

# POSICIONAMENTO DE INFRAESTRUTURAS DE COMUNICAÇÃO EM REDES VEICULARES: Algoritmos para a Maximização da Área da Rede Delta<sup>1</sup>

Vinícius Henrique Silveira<sup>2</sup>  
Prof. Dr. João Fernando Machry Sarubbi<sup>3</sup>

Submetido em: 28/05/2026 | Aprovado em: 09/06/2026

## RESUMO

O problema de posicionamento de infraestruturas em redes veiculares ad hoc (VANETs) consiste em determinar a localização de Roadside Units (RSUs) para maximizar a conectividade sob restrições de implantação. Abordagens tradicionais avaliam a conectividade de forma binária, desconsiderando a duração da conexão ao longo da viagem. A métrica Delta permite uma avaliação mais abrangente ao relacionar a fração do tempo de conexão dos veículos ( $\rho_1$ ) com a fração de veículos atendidos ( $\rho_2$ ). Neste trabalho, o problema é abordado pela maximização da área sob essa curva, considerando um número fixo de RSUs. Para sua resolução, são desenvolvidos um algoritmo guloso e um algoritmo baseado na metaheurística GRASP, cujos resultados são comparados com os obtidos pelo modelo exato resolvido via CPLEX. Os resultados mostram que ambos os algoritmos alcançam soluções ótimas em instâncias pequenas. Para uma instância realística de mobilidade veicular urbana, ambos os algoritmos produzem soluções equivalentes em todos os cenários avaliados. Experimentos sobre uma instância sintética com distribