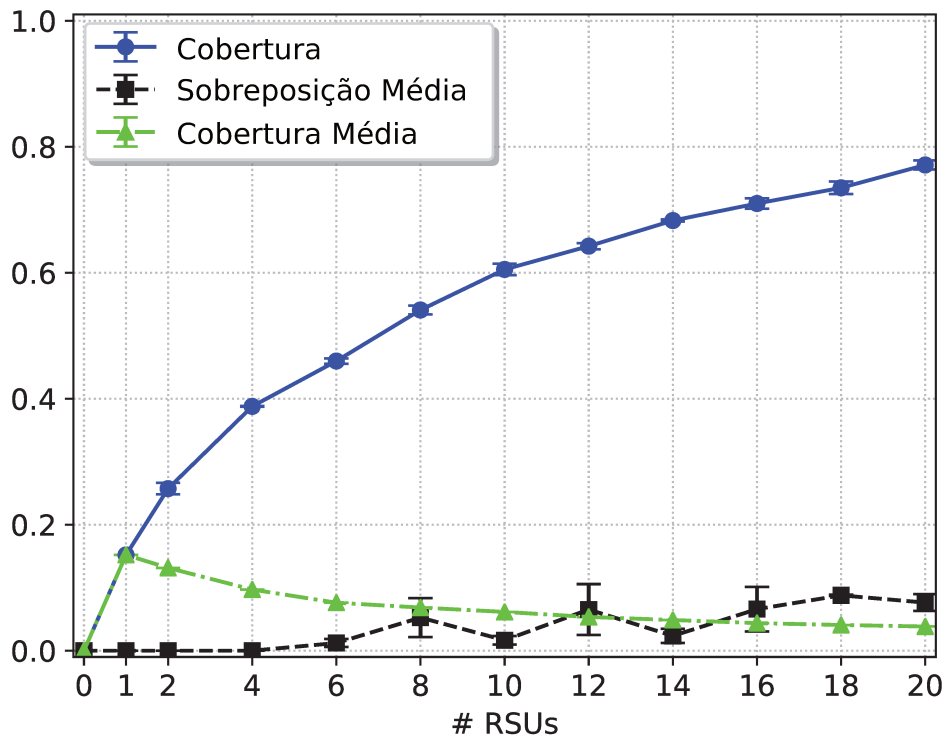


Figura 4.4: Comparação entre porcentagens de cobertura e sobreposição

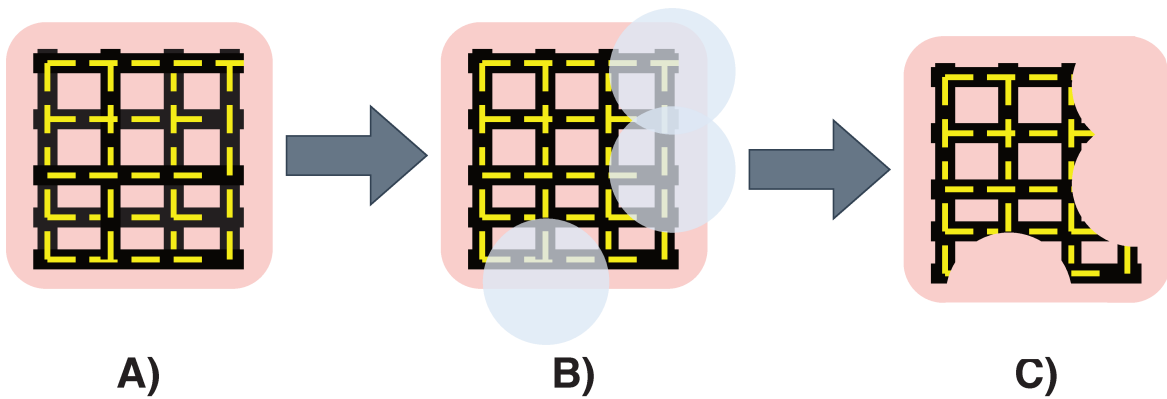


Figura 4.5: Funcionamento do calculo da cobertura

Após esse processo estar completo, o valor normalizado da área final dividido pela área inicial representa a porcentagem que foi coberta pelas RSUs, então, o número 1 subtraído esse valor representa a porcentagem que ainda falta ser coberta por algum dispositivo.

4.1.6 Caso de Uso: Sistema de Re-roteamento Inteligente

O re-roteamento do tráfego de veículos é uma das principais soluções para melhorar a mobilidade urbana e a experiência de condução [39]. Dessa forma, vários sistemas de re-roteamento inteligentes de tráfego foram propostos [20, 49, 23]. Esses sistemas se concentram em sugerir rotas mais rápidas para os veículos com base em uma visualização de tráfego atualizada (i.e., informação do tráfego global que descreve a condição atuali-

zada de cada via), devido à requisitos rigorosos relacionados à confiabilidade e precisão da visualização do tráfego. Grande parte dessas soluções são baseadas em ter uma infraestrutura presente para que seja possível alcançar os requisitos mínimos, já que soluções sem infraestrutura podem não ter a performance desejada [22]. O serviço de re-roteamento utilizado tem requisitos mínimos relacionados, principalmente, com a cobertura e atraso, já que não conseguir alcançá-lo pode resultar em problemas na tomada de decisão.

No entanto, essas soluções não levam em consideração a degradação do sinal produzida por obstáculos urbanos, como prédios, casas e outras construções, durante a alocação das RSUs. Essa degradação limita potencialmente a comunicação entre veículos e infraestrutura, o que, consequentemente, reduz drasticamente a precisão do tráfego e prejudica o desempenho geral do sistema [39]. Portanto, otimizar a alocação de cada RSU, considerando a propagação do sinal com esses obstáculos, é essencial para maximizar a cobertura do sistema e, consequentemente, preparar o caminho para alcançar o desempenho desejado.

Com esse caso de uso, queremos avaliar o impacto do posicionamento da RSU e a degradação do sinal causada por obstáculos urbanos em um sistema inteligente de re-roteamento de tráfego. Assim, realizamos a alocação das RSUs do sistema de re-roteamento proposto em [20], que não considera obstáculos no posicionamento da RSU e fornece um bom gerenciamento de tráfego. Dessa maneira, a Subseção 4.1.6 apresenta uma breve descrição sobre o sistema de re-roteamento implementado, enquanto a Subseção 4.1.6 descreve os resultados obtidos.

Descrição: Sistema de Re-roteamento Inteligente

Para permitir o gerenciamento eficiente do tráfego, o sistema de re-roteamento precisa ter conhecimento preciso sobre as condições atuais do tráfego nas vias. Para produzir esse conhecimento, o sistema conta com as informações compartilhadas pelos veículos por meio da rede veicular. Os veículos enviam sua posição e velocidade atuais para a RSU, sendo que cada veículo está conectado usando a comunicação V2I. Dessa maneira, cada RSU recebe e agrega o relatório de tráfego desses veículo e encaminha para um servidor central, que é responsável por produzir a visualização do tráfego, estimando a condição em cada via com base nas informações recebidas. A partir de então, o servidor central encaminha uma visualização contendo o estado das vias para os veículos, permitindo que eles calculem uma rota alternativa para melhorar sua mobilidade.

A visualização de tráfego é feita como um grafo ponderado das vias, na qual as interseções representam os vértices do grafo e as vias que conectam duas interseções representam as arestas. Por fim, a condição de tráfego das vias é representada pelo peso de cada aresta do grafo. Nesse cenário, cada veículo dentro de uma área de interesse (AoI) pode aplicar algoritmos de roteamento para melhorar sua mobilidade [49], que pode definir uma rota alternativa para evitar áreas congestionadas. Para este trabalho, usamos o algoritmo de re-roteamento *K-shortest path* para minimizar o problema de criar diferentes pontos de congestionamento [57]. A Figura 4.6 mostra um exemplo do sistema de redirecionamento descrito.

Vale ressaltar que a eficácia do sistema de re-roteamento depende da precisão das infor-

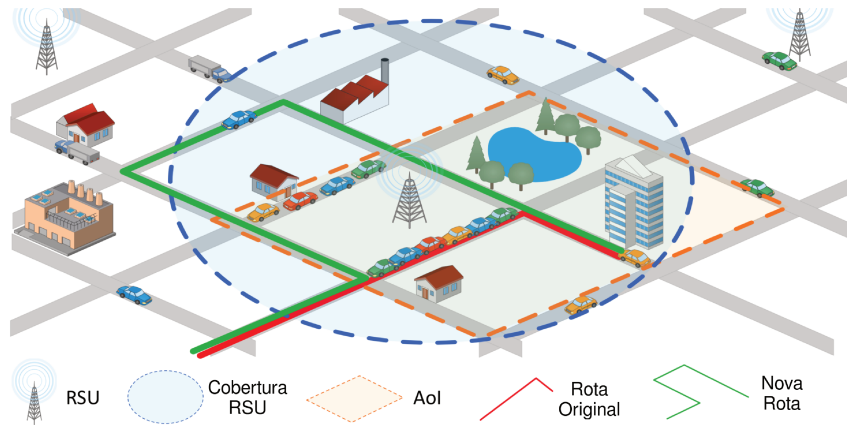


Figura 4.6: Cenário com sistema de re-roteamento inteligente

mações de trânsito, que são diretamente relacionadas com a área coberta pelas RSUs, pois as áreas sem cobertura não conseguem reportar seus dados e não conseguem obter uma visualização atualizada das vias. Assim, se algum veículo não puder se comunicar com a RSU devido a qualquer restrição (por exemplo, obstáculos, perda de sinal etc.), o conhecimento sobre a condição de tráfego será degradado e, conseqüentemente, o desempenho geral do sistema.

Análise do Gerenciamento Tráfego

Para avaliar o desempenho da alocação das RSUs obtida com o GA, comparamos a solução proposta com a CHIMERA [20], que é uma solução de literatura para gerenciamento de tráfego, baseado em infraestrutura que não considera a degradação do sinal causada por obstáculos urbanos durante o posicionamento da RSU. Dessa forma, as seguintes métricas foram avaliadas:

- **Precisão da visão do tráfego:** a porcentagem de estradas com condições estimadas de tráfego iguais às reais (extraídas do simulador) em cada intervalo de re-roteamento. Baixa precisão significa que o servidor central não está recebendo informações suficientes para representar adequadamente as condições de tráfego. Em outras palavras, as RSUs colocadas no cenário não cobrem a área para a qual foram alocadas.
- **Tempo de viagem relativo:** a relação entre o tempo de viagem do veículo usando um dos sistemas de re-roteamento e o tempo de viagem sem qualquer solução. Essa métrica resume a redução do tempo de viagem para cada veículo.
- **Perda de tempo relativa:** a relação entre a perda de tempo do veículo usando um dos sistemas de re-roteamento e a sua perda de tempo sem qualquer solução. Essa métrica resume o tempo gasto por cada veículo para percorrer todo o percurso.

A Figura 4.7 mostra os resultados para as métricas avaliadas considerando as duas soluções. Como pode ser visto, o ORC fornece uma melhor visualização do tráfego para os veículos, porque o algoritmo de otimização empregado para alocar cada RSU maximiza

a cobertura do sistema, o que significa que mais veículos são capazes de relatar as informações de tráfego para a RSU. O CHIMERA não considera a interferência causada por construções durante a alocação das RSUs, consequentemente, muitas delas podem não estar cobrindo a área para a qual foram projetadas.

O efeito resultante pela má alocação é a redução do número de veículos que podem relatar suas informações de tráfego, diminuindo diretamente a precisão da visualização. Em particular, o ORC aumenta a precisão da visualização do tráfego em aproximadamente 15% quando comparado ao CHIMERA, como é possível ver na Figura 4.8. Também é possível visualizar que a solução proposta fornece uma melhor gestão do tráfego, permitindo que os veículos terminem sua viagem mais rapidamente do que a solução original. O tempo de simulação é menor na solução proposta porque os veículos chegam aos seus destinos antes, o que pode ser observado na Figura 4.8.

A melhoria na visão de tráfego alcançada pelo ORC permite aos veículos computarem melhores rotas alternativas. Portanto, o ORC fornece um gerenciamento de tráfego melhor, resultando em menor perda de tempo e menor tempo de viagem para a maioria dos veículos, como pode ser observado nas Figuras 4.7a e 4.7b. Também é possível observar que o ORC reduz o tempo de viagem para 60% dos veículos, representado por valores menores que 1 na Figura 4.7a, enquanto aumenta o tempo de viagem para apenas 20% deles, representado por valores maiores que 1 na mesma Figura. Enquanto isso, o CHIMERA reduz o tempo de viagem apenas para 40% dos veículos e aumenta para 40% deles. Como CHIMERA alcança uma menor precisão na representação das vias, as rotas alternativas calculadas pelos veículos podem não ser as melhores opções para evitar uma área congestionada. Também devido à falta de conhecimento sobre as condições de tráfego, os veículos podem não ser capazes de calcular uma rota alternativa, consequentemente, mais veículos podem acabar encontrando congestionamento em suas rotas e a perda de tempo aumentará, como pode ser observado na Figura 4.7b, reduzindo a performance da solução.

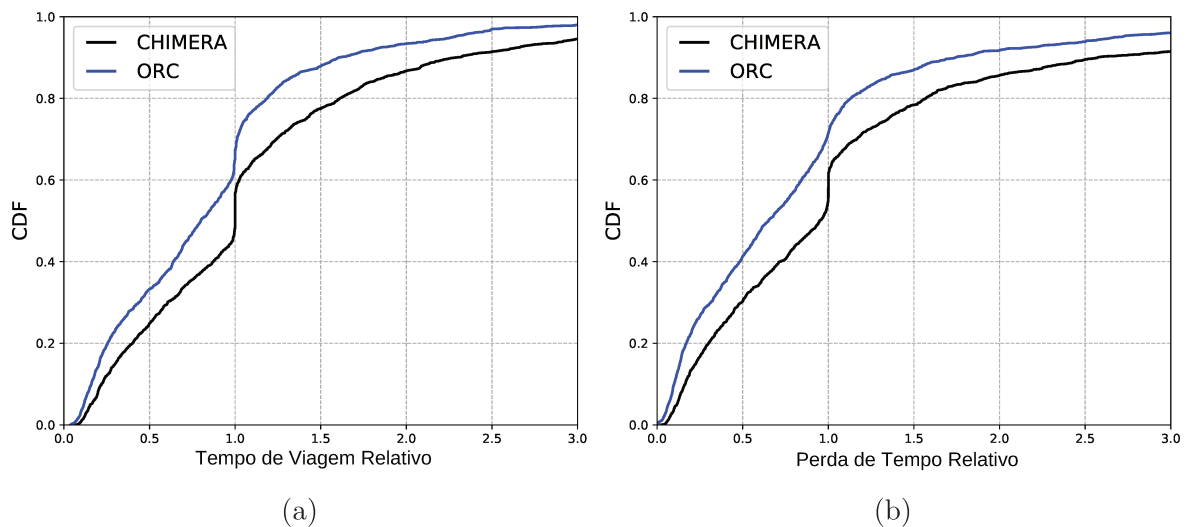


Figura 4.7: Comparação entre ORC e CHIMERA - Tempo.

A Figura 4.8 também apresenta um comportamento diferente após 1 hora de simulação. É possível ver que o ORC já chegou ao fim, que se deve, principalmente, pelo

resultado de uma melhor alocação. Ao selecionar locais melhores para alocar as RSUs os veículos conseguiram reportar suas posições com maior frequência e receber a situação atual de outras vias com maior precisão. Como o CHIMERA teve maior dificuldade nesse item, os veículos reportaram menos posições e obtiveram representações do cenários imprecisas, resultando em seleção de rotas mais demoradas. A menor quantidade de veículos reportando suas posições também resulta em um intervalo de confiança maior para o CHIMERA.

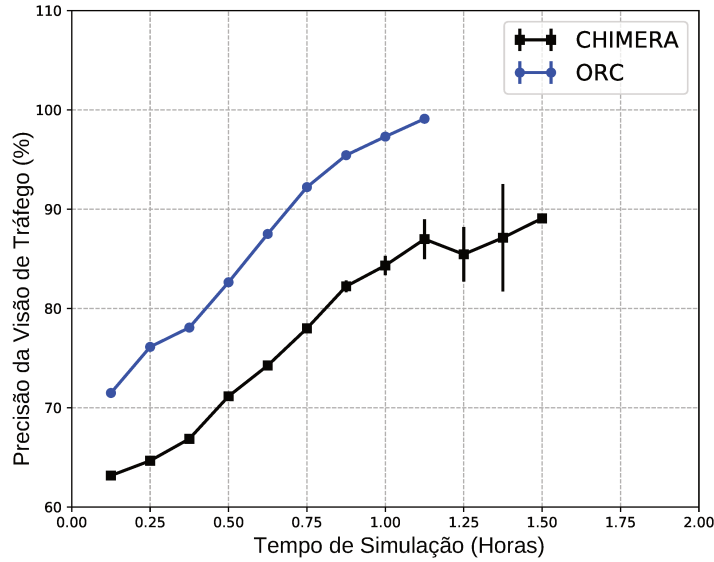


Figura 4.8: Comparação entre ORC e CHIMERA - Precisão.

4.2 RP-Sim: Radio Propagation Simulator

Nesse capítulo é apresentado o Radio Propagation Simulator. O processo realizado no capítulo 4.1, para determinar a área de comunicação, se mostrou custoso no que se refere ao tempo e com grande dificuldade de realização. A utilização de simuladores para determinar essa informação, além de requerer um grande conhecimento específico sobre os simuladores e redes sem fio, necessita de mais tempo para completar a simulação, o que pode aumentar rapidamente com o aumento do cenário. Esse tipo de abordagem também necessita de estágio de pós processamento para converter diversas coordenadas em polígonos.

Levar em consideração os limites presentes nessa situação é essencial para uma boa alocação e para o bom funcionamento dos serviços que dependem dessa comunicação. Os dados sobre potência de recepção e *Bit Error Rate* (BER) em diferentes locais são importantes para que os objetivos de cobertura definidos sejam atingidos de maneira eficiente. O BER é uma métrica utilizada para definir a taxa de perda de bits durante a comunicação.

O conhecimento detalhado da propagação das ondas de rádio nas cidades é indispensável para fazer o planejamento adequado da alocação de infraestrutura. Para isso, foi desenvolvido o *Radio Propagation Simulator* (RP-Sim), ferramenta capaz de considerar

diversas informações relevantes para essa tarefa e automatizar o processo de cálculo da propagação de sinal em diferentes ambientes, como urbanos e rurais. A ferramenta proposta também permite simular a propagação de sinal considerando as construções da região, o modelo de propagação de sinal e a modulação que mais se adéquem a necessidade. Além disso, a ferramenta também permite exportar as áreas onde a comunicação atinja valor mínimo de potência recebida ou máximo de BER. A ferramenta foi desenvolvida para ser utilizada de maneira fácil e rápida, até mesmo por pessoas com pouca experiência na área, auxiliando, assim, estudantes, professores, pesquisadores e empresas.

Com o objetivo de facilitar o uso, toda a ferramenta RP-Sim foi desenvolvida completamente com código aberto e planejada para ser utilizada direto pelo navegador do usuário, necessitando apenas de acesso à internet para carregar o mapa. A arquitetura utilizada foi pensada para permitir que os usuários consigam criar suas próprias melhorias e implementar novos modelos de propagação e modulação do sinal, permitindo que sua utilização seja cada vez mais abrangente. Por conta de sua flexibilização, também é possível avaliar novas tecnologias e o impacto delas nos serviços e aplicativos relacionados a comunicação sem fio. Com a utilização do RP-Sim esperamos que a academia e a indústria possam encontrar soluções melhores, mais rápidas, mais baratas e com menos esforços, beneficiando a todos neste ciclo. A Tabela 4.3 apresenta os parâmetros que estão disponíveis para configuração atualmente:

Tipo	Configuração
Ambiente	Temperatura Figura de ruído
Comunicação	Frequência Largura de banda Modulação
Transmissor	Potência de transmissão Ganho Altura
Receptor	Ganho Altura
Geral	Posição Antenas Coordenadas Construções
Propagação	8 Modelos para ar livre 1 Modelo de degradação por metro em obstáculo 1 Modelo de degradação por corte causada por obstáculo

Tabela 4.3: Configurações disponíveis

O código fonte, descrições adicionais e tutorial passo a passo estão disponíveis no endereço <https://github.com/MatheusFerraroni/RP-Sim>. A distribuição será feita com a licença *GNU Affero General Public License v3.0* (AGPL-3.0), por ser uma licença bastante permissiva, que concede o direito de alterar o código fonte de acordo com a necessidade dos usuários, desde que eles compartilhem suas alterações e mantenham a mesma licença.

4.2.1 Arquitetura

A arquitetura desenvolvida para o RP-Sim é apresentada na Figura 4.9, onde são apresentadas três principais divisões da ferramenta. A parte 'Mapa e Interação' é responsável por apresentar o mapa e controlar as interações com o mesmo. Na parte 'RP-Sim' estão os módulos do RP-Sim, bem como o cálculo da propagação do sinal e a gerência das construções e antenas. A parte 'Visualização' é responsável por apresentar os resultados obtidos de maneira visual sobre o mapa.

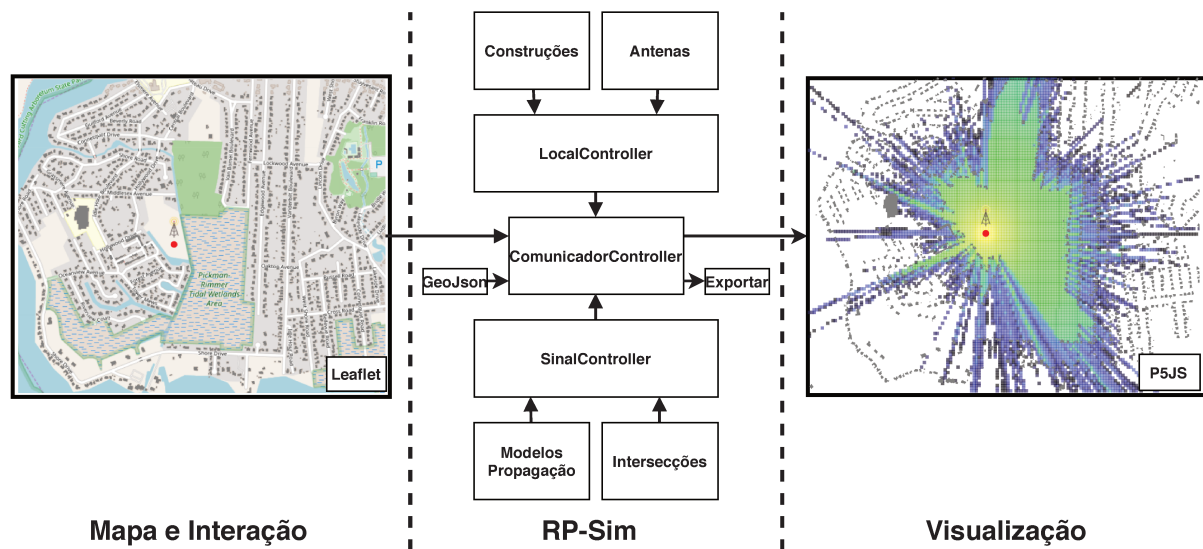


Figura 4.9: Arquitetura desenvolvida

O mapa Leaflet² foi utilizado por ter seu código aberto, por ser fácil de utilizar pelos usuários finais e por ter carregamento rápido e ser leve. Esse mapa já está preparado para ser utilizado em dispositivos móveis.

A parte 'RP-Sim' é responsável por realizar os processamentos necessários para as funcionalidades desenvolvidas. Essa parte está dividida em 7 módulos que são responsáveis desde o controle dos pontos amostrais, até a realização dos cálculos das intersecções da comunicação entre antena e ponto amostral. Os módulos da ferramenta são descritos abaixo:

1. **Construções:** Armazena informações sobre formato e altura para cada construção. Funções de criar, gerenciar e mostrar as construções no mapa, também estão implementadas aqui.
2. **Antenas:** Responsável pela gerência das antenas, guardando informações como posição e sendo responsável pela sua apresentação no mapa.
3. **LocalController:** Responsável por mapear os pontos amostrais onde o RSSI e o BER são calculados. A visualização dessas informações são definidas aqui.

²<https://leafletjs.com/>

4. **Intersecções:** Responsável por calcular a interseção entre a linha formada pela antena e ponto amostral com as construções do cenário. Foi criado de forma separada, pois a posição de checagem de RSSI muda de acordo com a granularidade de checagem utilizada, que se ajusta automaticamente com o zoom no mapa.
5. **Modelos Propagação:** São classes responsáveis por definir a propagação do sinal. Os modelos de propagação implementados na ferramenta são:
 - Sem obstáculo
 - SUI [4]
 - Egli [24]
 - Ericsson [14]
 - Nakagami-m [9]
 - Free Space [26]
 - Okomura Hata [47]
 - Two Ray Ground [55]
 - Fixed Distance
 - Com Obstáculos
 - Por metro em obstáculos [54]
 - Por corte no sinal
 - Por corte causado por obstáculos [54]

Também considera-se a modulação da comunicação para definir o BER [42]. As modulações implementadas na ferramenta são:

- BSPK, QPSK
 - 8PSK
 - 16PSK
 - 4QAM
 - 16QAM
 - 64QAM
6. **SinalController:** As configurações de frequência, largura da banda, potência de transmissão, ganhos, altura de transmissão e recepção e qual modulação está selecionada, ficam salvas nesse módulo para serem convertidas para as medidas necessárias. Qual modelo de propagação está ativo e sua execução também são responsabilidades desse módulo.
 7. **ComunicadorController:** Esse módulo é responsável pela integração de todos os outros. Responsável, efetivamente, por calcular a propagação do sinal entre a posição das antenas e do locais definidos para checagem, considerando as informações disponíveis e configurações definidas. Também implementa as funções de importação e exportação dos dados necessários.

A terceira parte é responsável por apresentar os resultados de maneira visual. Essa apresentação é feita com uma sobreposição da camada de visualização sobre o mapa, facilitando o entendimento. Para colocar as informações necessárias na camada de visualização, foi utilizada a biblioteca Mappa.js³ em conjunto com a biblioteca P5JS⁴. A biblioteca Mappa.js se encarrega de alinhar a sobreposição da camada de visualização com o mapa e a biblioteca P5JS permite que os resultados obtidos sejam apresentados de maneira simples e rápida.

4.2.2 Implementação

Para realizar a implementação da ferramenta foram adotadas tecnologias WEB, principalmente a linguagem de programação Javascript, por ser a linguagem mais popular de 2014 a 2019 [28]. A utilização dessa linguagem também permite acesso a avançados recursos visuais e inúmeras bibliotecas que podem auxiliar na evolução da ferramenta. Com a utilização de tecnologias WEB, é possível que a execução aconteça diretamente do navegador do usuário, facilitando o acesso e removendo requisitos como instalação de programas no dispositivo. A ferramenta também foi desenvolvida de maneira que não seja necessário a utilização de servidores dedicados para seu funcionamento, apenas o computador por onde ela foi acessada.

A Figura 4.10 apresenta o diagrama de classes que foi implementado para as classes de propagação de sinal. É possível ver que existem 3 especializações distintas para a propagação de sinal: 'PropagationCorte', que considera o número de vezes que o sinal foi interrompido; 'PropagationObstacle', que considera a distância percorrida dentro de obstáculos; e 'PropagationNoObstacle', que considera a distância percorrida fora de obstáculos.

Pelo diagrama apresentado, é possível observar que a implementação de um novo modelo pode ser feita de maneira simples, apenas especializando uma das 3 principais classes e implementando o método 'calculate'. O método 'calculate' recebe sempre 3 parâmetros: a distância percorrida fora de obstáculos em metros; a distância percorrida dentro de obstáculos em metros; e a quantidade de vezes que o sinal foi interrompido por obstáculos. Para implementar o modelo de propagação 'FreeSpace', por exemplo, basta somar as distâncias de interesse, dividir por 1000 para transformar em quilômetros, aplicar a fórmula do FreeSpace e retornar o valor obtido.

Quando configurações específicas para um novo modelo de propagação são necessárias, é possível adaptar o código que insere as informações nas configurações dos modelos de propagação para que campos especiais sejam adicionados.

Entrada

O padrão de inserção de dados externos é o GeoJson, que, atualmente, é utilizado para carregar as localizações para as antenas e carregar dados, como formato e posição dos obstáculos. O GeoJson é um formato para armazenar informações geográficas utilizando

³<https://mappa.js.org/>

⁴<https://p5js.org/>

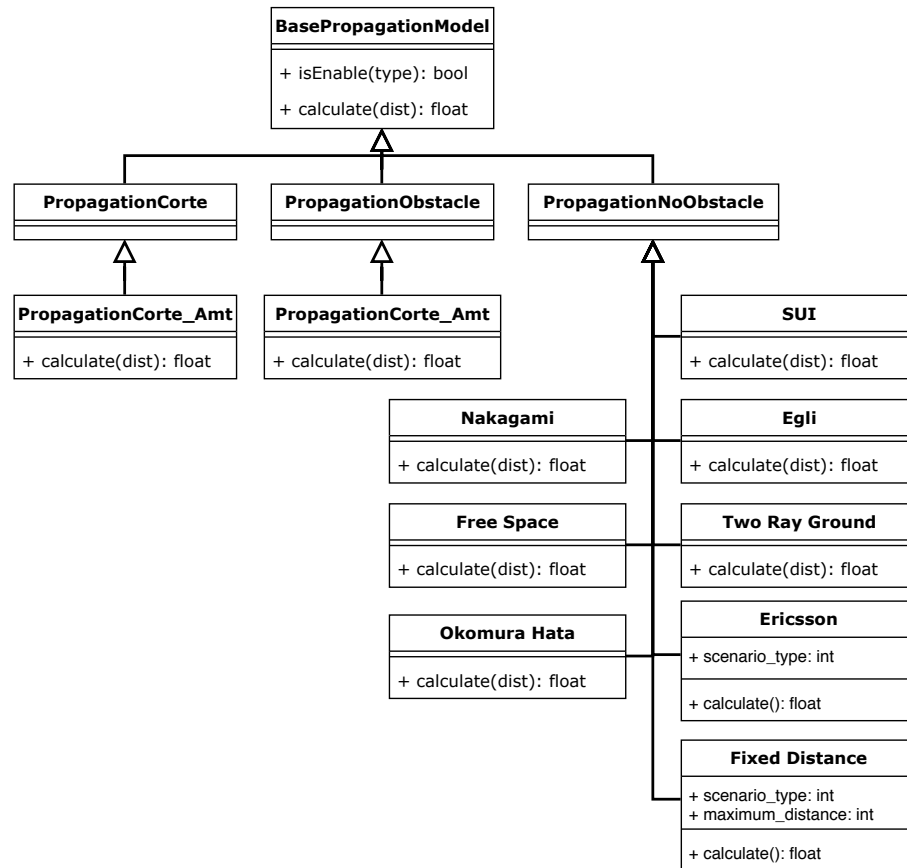


Figura 4.10: Classes de propagação de sinal

Json. A obtenção dos dados sobre os obstáculos pode ser feita utilizando a ferramenta Overpass Turbo ⁵, que consegue requisitar de maneira rápida e fácil os dados do OpenStreetMaps.

Saída

A saída de dados foi padronizada no formato GeoJson, por ser um formato amplamente conhecido e utilizado. O principal dado exportado na ferramenta são os polígonos da área de comunicação com as antenas. Esses polígonos são definidos de acordo com as configurações que foram fornecidas, como frequência e potência de transmissão, e o usuário pode definir qual o valor mínimo de RSSI ou valor máximo para o BER. Durante o estágio de RSSI mínimo é apresentado o valor do BER esperado para o RSSI selecionado, assim é possível alterar o RSSI mínimo para ser compatível com o BER máximo aceitado. Os polígonos exportados são posições no padrão de coordenadas WGS84, que é o mais utilizado em sistemas geográficos.

4.2.3 Funcionalidades

A ferramenta desenvolvida compreende diversos módulos para calcular a propagação do sinal, o RSSI recebido e o BER, de acordo com o RSSI. Para isso, foram implementadas

⁵<https://overpass-turbo.eu/>

funções para importação e exportação de dados, classes para propagação de sinal e diversos controles e configurações, que permitem utilizar a ferramenta para as mais diversas finalidades que estejam relacionadas com o cálculo da propagação de sinal.

A parte de visualização dos resultados obtidos foi montada para funcionar em conjunto com o mapa, já que os dados gerados são todos geo-referenciados. Uma camada de visualização foi adicionada sobre o mapa e as informações foram apresentadas de maneira intuitiva em conjunto com diferentes escalas de cores. É possível visualizar a propagação do sinal, o RSSI recebido em diferentes locais, o BER calculado de acordo com o RSSI e mapear áreas com uma qualidade mínima de sinal ou de RSSI.

Para determinar o RSSI recebido e o BER, são consideradas as seguintes configurações: temperatura; altura da antena transmissora; altura da antena receptora; figura de ruído; frequência de comunicação; largura do canal; ganho de transmissão; ganho de recepção; diferentes modelos de propagação de sinal e diferentes modulações para a comunicação. Além disso, um ponto importante considerado são os módulos específicos para definir o impacto causado pelos obstáculos no cenário.

Além das funcionalidades apresentadas na arquitetura, as classes de propagação de sinal foram projetadas de maneira que evoluções, adaptações e melhorias, possam ser feitos de maneira simples e rápidas, incentivando o uso da ferramenta pela comunidade, que pode adaptá-la às suas necessidades.

4.2.4 Demonstração

A utilização da ferramenta foi idealizada para ser o mais simples possível, para permitir que até mesmo pessoas com pouca experiência consigam fazer o uso. Para isso, configurações básicas e pré configuradas já são carregadas junto com a ferramenta.

A utilização foi definida em passos simples e diretos. Para utiliza-la é necessário abrir o site, importar os dados necessários com as posições dos obstáculos ou antenas, calcular a propagação do sinal e as áreas de cobertura. Sua utilização depende apenas de um computador com navegador e acesso à internet.

Os cálculos de intersecção da linha de comunicação, entre a antena transmissora e a receptora, é um processo mais lento, por isso cria-se um cache na primeira execução. Assim, sempre que alguma configuração for alterada, exceto a posição dos obstáculos, das antenas, ou o zoom, os cálculos são refeitos de maneira quase instantânea, assumindo que a intersecção já está calculada. O zoom também tem um papel fundamental nesse cache, devido à sua utilização para definir automaticamente a granularidade dos locais para realizar a amostragem do RSSI e do BER.

A Figura 4.11 apresenta um exemplo da ferramenta sendo utilizada para visualizar a propagação do sinal. As cores no mapa indicam o RSSI sendo recebido por cada local avaliado. As configurações nesse teste representam a alocação de 2 *RoadSide Units* em uma região com poucas construções.

Uma vez que o processo de calcular o RSSI recebido esteja completo é possível trocar a visualização para o modo BER, apresentado na Figura 4.12, e para o modo onde são gerados polígonos de qualidade de sinal, apresentado na Figura 4.13.

Os resultados obtidos na Figura 4.13 podem ser exportados no formato geojson, con-

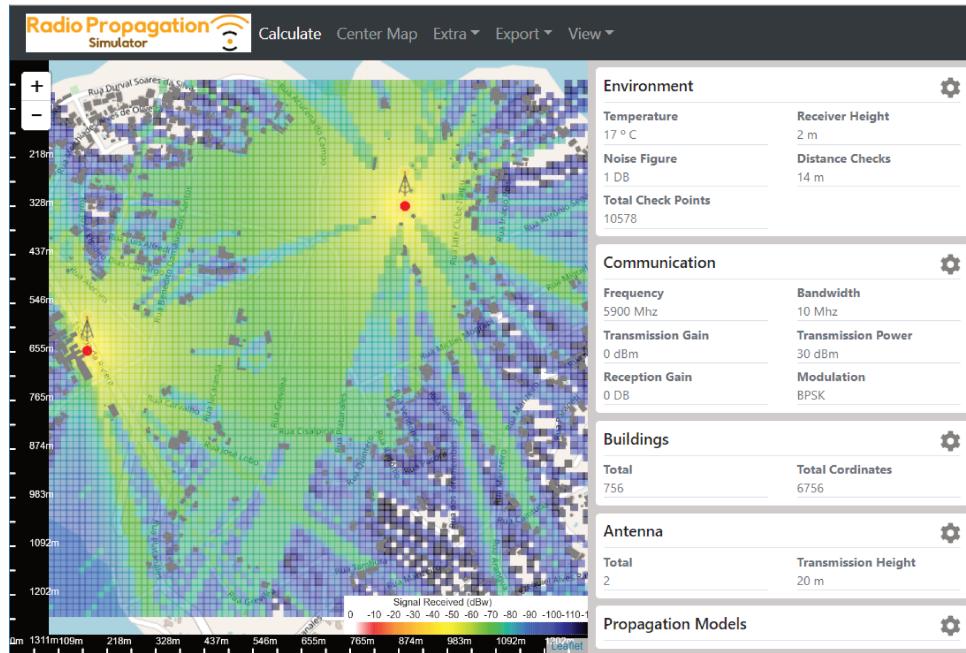


Figura 4.11: Exemplo de resultado obtido

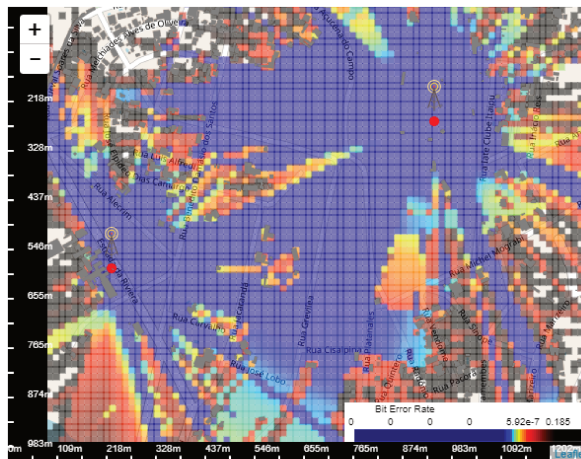


Figura 4.12: BER

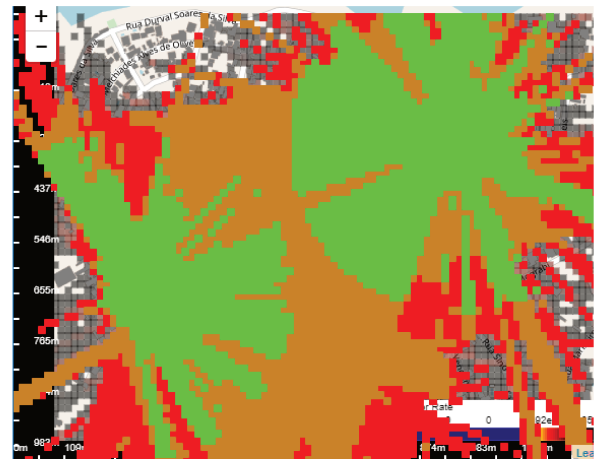


Figura 4.13: Área vs RSSI mínimo

tendo os 3 polígonos que contém a área de comunicação com sinal considerado bom, médio e ruim. É possível alterar o valor mínimo de RSSI e o BER máximo para que uma área seja considerada boa, média ou ruim.

4.3 Otimização de RSU com Diferentes Modelos de Propagação de Sinal

Neste capítulo, apresenta-se a evolução do trabalho apresentado no Capítulo 4.1. Como explorado anteriormente, a utilização de simuladores é um processo temporalmente custoso e, aliado a isso, a utilização de apenas um modelo de propagação pode levar até uma má tomada de decisão durante a alocação dos recursos disponíveis, pois como foi mostrado anteriormente há diferentes modelos de propagação podem ser mais adequadas

ao cenário. Para auxiliar nesse processo, a ferramenta que foi introduzida no Capítulo 4.2 é utilizada, tanto para ganhar velocidade, quanto para expandir a quantidade de modelos de propagação considerados.

Considerar outros modelos também mostra a versatilidade da abordagem empregada e da ferramenta desenvolvida, permitindo que essa abordagem seja aplicada para realizar alocação de outros tipos de dispositivos que se comunicam de maneira sem fio e precisam ter sua cobertura maximizada, como antenas LTE ou 5G. O algoritmo genético utilizado nessa proposta é uma adaptação do algoritmo do Capítulo 4.1, divergindo apenas nos dados de entrada e na configuração do GA.

Apesar do algoritmo genético empregado ser o mesmo, existem diversas diferenças e evoluções. A Tabela 4.4 apresenta as principais diferenças e evoluções entre o trabalho atual e o que foi apresentado no Capítulo 4.1.

	Capítulo 4.1	Atual
Mapeamento de comunicação	Veins e OMNeT	RP-Sim
Tempo com mapeamento	Horas	Segundos
Modelos de propagação	FreeSpace e obstáculos	4 modelos de propagação Obstáculos Corte
Posições candidatas	Cruzamentos	Espaçamento fixo cruzamentos
Complexidade do cenário	Baixa	Alta
Cáculo da cobertura	Apenas Vias	Cenário todo

Tabela 4.4: Diferenças e evoluções

Os dois dados que são utilizados na entrada do GA são informações sobre o cenário a ser coberto e informações sobre as posições e áreas de cobertura de cada dispositivo. As informações sobre o cenário variam porque nesse momento o cenário utilizado é o centro da cidade de São Paulo. As informações sobre as áreas de cobertura foram obtidas utilizando a ferramenta apresentada no Capítulo 4.2.

Também é avaliada a alocação de dois conjuntos distintos de posições candidatas para realizar a alocação das RSUs. No primeiro grupo, as posições candidatas são selecionadas com um espaçamento fixo entre elas e no formato de GRID, enquanto no segundo, as posições selecionadas são em intersecções do cenário. Comparando esse tipo de diferença nas posições torna-se possível discutir os impactos de selecionar as posições candidatas com mais ponderação.

4.3.1 Identificar Área de Comunicação

Como já mencionado, a identificação da área de comunicação foi feita utilizando a ferramenta RP-Sim, que foi apresentada no Capítulo 4.2. Com a utilização dessa ferramenta é possível definir a área de comunicação de cada RSU para toda a área de interesse, não apenas em regiões que estão sobre as vias do cenário. Esse cálculo pode ser feito de maneira rápida para diferentes modelos de propagação, mesmo que sejam considerados os obstáculos do cenário.

Os parâmetros que foram utilizados durante o estágio de configuração da ferramenta RP-Sim são apresentados na tabela 4.5. É possível observar que os valores empregados foram escolhidos para simular uma rede do padrão 802.11p, que é o padrão utilizado para a comunicação veicular e está presente nas RSUs do cenário.

Parâmetros	Valor
Frequência	5900 Mhz
Potência de Transmissão	30 dBm
Ganho de transmissão	0 dB
Ganho de recepção	0 dB
Altura da antena de transmissão	30 metros
Altura da antena de recepção	1.5 metros
Construções	6131

Tabela 4.5: Parâmetros utilizados no RP-Sim

Com a ferramenta parametrizada, os cálculos para definir a área de comunicação podem ser iniciados de forma sequencial para cada posição candidata. A Figura 4.14 apresenta um dos resultados utilizando a ferramenta desenvolvida para calcular a área de comunicação. Essa informação é exportada como um geojson que contem os dados sobre os polígonos onde a qualidade de comunicação é considerada boa, média e ruim, que estão respectivamente representadas na figura com as cores verde, laranja e vermelha.

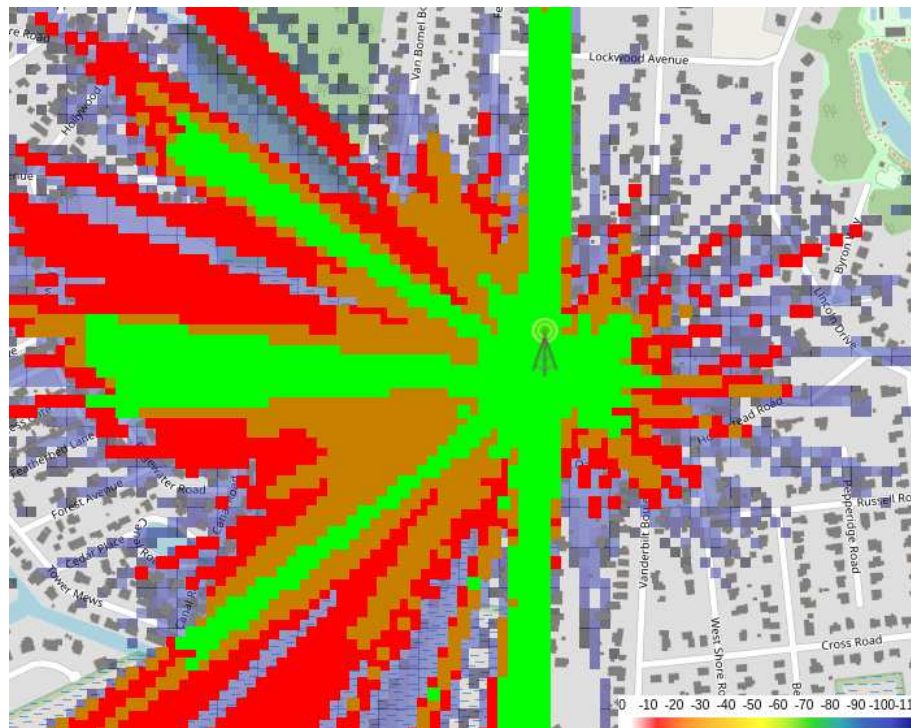


Figura 4.14: Exemplo de área de comunicação obtido para uma RSU

O RSSI foi utilizado para determinar as regiões onde a comunicação foi classificada como boa, média ou ruim. A utilização dessa informação isoladamente não é o suficiente para determinar se a comunicação acontece em determinado local, mas com ela é possível

filtrar locais onde o receptor não consegue detectar a comunicação. Para auxiliar nesse processo, foi adicionado um limite do BER em regiões onde o receptor consiga receber o sinal, para que, quando a taxa de erro for alta, tais regiões sejam descartadas.

4.3.2 Descrição do Cenário

Esta seção apresenta o cenário que foi utilizado para avaliar a solução proposta. O cenário escolhido foi o centro da cidade de São Paulo, em razão de existirem bastante construções e por facilitar a visualização da influência dos obstáculos que foi discutida até o momento. As configurações do cenário utilizado é apresentada na Tabela 4.6.

Parâmetro	Valor
Área Total	2.3km ²
Cidade	São Paulo
Região	Urbana
Centro	-23.533320,-46.613510
Construções	6131
Coordenadas Construções	61000+

Tabela 4.6: Configurações do cenário

As informações sobre posições e formatos das construções, vias e cruzamentos, foram obtidas utilizando o OpenStreetMaps. É possível observar que cada construção é constituída por uma série de coordenadas, o que tem impacto direto no tempo de execução dos algoritmos de ray tracing utilizados na ferramenta.

4.3.3 Algoritmo Genético

Como já mencionado, o problema de cobertura é NP-difícil [43] e, por isso, para encontrar soluções viáveis para este problema foi utilizado um algoritmo genético, que é um tipo de algoritmo evolutivo [44]. As configurações utilizadas pelo algoritmo genético estão presentes na tabela 4.7.

Para definir o *score* calcula-se a porcentagem normalizada da área coberta pelas RSUs selecionadas em cada elemento da população do algoritmo genético. Com isso é possível descartar os elementos que tiveram uma performance ruim e utilizar os que tiveram o *score* maior para gerar a nova população.

É importante ressaltar que existem diferenças relacionadas ao cálculo da cobertura proposto no Capítulo 4.1 e a cobertura proposta aqui. Nessa seção, a cobertura refere-se a área total do cenário, enquanto a área considerada anteriormente referente-se apenas as vias do cenário.

O tipo de genoma escolhido foi do tipo *float*, isso significa que cada item do genoma precisa ter associada uma informação, que pode ser usada para construir uma solução e calcular o score. Cada *float* do genoma tem associada uma posição candidata para alocar uma RSU, e, após ser ordenado, as primeiras K posições são selecionadas para alocar uma RSU, onde K é a quantidade de RSUs sendo selecionada naquele momento.

Parâmetro	Valor
Seleção	Baseada em Rank
Codificação	Float
Tamanho da População	100
Inserção de Mutação	Novos Elementos
Limite de Iteração	300
Quantidade de RSU	[1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
Quantidade de Melhores Elementos	10%
Quantidade de Elementos Aleatórios	20%
Quantidades de Elementos Filhos	70%
Porcentagem de Herança de Genoma	80%/20%

Tabela 4.7: Configurações do algoritmo genético

A equação 4.1 apresenta a função objetivo, onde a cobertura é maximizada de acordo com a quantidade de RSUs sendo utilizadas. Em um conjunto de n localizações candidatas para alocar uma RSU, a cobertura total é representada pela união da área coberta pelas k RSUs selecionadas.

$$\text{maximizar } \sum_{i=1}^n \text{Cobertura}_i * s \quad (4.1)$$

$$s = \begin{cases} 1, & \text{se posição pertence a } k \\ 0, & \text{se posição não pertence a } k \end{cases} \quad (4.2)$$

4.3.4 Cálculo da Área de Cobertura

Diferente do cálculo executado na Seção 4.1.5, a área que deve ser coberta nesse momento não tem mais o formato das vias no cenário, mas sim o formato do cenário inteiro. O cenário é transformado em um polígono com o mesmo tamanho e formato, bem como as áreas de comunicação com cada RSU.

A utilização da ferramenta apresentada no Capítulo 4.2 permite calcular a área de comunicação para uma região inteira e não apenas para posições sobre as vias. Com isso a cobertura calculada relaciona-se ao cenário todo, considerando vias, parques, construções e tudo mais dentro da área de interesse. A Figura 4.15 apresenta um exemplo do funcionamento desse processo.

No estágio A, forma-se um polígono que contém a área de interesse inteira e tudo que tem dentro dela. No estágio B, as áreas de comunicação com as RSUs são adicionadas, representadas pelos polígonos azuis. No estágio C, calcula-se a diferença entre esses polígonos.

Como resultado desse processo tem-se um polígono que representa as regiões que não são cobertas por nenhuma RSU. A área final dividida pela área inicial resulta na porcentagem normalizada da área que ainda falta ser coberta por uma RSU.

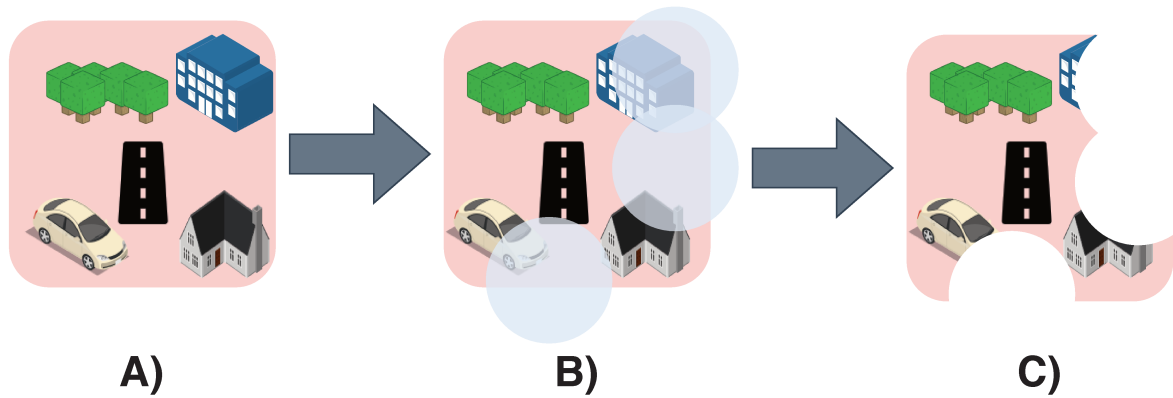


Figura 4.15: Comparação entre cobertura e sobreposição

4.3.5 Avaliação da Alocação

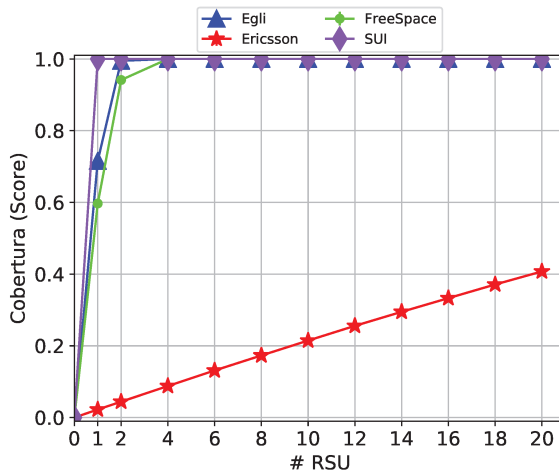
Um algoritmo genético foi utilizado para realizar a alocação de RSUs em um cenário no centro da cidade de São Paulo considerando diferentes modelos de propagação de sinal e obstáculos no cenário. As áreas de comunicação para cada RSU foram definidas utilizando a ferramenta RP-Sim. Além disso, foram utilizados dois tipos de localizações candidatas para o mapeamento de área de comunicação: no primeiro, são selecionadas posições com espaçamento fixo, a cada 150 metros; no segundo método, os locais para alocar uma RSU são os cruzamentos da região selecionada. Os resultados foram calculados com intervalo de confiança de 95% para 33 execuções e podem ser vistos nas Figuras 4.16 e 4.17.

A Figura 4.16 apresenta o resultado da cobertura quando a alocação está considerando apenas as posições com espaçamento fixo. A Figura 4.16a apresenta a cobertura quando a alocação não considera os obstáculos para calcular a propagação do sinal, dessa maneira a área de comunicação de cada RSU é bem maior do que na realidade, resultando em uma cobertura total da região selecionada com poucas RSUs. O modelo Ericsson é o único que apresenta uma cobertura mais baixa, devido a sua degradação ser maior por conta de sua parametrização, que se ajusta ao cenário urbano, gerando uma maior perda de sinal.

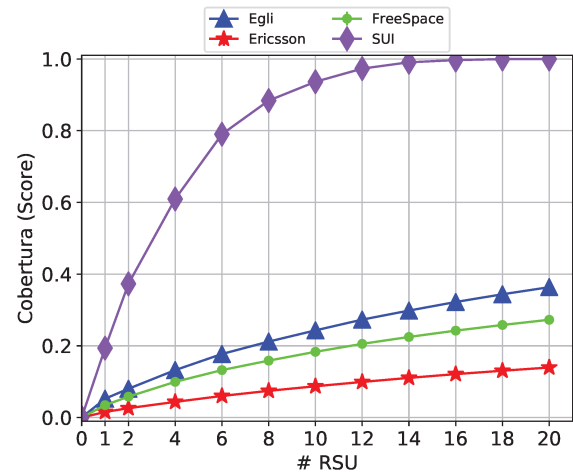
Na Figura 4.16b os obstáculos são considerados para calcular a propagação do sinal. Uma vez que essa informação é considerada durante o cálculo da área de comunicação, os resultados são áreas bem menores, sendo mais próximos da área de comunicação do mundo real. A alocação com essas áreas torna evidente a necessidade de mais dispositivos para aumentar a cobertura. Os resultados utilizando os modelos Egli, FreeSpace e Ericsson passam a ser semelhantes, mas ainda abaixo de 0.4 devido a grande atenuação causada pelos obstáculos.

A Figura 4.17 apresenta os resultados após a alocação em que as posições para o cálculo das áreas de cobertura foram feitas utilizando os cruzamentos do cenário. É possível observar que a alocação nos cruzamentos desfavoreceu o modelo Ericsson, já que ele é um modelo que já tem uma perda maior embutida nos seus cálculos, simulando um cenário urbano.

É possível ver na Figura 4.17a que, sem considerar os obstáculos, o mesmo comportamento dos resultados com posição fixa é obtido. A cobertura sobe rapidamente para quase todos os modelos. Quando os obstáculos são considerados para a degradação do



(a) Região urbana sem considerar obstáculos

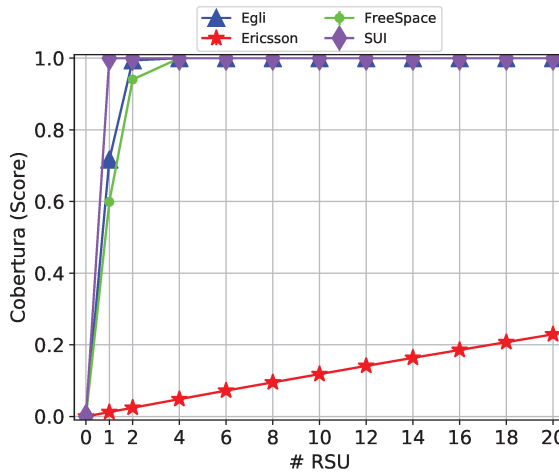


(b) Região urbana considerando obstáculos

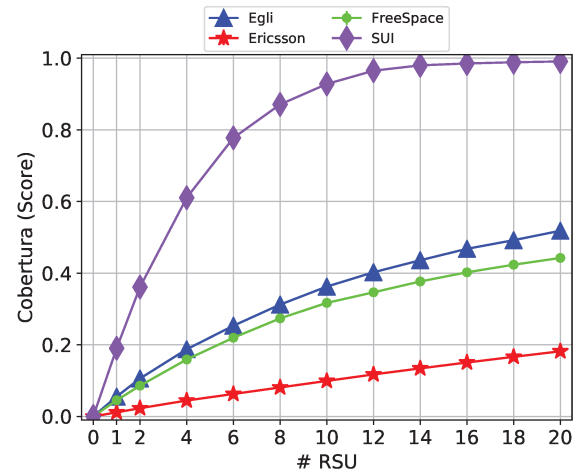
Figura 4.16: Resultado utilizando posições com espaçamento fixo

sinal, é possível ver na figura 4.17b que a cobertura se reduz, sendo necessárias mais RSUs para obter a mesma cobertura.

Ao comparar os resultados das figuras 4.16b e 4.17b, com posições candidatas fixas e posições candidatas nos cruzamentos, é possível notar a melhoria na cobertura das RSUs ao selecionar os cruzamentos para alocar as RSUs, já que, agora, o sinal consegue uma distância maior em linha de visão com diversas localidades.



(a) Região urbana sem considerar obstáculos



(b) Região urbana considerando obstáculos

Figura 4.17: Resultado utilizando cruzamentos para alocação

Capítulo 5

Considerações Finais e Trabalhos Futuros

5.1 Conclusões

Este trabalho utiliza um algoritmo genético para realizar a alocação de *Road Side Units* em diferentes configurações. Dessa forma, a alocação desses dispositivos pode ser executada em cenários urbanos, considerando diferentes modelos de propagação de sinal e o impacto causado por obstáculos. A área de comunicação foi definida de duas maneiras: utilizando simulador e modelo de propagação de sinal, no Capítulo 4.1, e utilizando a ferramenta RP-Sim com diferentes modelos de propagação de sinal, no Capítulo 4.3. Um caso de uso foi explorado na seção 4.1.6, mostrando a melhoria que é obtida ao considerar os obstáculos para definir onde alocar as RSUs. Também foi comparado as diferenças obtidas realizando a alocação com áreas de comunicação derivadas dos diferentes modelos de propagação de sinal. A partir dos resultados é possível encontrar soluções viáveis e que podem melhorar significativamente o funcionamento das aplicações implementadas quando a alocação das RSUs é feita de maneira adequada.

A criação e disponibilização da ferramenta RP-Sim, apresentada no Capítulo 4.2, também foi feita durante esse trabalho. Essa ferramenta tem como objetivo facilitar a visualização da perda de sinal e calcular a área de comunicação considerando diferentes modelos de propagação de sinal, levando em conta diversas configurações para a comunicação, antenas e cenário. A implementação de novos modelos de propagação de sinal também é possível e pode ser realizada de maneira simplificada, permitindo o uso dessa ferramenta para diversas finalidades.

Também é possível destacar que os modelos de propagação disponíveis atualmente apresentam características bastante distintas, bem como vantagens e desvantagens. Escolher um modelo de propagação que se adeque às configurações de comunicação e antenas utilizadas é um passo essencial para obter resultados próximos aos que seriam obtidos no mundo real. Dessa forma, não existe o melhor modelo de propagação, mas, sim, o modelo que melhor representa uma determinada configuração.

5.2 Publicações

As produções científicas feitas durante o período do mestrado estão listadas nas Listas 5.2.1 e 5.2.2.

5.2.1 Publicações Como Primeiro Autor

- ★¹ Matheus Ferraroni Sanches, Welligton Viana Lobato Junior, Fábio Luiz Usberti, Leandro Aparecido Villas. Genetic Algorithm for the Maximum Coverage Location Problem Applied to Medical Emergency. In: LI Simpósio Brasileiro de Pesquisa Operacional, 2019, Limeira.
- Matheus Ferraroni Sanches, Allan Mariano de Souza, Wellington Viana Lobato Junior, Leandro Aparecido Villas. Optimizing Infrastructure Placement with Genetic Algorithm: A Traffic Management Use Case. In: IEEE Latin-American Conference on Communications, 2019, Salvador.
- (Submetido) - Matheus Ferraroni Sanches, Oscar Jaime Ciceri Coral, Leandro Aparecido Villas. RP-Sim: Radio Propagation Simulator. In: Salão de Ferramentas do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2020.

5.2.2 Cooperação em Outros Trabalhos

- Lucas Zanco Ladeira; Allan Mariano de Souza ; Geraldo P. Rocha Filho; Thiago Henrique Silva; Matheus Ferraroni Sanches; Leandro Aparecido Villas. MARTINI: Towards a mobile and variable timewindow identification for spatio-temporal data. In: IEEE Latin-American Conference on Communications, 2019, Salvador.
- Lucas Zanco Ladeira; Matheus Ferraroni Sanches; Cassio Viana; Leonardo Castro Botega. Assessing the Impact of Mining Techniques on Criminal Data Quality. In: Workshop de Computação Urbana (COURB), 2. , 2018, Campos do Jordão. Anais do II Workshop de Computação Urbana. Porto Alegre: Sociedade Brasileira de Computação, maio 2018.

5.3 Trabalhos Futuros

Muitos aspectos ainda podem ser considerados para a realização da alocação de RSUs. Esses aspectos podem estar ligados com as informações que são utilizadas para realizar a alocação ou, até mesmo, com a forma que a alocação é feita pelo algoritmo genético. A seguir serão apresentados alguns dos principais pontos de melhora que foram identificados ao longo da execução desse trabalho.

¹★Menção Honrosa

- **Quantidade de Conexões:** Considerar a quantidade de conexões que uma RSU suporta é importante para determinar se a alocação atual é suficiente para atender as possíveis demandas. Considerar esse tipo de informação pode permitir que a quantidade de RSUs e a potência de transmissão de cada RSU ao longo do dia seja ajustada para melhorar a conexão com os dispositivos.
- **Variação de Conexões e Potência de Transmissão:** A quantidade de dispositivos podem variar ao longo do dia na mesma região. Considerar esse tipo de variação permite alocar em regiões com maior densidade RSUs mais próximas e com uma potência menor, fazendo com que a quantidade de dispositivos em comunicação com elas seja menor e a qualidade da conexão seja melhor. Regiões menos densas podem ter menos RSUs com uma potência maior, abrangendo uma área de comunicação maior.
- **Aprimorar Algoritmo Genético:** Atualmente o algoritmo genético enfrenta o problema apresentado de uma só vez, procurando a melhor solução, considerando todas as possíveis localizações para alocar uma RSU. Esse trabalho pode ser precedido por encontrar soluções para sub-regiões do cenário, permitindo iniciar o algoritmo genético que considera o cenário todo com solução melhor e buscar apenas refinamentos.
- **Considerar Crescimento Urbano:** A expansão das cidades é um evento natural que pode afetar o desempenho das soluções encontradas a longo prazo. Considerar a expansão horizontal e vertical das cidades durante a alocação pode permitir que a solução obtida se mantenha boa por um período maior.

Referências Bibliográficas

- [1] V. S. Abhayawardhana, I. J. Wassell, D. Crosby, M. P. Sellars, and M. G. Brown. Comparison of empirical propagation path loss models for fixed wireless access systems. In *2005 IEEE 61st Vehicular Technology Conference*, volume 1, pages 73–77 Vol. 1, May 2005.
- [2] A. T. Akabane, R. W. Pazzi, E. R. M. Madeira, and L. A. Villas. Modeling and prediction of vehicle routes based on hidden markov model. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, Sep. 2017.
- [3] Rafael Alves, Igor Campbell, Rodrigo Couto, Miguel Campista, Igor Moraes, Marcelo Rubinstein, Luís Costa, Otto Carlos M. B. Duarte, and Michel Abdalla. *Redes Veiculares: Princípios, Aplicações e Desafios*, pages 199–254. SBRC, 05 2009.
- [4] V. S. Anusha, G. K. Nithya, and S. N. Rao. A comprehensive survey of electromagnetic propagation models. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 1457–1462, April 2017.
- [5] Rafael Kendy Arakaki. Estudos computacionais para o problema de roteamento em arcos capacitado e aberto. Master’s thesis, UNICAMP, 2016.
- [6] B. Ayyappan and P. M. Kumar. Vehicular ad hoc networks (vanet): Architectures, methodologies and design issues. In *2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM)*, pages 177–180, March 2016.
- [7] J. Barrachina, P. Garrido, M. Fogue, F. J. Martinez, J. Cano, C. T. Calafate, and P. Manzoni. Road side unit deployment: A density-based approach. *IEEE Intelligent Transportation Systems Magazine*, 5(3):30–39, Fall 2013.
- [8] Mrinmoy Barua, Xiaohui Liang, Rongxing Lu, and Xuemin (Sheman) Shen. Rcare: Extending secure health care to rural area using vanets. *Mobile Networks and Applications*, 19(3):318–330, Jun 2014.
- [9] N. C. Beaulieu and C. Cheng. Efficient nakagami-m fading channel simulation. *IEEE Transactions on Vehicular Technology*, 54(2):413–424, March 2005.
- [10] Seif Ben Chaabene, Taoufik Yeferny, and Sadok Ben Yahia. A roadside unit placement scheme for vehicular ad-hoc networks. In Leonard Barolli, Makoto Takizawa,

- Fatos Xhafa, and Tomoya Enokido, editors, *Advanced Information Networking and Applications*, pages 619–630, Cham, 2020. Springer International Publishing.
- [11] Guillem Boquet, Ivan Pisa, Jose Lopez Vicario, Antoni Morell, and Javier Serrano. Adaptive beaconing for rsu-based intersection assistance systems: Protocols analysis and enhancement. *Vehicular Communications*, 14:1 – 14, 2018.
 - [12] X. Cao, Q. Cui, S. Zhang, X. Jiang, and N. Wang. Optimization deployment of roadside units with mobile vehicle data analytics. In *2018 24th Asia-Pacific Conference on Communications (APCC)*, pages 358–363, 2018.
 - [13] Evellyn S. Cavalcante, André L.L. Aquino, Gisele L. Pappa, and Antonio A.F. Loureiro. Roadside unit deployment for information dissemination in a vanet: An evolutionary approach. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '12*, pages 27–34, New York, NY, USA, 2012. ACM.
 - [14] Ahmed Chariyev, Tang Low, and MOHAMAD NAUFAL MOHAMAD SAAD. Path loss simulation in different radio propagation models with 1.8 ghz and 2.6 ghz bands. In *Ericsson*, pages 1–5, 06 2014.
 - [15] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao. Vehicle-to-everything (v2x) services supported by lte-based systems and 5g. *IEEE Communications Standards Magazine*, 1(2):70–76, 2017.
 - [16] Felipe Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Viana, Raquel A. F. Mini, and Antonio A. F. Loureiro. Data communication in vanets: Protocols, applications and challenges. *Ad Hoc Networks*, pages 90–103, July 2016.
 - [17] Felipe D. Cunha, Guilherme Maia, Clayson S. F. S. Celes, Bruno P. Santos, Paulo H. L. Rettore, André B. Campolina, Daniel Guidoni, Fernanda Sumika H. Souza, Heitor Ramos, Leandro Villas, Raquel A. F. Mini, and Antonio A. F. Loureiro. Sistemas de transporte inteligentes: Conceitos, aplicações e desafio. Mini curso, 2017.
 - [18] Alan Davis. Statewide connected vehicles. Technical report, Georgia Department of Transportation, 2018.
 - [19] Raí Caetano de Jesus. Formulações e algoritmos para o problema da poligonização de área máxima. Master’s thesis, UNICAMP, 2018.
 - [20] A. M. de Souza, R. S. Yokoyama, G. Maia, A. Loureiro, and L. Villas. Real-time path planning to prevent traffic jam through an intelligent transportation system. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 726–731, June 2016.
 - [21] Allan M. de Souza, Leonardo C. Botega, Islene Calciolari Garcia, and Leandro A. Villas. Por aqui é mais seguro: Melhorando a mobilidade e a segurança nas vias

- urbanas. In *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Porto Alegre, RS, Brasil, 2018. SBC.
- [22] Allan Mariano de Souza and Leandro Aparecido Villas. A fully-distributed traffic management system to improve the overall traffic efficiency. In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '16, pages 19–26, New York, NY, USA, 2016. ACM.
 - [23] R. Doolan and G. M. Muntean. Ecotrec: A novel vanet-based approach to reducing vehicle emissions. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):608–620, March 2017.
 - [24] John J Egli. Radio propagation above 40 mc over irregular terrain. *Proceedings of the IRE*, 45(10):1383–1391, 1957.
 - [25] I. K. Eltahir. The impact of different radio propagation models for mobile ad hoc networks (manet) in urban area environment. In *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*, pages 30–30, 2007.
 - [26] Harald T Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, 1946.
 - [27] Mevlut Garip, Mehmet Gursoy, Peter Reiher, and Mario Gerla. Congestion attacks to autonomous cars using vehicular botnets. In *Workshop on Security of Emerging Networking Technologies*, 01 2015.
 - [28] Github. The state of the octoverse | the state of the octoverse celebrates a year of building across teams, time zones, and millions of merged pull requests. <https://octoverse.github.com/>. Acessado: 2019-12-24.
 - [29] Andrew S Glassner. *An introduction to ray tracing*. Elsevier, 1989.
 - [30] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
 - [31] A. Guerna and S. Bitam. Gica: An evolutionary strategy for roadside units deployment in vehicular networks. In *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–6, 2019.
 - [32] Juan Guerrero-Ibáñez, Sherali Zeadally, and Juan Contreras-Castillo. Sensor technologies for intelligent transportation systems. *Sensors*, 18(4):1212, Apr 2018.
 - [33] X. Hu, X. Li, E. C. H. Ngai, J. Zhao, V. C. M. Leung, and P. Nasiopoulos. Health drive: Mobile healthcare onboard vehicles to promote safe driving. In *2015 48th Hawaii International Conference on System Sciences*, pages 3074–3083, Jan 2015.

- [34] IBGE. Ibge | cidades@ | são paulo | pesquisa | frota de veículos | veículo. <https://cidades.ibge.gov.br/brasil/sp/pesquisa/22/28120>. Accessed: 2020-01-16.
- [35] Wellington Lobato Junior, Denis Rosário, Leandro A. Villas, and Eduardo Cerqueira. Mecanismo para cooperação e coligação de veículos baseado na teoria dos jogos para transmissão de vídeos em vanets. In *Anais do XVI Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, Porto Alegre, RS, Brasil, 2017. SBC.
- [36] Nidhi Chahal Jyoti Jain*. A REVIEW ON VANET, TYPES, CHARACTERISTICS AND VARIOUS APPROACHES. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*, 5(9):239–245, September 2016.
- [37] Oliver Kramer. *Genetic Algorithm Essentials*. Springer, Cham, 679 edition, 2017.
- [38] Lucas Ladeira, Allan Souza, Geraldo Pereira, Thiago Henrique Silva, and Leandro Villas. Serviço de sugestão de rotas seguras para veículos. In *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 608–621, Porto Alegre, RS, Brasil, 2019. SBC.
- [39] J. Liu, J. Wan, D. Jia, B. Zeng, D. Li, C. Hsu, and H. Chen. High-efficiency urban traffic management in context-aware computing and 5g communication. *IEEE Communications Magazine*, 55(1):34–40, January 2017.
- [40] W. Lobato, D. Rosario, M. Gerla, and L. A. Villas. Platoon-based driving protocol based on game theory for multimedia transmission over vanet. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, Dec 2017.
- [41] F. Lyu, H. Zhu, H. Zhou, W. Xu, N. Zhang, M. Li, and X. Shen. Ss-mac: A novel time slot-sharing mac for safety messages broadcasting in vanets. *IEEE Transactions on Vehicular Technology*, 67(4):3586–3597, April 2018.
- [42] Vahid Meghdadi. Ber calculation. *Wireless Communications*, 2008.
- [43] Lucas Henrique Sousa Mello, Flávio M. Varejão, Alexandre L. Rodrigues, and Thomas W. Rauber. Np-hardness of minimum expected coverage. *Pattern Recognition Letters*, 117:45 – 51, 2019.
- [44] Seyedali Mirjalili. *Genetic Algorithm*, pages 43–55. Springer International Publishing, Cham, 2019.
- [45] Moovit. Moovit public transit index. https://moovitapp.com/insights/en/Moovit_Insights_Public_Transit_Index-countries?__hstc=171844735.364e85bdfbee9befe3ae5852a451c058.1579376255994.1579376255994.1579376255994.1&__hssc=171844735.4.1579376255995&__hsfp=4062131045. Acessado: 2020-01-18.
- [46] Douglas LL Moura, Raquel S Cabral, Thiago Sales, and Andre LL Aquino. An evolutionary algorithm for roadside unit deployment with betweenness centrality pre-processing. *Future Generation Computer Systems*, 88:776–784, 2018.

- [47] Zia Nadir and Muhammad Ahmad. Pathloss determination using okumura-hata model and cubic regression for missing data for oman. *Lecture Notes in Engineering and Computer Science*, 2181, 03 2010.
- [48] H. A. Omar, N. Lu, and W. Zhuang. Wireless access technologies for vehicular network safety applications. *IEEE Network*, 30(4):22–26, July 2016.
- [49] J. (. Pan, I. S. Popa, and C. Borcea. Divert: A distributed vehicular traffic re-routing system for congestion avoidance. *IEEE Transactions on Mobile Computing*, 16(1):58–72, Jan 2017.
- [50] Geraldo Pereira, Rodolfo Meneguette, Gustavo Pessin, Jó Ueyama, and Leandro Villas. Speed: Uma solução inter-veicular para detectar e controlar vias congestionadas no tms. In *Anais do III Workshop de Computação Urbana*, pages 43–56, Porto Alegre, RS, Brasil, 2019. SBC.
- [51] Geraldo Pereira, Rodolfo Meneguette, Gustavo Pessin, Jó Ueyama, and Leandro Villas. Speed: Uma solução inter-veicular para detectar e controlar vias congestionadas no tms. In *Anais do III Workshop de Computação Urbana*, pages 43–56, Porto Alegre, RS, Brasil, 2019. SBC.
- [52] Jorge Pereira, Leandro Ricardo, Miguel Luís, Carlos Senna, and Susana Sargento. Assessing the reliability of fog computing for smart mobility applications in vanets. *Future Generation Computer Systems*, 94:317 – 332, 2019.
- [53] Said Salhi, Richard Eglese, and Eleni Hadjiconstantinou. Preface: advances in theoretical and practical combinatorial optimization. *Annals of Operations Research*, 272(1):1–2, Jan 2019.
- [54] C. Sommer, D. Eckhoff, R. German, and F. Dressler. A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments. In *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*, pages 84–90, Jan 2011.
- [55] C. Sommer, S. Joerer, and F. Dressler. On the applicability of two-ray path loss models for vehicular network simulation. In *2012 IEEE Vehicular Networking Conference (VNC)*, pages 64–69, Nov 2012.
- [56] N. Sumon and A. Jaekel. Fuel efficient route planning using vanet. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 468–474, Nov 2018.
- [57] A. Taha and N. AbuAli. Route planning considerations for autonomous vehicles. *IEEE Communications Magazine*, 56(10):78–84, OCTOBER 2018.
- [58] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas. Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075, May 2014.

- [59] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Simutools '08, pages 60:1–60:10, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [60] Leandro A. Villas, Heitor S. Ramos, Azzedine Boukerche, Daniel L. Guidoni, Regina B. Araujo, and Antonio A.F. Loureiro. An efficient and robust data dissemination protocol for vehicular ad hoc networks. In *Proceedings of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '12, pages 39–46, New York, NY, USA, 2012. ACM.
- [61] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325 – 344, 2014.
- [62] Lixia Xue, Yuchen Yang, and Decun Dong. Roadside infrastructure planning scheme for the urban vehicular networks. *Transportation Research Procedia*, 25:1380 – 1396, 2016. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.
- [63] Michel Daoud Yacoub. *Foundations of mobile radio engineering*. Routledge, 2019.
- [64] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu. Machine learning for vehicular networks: Recent advances and application examples. *IEEE Vehicular Technology Magazine*, 13(2):94–101, June 2018.

Apêndice A

Trabalho Publicado no SBPO 2019

Genetic Algorithm for the Maximum Coverage Location Problem Applied to Medical Emergency

Matheus Ferraroni Sanches, Wellington Lobato Junior
UNICAMP

Cidade Universitária Zeferino Vaz - Barão Geraldo, Campinas - SP
[ferraroni, wellington]@lrc.ic.unicamp.br

Fábio Luiz Usberti, Leandro Aparecido Villas
UNICAMP

Cidade Universitária Zeferino Vaz - Barão Geraldo, Campinas - SP
[fusberty, leandro]@ic.unicamp.br

RESUMO

Com o crescimento das áreas urbanas, vários serviços precisam ser atualizados para manter uma boa qualidade. Uma área profundamente afetada são os Serviços de Emergência Médica (EMS), que se não for bom e rápido pode resultar em perdas de vidas. A localização onde os veículos de emergência estão tem uma influência significativa no tempo levado para que ele chegue até o local de uma emergência. Em ambientes complexos, como os de grandes cidades, a localização pode ter um impacto ainda maior no tempo gasto, já que congestionamentos e bloqueios podem rapidamente aumentar esse tempo. Este trabalho propõe uma abordagem que usa algoritmos genéticos para selecionar a localização das estações bases, onde veículos de emergência podem ficar, considerando informações de tráfego para determinar qual a área de alcance a partir de suas bases.

PALAVRAS CHAVE. Algoritmos Genéticos, Problema de Cobertura Máxima, Otimização.

Tópicos: PO na Área da Saúde

ABSTRACT

With the growth of urban areas, various services need to be updated in order to keep attendance with good quality. One profoundly affected area is the Emergency Medical Services (EMS), which if the service is not good and fast may result in loss of life. The location where the emergency vehicles are positioned has a significant influence on the time taken by the emergency vehicle to arrive at an emergency location. In a complex environment, as the large cities, the location may have an even more impact on that time, as congestion levels and road blocks may quickly increase the time taken. This work proposes an approach using a genetic algorithm to select the location of the standby sites where the emergency vehicles can stay, considering traffic information to determine the area that these vehicles can reach in a given time.

KEYWORDS. Genetic Algorithm, Maximum Coverage Problem, Optimization.

Paper topics: PO in the Health Area

1. Introduction

Emergency Medical Services organizations play an important role in modern health systems, and their ability to accurately respond to emergency calls can have a significant impact on patients health and recovery Andersen [2019]. EMS are critical elements of modern health systems, and they are responsible for the pre-hospital component of health systems. With the growth of urban centers and the number of vehicles present in them, the demands in various sectors are affected, such as food, safety, productivity, and mobility areas.

The technologies present in cities are being adopted more, these technologies range from a GPS sensor, capable of providing vehicle location in real time, to intelligent traffic management systems capable of improving traffic at peak times through intelligent traffic light control Kang et al. [2014].

Existing transport systems can take advantage of sensors available on vehicles and roads to gather information and make smarter decisions, such as selecting faster routes de Souza et al. [2018]. The set of subsystems that operate in the transport system is known as Intelligent Transport System (ITS) and offers advantages such as less polluting traffic, reduced congestion, and faster travel Cunha et al. [2017]. Services such as emergency management can take advantage of ITS to move their vehicles faster in cities.

Recent studies on the management of EMS show that there are three main areas: standby sites location, emergency vehicle reallocation, and emergency vehicle dispatch Bélanger et al. [2019]. The vehicle dispatch tries to find the best vehicle to dispatch to a specific demand. The reallocation problem is concerned about changing the vehicles between the standby sites to better attend the demands. The static location problem tries to find the best places to set standby sites in a given area.

The increasing urban growth raises essential needs to the cities, related to traffic congestion, safety, and mobility. The safety in the cities is a demand that is profoundly affected by diverse factors. The position of the standby sites where the emergency vehicles are parked has a significant impact on the time spent to arrive at an emergency location. If the transit on the route used by the emergency vehicle dispatched has a high level of congestion, this time can increase significantly.

This work proposes the result of using a Genetic Algorithm (GA) to deal with the static location problem, which is by itself an NP-hard problem [Farahani e Hekmatfar, 2009]. The data used to calculate the best location for the standby sites is their respective coverage area. Also, the areas that can be reached from the standby site are not circles or square areas, but a polygon that considers traffic information about a busy day at the rush hour. The polygonal coverage of each site increases the complexity of the calculations considerably, since if a standby site is moved a few meters from its original position the polygon may be substantially different.

The use of optimization models solves problems like the maximum coverage location are a good way to face them. During this work, we use a model inspired on evolution, the Genetic Algorithm. These kinds of algorithms store the important information in a data structure similar to a chromosome and apply different combinations and mutations in order to improve how good one element can solve the problem faced Whitley [1994].

The remainder of this paper is structured as follows. Section II outlines related works from the literature. Section III describes how the methodology was implemented and how it operates. Section IV discusses the scenario, evaluation and results obtained. Section V gives the conclusions of this paper.

2. Related work

Several works in the lasts years have been developed in order to improve how emergencies are handled, and efforts to reduce waiting time are possible through a better allocation of resources.

Toregas et al. [1971] was the first one to approach the emergency vehicle location considering the coverage. In this work, an area is considered covered if it can be reached by at least one vehicle within a time or distance. The main goal of this work is to minimize the number of vehicles that are used to cover the defined area. To perform its calculation, this work uses integer linear programming fed by a matrix of the minimum distance between any two locations.

In Nickel et al. [2016], the authors investigate the problem of choosing the location and number of ambulances and their bases in a specific region. The paper also shows that the possibility of considering just one sample of scenarios can lead to a misleading solution. The main goal of the work is to minimize the total cost for maintaining and installing these facilities but assuring a minimum coverage level. However, these works do not consider real information about the city, which may alter the results. Moreover, such approaches do not take into account the characteristics of the scenario.

Liu et al. [2016] introduce a new Double Standard Model (DSM) for EMS that works along with a GA to solve vehicle allocation problem that ensures acceptable service reliability with limited vehicle resources. The authors used DSM to deal with different types of vehicles and different priorities for the demands, using data from vehicles crashes in Chicago from 2004 to 2010. The proposed model requires that any demand site is covered by at least one EMS vehicle and the secondary standard with a portion of the covered sites achieving the service reliability requirement. This work aims to maximize the demand sites covered within the first standard meeting the required service reliability. However, Liu et al. only consider a limited quantity of standby sites, which means that the approach did not allocate enough resources to cover all the city. Besides, this approach did not evaluate the real behavior of the scenario.

Zhang e Li [2015] proposes a probabilistic model with chance constraints for locating and sizing emergency medical service stations. In the proposed model, the chance constraints are approximated as second-order cone constraints to overcome computational difficulties for practical applications. The proposed approximations associated with different estimation accuracy of the stochastic nature are meaningful on a practical uncertainty environment.

Based on the analysis of the related work, we conclude that it is essential to consider an approach using genetic algorithms to allocate the facilities in the best positions to reduce waiting time. This involves analyzing the behavior of the different data collected and optimizing the allocation based on the possible routes.

3. Genetic Algorithm Optimization

In this Section, we describe the proposed methodology, a greedy solution for this problem and the real world solution. In Section 3.1, we present the problem formulation for the maximum coverage location problem and how is the data acquisition from the city of Chicago. Section 3.2 describes the greedy approach employed to generate the initial population and also used to compare with the genetic-approach. Section 3.3 presents the genetic algorithm approach applied in the data acquired and how it selects the best individuals, in each generation.

3.1. Problem Formulation

The formulation assumes that a standby site can be placed anywhere in the city that has a road. This simplifies the process of checking if a location is available to set the standby site and allow the main goal of this work to be better explored. The number of emergency vehicles that should stay in each of the chosen locations is not calculated at this moment, only the area that can be covered is considered.

An area is considered covered if it can be reached by at least one standby site. If the same area can be reached by more than one standby site, this area is considered as overlap. Areas that

cannot be reached with the times used in this work were not removed, which means that lakes and parks are considered during the calculation of the coverage. Not removing these areas means that the coverage of 100% of the desired area is not possible, but this keeps the scenario closer to the real world. The number of standby sites is considered a decision variable in the formulation.

3.2. Greedy Approach

The greedy solution developed in this project checks all possible locations, and it attempts to fit a standby site. If there is a location with no overlap, it is selected. The solution found by this approach was enough to achieve coverage between 40% and 60%. Another approach, which randomly add standby sites and removed the ones with overlap, were also tested, but on the comparison between these greedy solutions, the first one provided a better solution and was chosen to compare with the proposed genetic algorithm.

3.3. Proposed Genetic Algorithm

As soon as the solution is initialized, there is a process of coordinate conversion which is done in order to simplify the mutation stage. As soon as this conversion is completed, the first pool can be created. Until this moment, the amount of standby sites needed to cover the city is unknown. To determine how many standby sites the first pool should be created, a simple greedy algorithm is used. This greedy algorithm keeps selecting locations to insert until there is no place left that can be selected without overlap. Once the element generated with the greedy algorithm is complete, a pool of random elements is generated using as a base the number of standby sites determined by the greedy algorithm. The greedy element is inserted into the pool with the new random elements.

At this point, the generation loop is initialized. If it is the first iteration, the pool will contain only the random elements and one element generated by one greedy algorithm. If it is one of the following generations, the pool will contain a percentage of new random elements and a percentage of elements generated based on previous generations.

The first step of the generation loop is to check if the stop criteria is satisfied. If this criteria is satisfied the execution is terminated. In this work, the stop criteria was set as 300 iterations of the generation loop.

The following step is to evaluate the fitness of each element. As the objective is to cover as much as possible with the least overlap as possible, the first step is to determine each standby site covered. This calculation can be viewed in line 24 of Algorithm 1, where the final area is divided by the initial area. The polygon containing the area of the city is restarted for every fitness calculation; this way the polygon can be updated and cropped, which is done every calculation. Once the coverage of a polygon is calculated, this area is subtracted from the area to cover, in this way areas that cover the same place are not counted.

The overlap between the standby sites in the same element is also computed and can be view in line 19 of Algorithm 1. The polygons in this calculation are not altered, as they will always be the same for the same standby site in the time used.

Once the coverage by the standby sites and their overlap are calculated, the fitness can be assigned to the solution. As shown in line 25 of Algorithm 1, if the overlap between the polygons is higher than the coverage, the fitness of that solution may be negative. As the first generation guarantees at least one solution with positive fitness, generated by the greedy algorithm, there will be at least one solution with positive fitness and no overlap. The parent selection guarantees this property in the new generation step.

For the parents' selection and the generation of a new pool of elements, the elements in the actual pool are ordered by higher fitness to low, which can be seen in line 6 of the Algorithm 2. Half of the elements with lower fitness score are eliminated from the next steps, in line 7, in order

Algorithm 1: Optimization Of The Maximum Maximum Coverage Problem

```
1 Input
2 areaToCover  $\leftarrow$  ChicagoBoundaries
3 K  $\leftarrow$  Time to consider that can be reached from a standby site
4 Output
5 bestSolutionFound is the best solution found during the generations.
6 begin
7   timeReach  $\leftarrow$  K
8   areaToCover  $\leftarrow$  coordinateConversion(ChicagoBoundaries)
9   greedyElement  $\leftarrow$  generateGreedySolution(timeReach)
10  amtStandbySites  $\leftarrow$  greedyElement.AmountStandbySites
11  pool  $\leftarrow$  generateFirstPool(amtStandbySites)
12  pool.add(greedyElement)
13  penalty  $\leftarrow$  PenaltyRate
14  while not (stopCriteria.satisfied) do
15    for element in pool do
16      areaToCover  $\leftarrow$  areaOfChicago
17      polygons  $\leftarrow$  getPolygonsFromElement(element)
18      n  $\leftarrow$  polygons.size
19      overlap  $\leftarrow$  ( $\sum_{i=1}^n \sum_{j=i+1}^n \text{overlap}(\text{polygons}_i, \text{polygons}_j)$ )
20      overlap  $\leftarrow$   $\frac{\text{overlap} * \text{penalty}}{n}$ 
21      while polygons  $\neq \emptyset$  do
22        areaToCover.updateArea(polygons[0])
23        polygons.pop(0)
24      coverage  $\leftarrow$   $\frac{\text{areaToCover.finalArea}}{\text{areaToCover.initialArea}}$ 
25      element.score  $\leftarrow$  coverage-overlap
26    pool  $\leftarrow$  createNewGeneration(pool)
27  bestSolutionFound  $\leftarrow$  getBestElement(pool)
```

to increase the mean fitness score and try to let the best elements to be used as the base for the next generations.

The rest of the elements are used to generate the next pool. Using the randomness rate, defined at the initialization, is possible to determine how many elements are generated with a completely random set of standby sites and how many must inherit from the actual generation. During the development of this work, the randomness rate was defined as 15%, which means that 15% of the elements in each generation, after the first one, will be randomly generated. Having a percentage of new random elements in each generation is a good way to insert variability in the current pool.

The remaining 85% of the elements that should be part of the next pool will inherit from an element of the current pool. To select which element will be used as the base to new ones, a probability vector was defined, which guarantee that the greater the fitness of an element, the higher the chance that this element will be used to generate new elements. In line 9 of Algorithm 2 a vector of probability is created to select these elements.

Once a new element is generated inheriting the genome from another element, the entire genome is copied and will be mutated later. There is a chance that the number of standby sites changes during this process. It can be randomly increased by one or decreased by one, as it can be viewed in line 14 of Algorithm 2. If the number of standby sites is increased by one, a random standby site is added at the end of the genome. If the number of standby sites is decreased, a standby site is removed from the genome, which may be a random base station or the standby sites with most overlap with other standby sites at that element.

When the number of standby sites changes from the parent, a secondary element is created, and this secondary element will not endure mutation. This is done in order to evaluate the parent element with change restricted to the number of base stations. This process can be seen in lines 18 to 20 of Algorithm 2.

The best elements from a generation are copied to the new generation, in order to guarantee that the best solutions found so far keep being considered during the next generations. This process can be seen in lines 23 and 24 of Algorithm 2.

Algorithm 2: Creation Of New Generation

```

1 begin
2   pool ← pool of elements from actual generation
3   newPool = []
4   extend ← 1 - randomness rate
5   limitElements ← Total elements per generation
6   pool.orderByScore()
7   pool.discardWorseHalf()
8   n ← pool.size
9   distribution ← [n, n - 1, n - 2, ..., 1] ·  $\frac{1}{\sum_{i=1}^n i}$ 
10  i ← 0
11  while i < limitElements*extend do
12    i ← i+1
13    parent ← pickRandom(pool, distribution)
14    newStandbySites ← parent.standbySites + randomInt(-1,1)
15    element1 ← new Element(parent,newStandbySites)
16    element1.mutationActive()
17    newPool.add(element1)
18    if not (parent.standbySites==newStandbySites) then
19      element2 ← new Element(parent,newStandbySites)
20      newPool.add(element2)
21    while newPool.size() < limitElements*0.1 do
22      newPool.add(new Element(None, pool[0]))
23    i ← 0
24    while limitElements*0.1 do
25      newPool.add(pool[i]) i ← i+1
26  pool ← newPool

```

4. Computational Experiments

This Section presents the evaluation of the proposed genetic algorithm based solution for allocating standby sites. We evaluated the impact of different standby sites locations for better coverage while trying to reduce the overlap between the covered area by more than one standby site. We analyzed how the proposed approach works differently concerning the time required to reach the standby site. Moreover, we provide the optimal number and positions of standby sites for the city of Chicago.

4.1. Scenario Description

We used the API from HERE¹ to acquire the information used to calculate how far a vehicle can get from a standby site in a given time, considering the data of Chicago as a base scenario. The city of Chicago has been chosen since different data are available on this. All the data used is provided by the city hall. This allows us to reproduce the real behavior of the city dynamics according to empirical data. We conducted 300 iterations of the generation loop, and the randomness rate used during the tests was fixed at 15%. We set different parameters related with mutation rate during generations, time to reach from a standby site and seed. The programming language adopted was Python 2.7.

The data used in the experiments consider one day, reflecting the rush hour usually presented in big cities such as Chicago. We used four different times to compute the area that can be reached from a standby site. All the main simulation parameters are summarized in Table 1.

Table 1: Genetic Algorithm Parameters

Parameters	Value
Initial population	60
Selection	Rank-based selection
Encoding	Binary
Penalty rate	1000
Amount of elements	60
Mutation Rate	From 0.25 to 0.0075
Randomness rate	15%
Iteration limit	300
Time to reach an event (minutes)	4,5,6 and 7
Starting standby sites	Defined by greedy solution
Amount of elements discarded	50%
Amount of best elements reinserted	10%

4.2. Results

The allocation of standby sites, after the application of the genetic algorithm, can be visualized in Figure 1. In each image the result obtained after the interactions carried out by the methodology, changing only the minutes of reach from a standby site. It is possible to notice the relation between time and quantities of standby sites. As the number of standby sites decreases, the higher is the time reaching. Figure 1(a) is the one with most standby sites, as this one has the smaller time used. Figures 1(b) and 1(c) show that the amount of standby sites decreased quickly. Figure 1(d) is the one with the fewer standby sites, as it has the largest time between the tests performed.

According to the evolution presented in Figure 2 it is possible to see the changes during the generation phase. The evolution of the coverage during the generations is presented in Figure 2(a). This value is the mean coverage of the best creature of every seed used in the same generation.

¹www.here.com

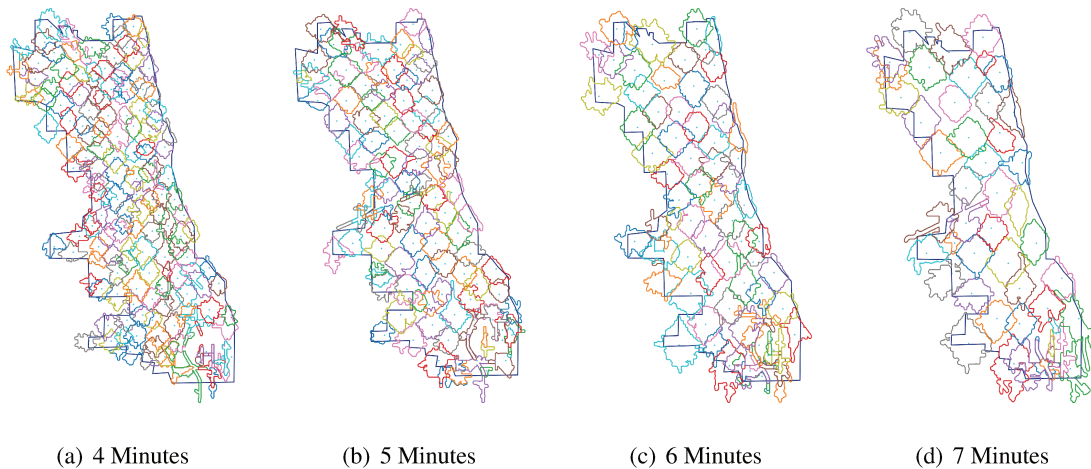


Figure 1: Different time to reach from the standby site

The starting coverage of all the scenarios was close to 35%, and the final values are close to 8%. A coverage close to 100% is a hard thing to accomplish with the times used, as the city has parks and lakes that can not be covered with the times used. Furthermore, when the time used in the scenario is more significant, the evolution tends to occur faster in the beginning, this can be explained by the addition of new standby site having a higher influence in the coverage calculation.

The changes in the amount of standby site are represented in Figure 2(b). Unlike the coverage, when the time used in the scenario is greater, the number of standby site tend to be reduced at the beginning. As this is defined by a greedy algorithm, with a smaller time being used, the polygons generated are smaller allowing more candidate polygons to be chosen for the same area. This results in a higher number of standby sites at the beginning for almost the same coverage. This difference is also reinforced at the end of the generations when the scenario with 4 minutes needs more than three times the standby sites than the scenario with 7 minutes to achieve almost the same Chicago coverage.

The overlap between the standby sites is one important metric, as it is used to calculate the fitness for each element in the pool. This information is represented in Figure 2(c). The initial overlap has a small variation, smaller than 7%. During the generations, the overlap changes quickly, but during all the generations it is smaller than 10% and tends to be minimized at the last generations. The minimization at the last generations is due to the optimization performed by the proposed methodology. Once the number of standby sites used is enough, small changes may help to reduce the overlap and increase fitness. The best fitness along with the generations never reduces, as the best element from the prior generation is copied to the actual generation. This behavior can be seen in Figure 2(d).

In Figure 3 the coverage of Chicago is compared between the chosen approaches, varying just the time to reach from the standby site. The coverage achieved by the proposed solution can keep close to 80% in almost all times. The real world has lower coverage when the time used is 4 minutes, but shows good performance when the times are 6 or 7 minutes. The greedy solution presented a lower fitness, with a significant difference, in all scenarios. To achieve 100% of coverage is a hard, if not impossible, task, as there are areas that are not covered within the times considered in the experiments.

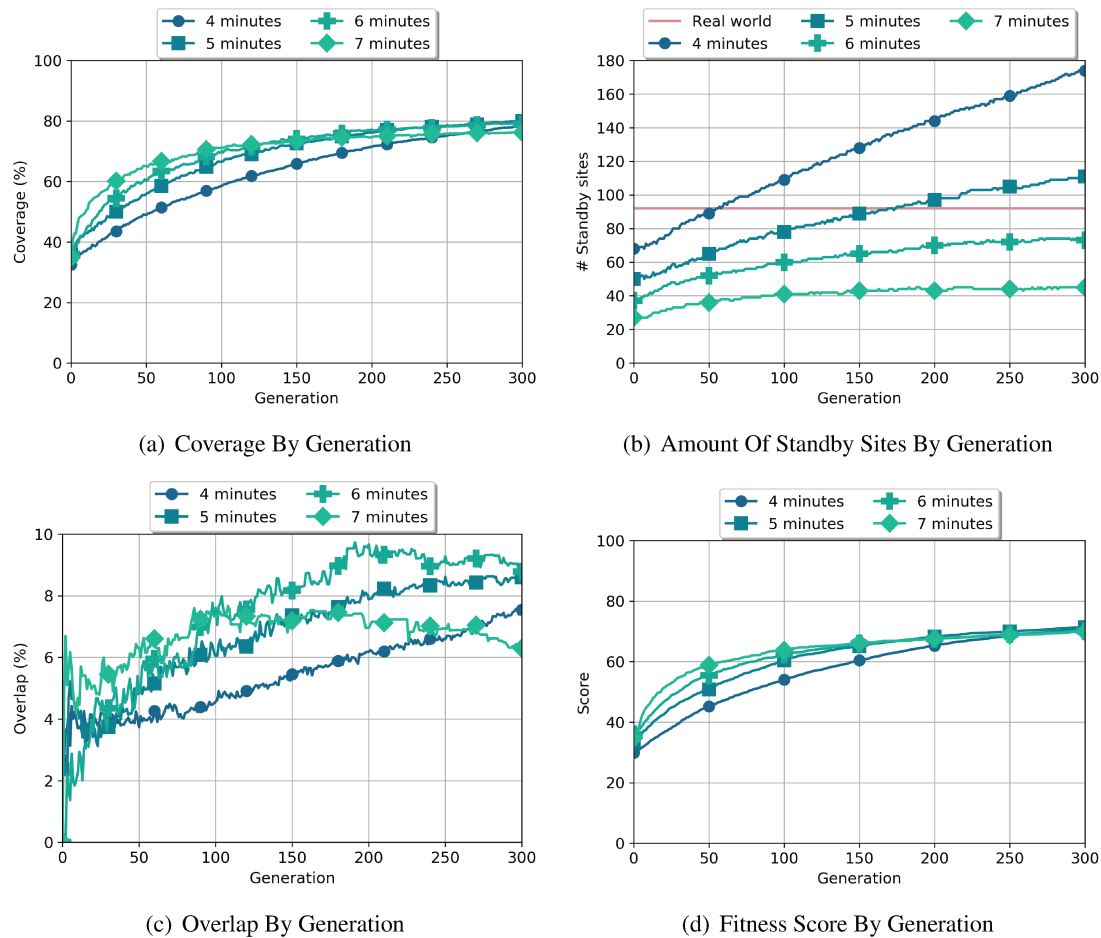


Figure 2: Evolution Over Generations

Figure 4 presented the overlap between the standby sites. The greedy solution was set to have no overlap during its calculation, so its data is not plotted. The mean average of the proposed solution is smaller than 10% for all scenarios, which is close to the real world only in the 4 minutes calculation. When the time increases, the real world has a quick increase in the overlap, achieving total overlap in the 7 minutes calculation. Even though the real world may show a slight improvement in the coverage with 6 and 7 minutes, the overlap increases way more than the proposed solution.

Alongside with coverage and overlap, it is possible to evaluate the number of standby sites used by each solution to achieve the presented results. Figure 5 shows the amount of standby sites used by the solutions. The data presented about the real world is fixed, as they are already in position. The data about the greedy solution is smaller in almost all scenarios evaluated, as this solution was set to have no overlap. The proposed solution starts with a higher number, but decrease quickly. Even though the number of standby sites was significantly smaller than the real world, this solution was able to keep a high coverage and slow overlap.

From our performance evaluation analysis, we identified that our genetic algorithm approach delivered standby sites with 17% more coverage of the city than the real disposal of facilities, and also reduced the overlap between the standby sites up to 95%. When the proposed approach is

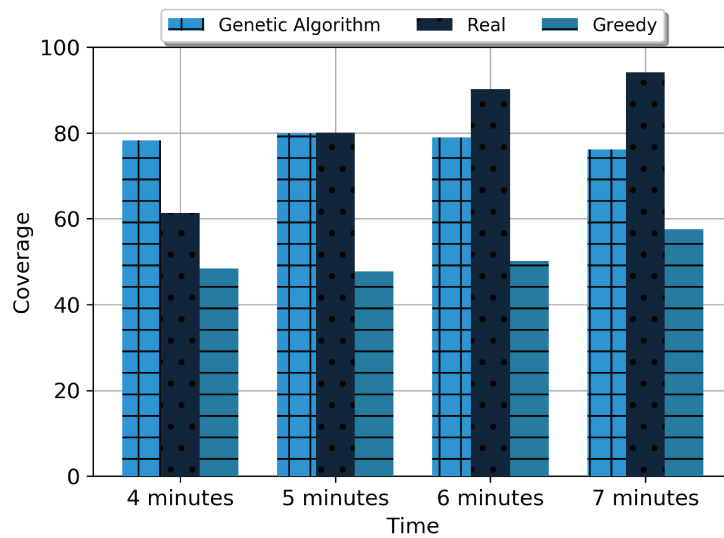


Figure 3: Coverage Comparison

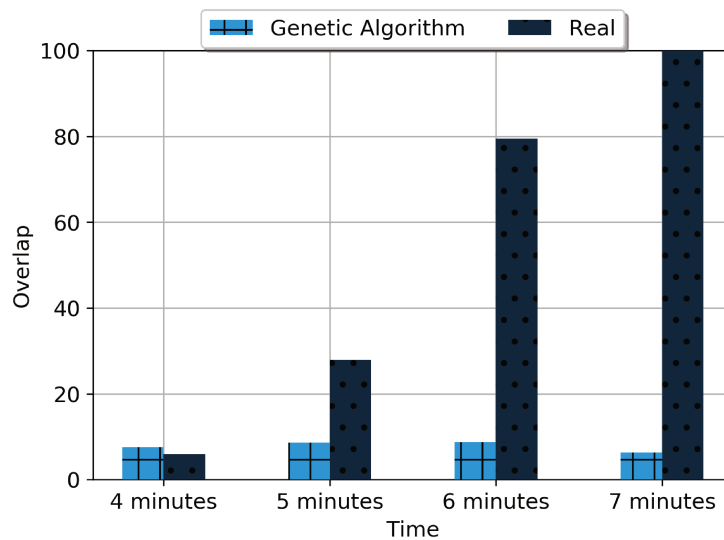


Figure 4: Overlap Comparison

compared with other solutions, the coverage difference increases even more, which occurs because the greedy solution is not capable of optimizing the overlap.

5. Conclusion and Future Work

This work introduced an approach using a genetic algorithm that considers traffic information to select the best position to allocate standby sites for ambulances in the city of Chicago. The genetic algorithm was able to consider the coverage polygons in order to fit them into the city boundaries while reducing the overlap between them. From our performance evaluation analysis, we identified that the approach solution was able to increase the cover of the city up to 17% more than the real disposal of standby sites, and also reduce the overlap between the standby sites up to 95%. This is because the genetic algorithm took into account the different characteristics of the collected data to allocate the standby sites in the most suitable places. The results show the efficiency

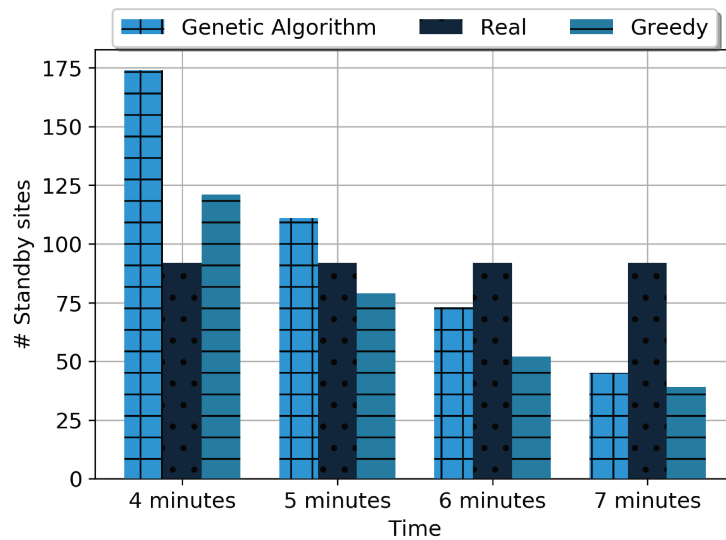


Figure 5: Amount of Standby Sites Comparison

of the approach compared to the real world concerning ensuring low servicing times for the health units.

As future works, we aim to consider a higher amount of data to compute the area that can be reached from a standby site within a given time. Also, a context-aware approach should be investigated, considering regions with different levels of crashes or even more populated regions. We also intend to evaluate the evolution of this approach in different scenarios, like other cities.

References

- Andersen, B. M. (2019). Ambulances, emergency medical service (ems) and other transports of patients. In *Prevention and Control of Infections in Hospitals*, p. 973–983. Springer.
- Bélanger, V., Ruiz, A., e Soriano, P. (2019). Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. *European Journal of Operational Research*, 272(1):1–23.
- Cunha, F. D., Maia, G., Celes, C. S. F. S., Santos, B. P., Rettore, P. H. L., Campolina, A. B., Guidoni, D., Souza, F. S. H., Ramos, H., Villas, L., Mini, R. A. F., e Loureiro, A. A. F. (2017). Sistemas de transporte inteligentes: Conceitos, aplicações e desafio. Mini curso.
- de Souza, A. M., Pedrosa, L. L. C., Botega, L. C., e Villas, L. (2018). itssafe: an intelligent transportation system for improving safety and traffic efficienc. *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. ISSN 2577-2465. URL <https://doi.org/10.1109/VTCSpring.2018.8417760>.
- Farahani, R. Z. e Hekmatfar, M. (2009). *Facility location: concepts, models, algorithms and case studies*. Springer.
- Kang, W., Xiong, G., Lv, Y., Dong, X., Zhu, F., e Kong, Q. (2014). Traffic signal coordination for emergency vehicles. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, p. 157–161.

- Liu, Y., Li, Z., Liu, J., e Patel, H. (2016). A double standard model for allocating limited emergency medical service vehicle resources ensuring service reliability. *Transportation research part C: emerging technologies*, 69:120–133.
- Nickel, S., Reuter-Oppermann, M., e Saldanha-da Gama, F. (2016). Ambulance location under stochastic demand: A sampling approach. *Operations Research for Health Care*, 8:24–32.
- Toregas, C., Swain, R., ReVelle, C., e Bergman, L. (1971). The location of emergency service facilities. *Operations research*, 19(6):1363–1373.
- Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85. ISSN 1573-1375. URL <https://doi.org/10.1007/BF00175354>.
- Zhang, Z.-H. e Li, K. (2015). A novel probabilistic formulation for locating and sizing emergency medical service stations. *Annals of Operations Research*, 229(1):813–835.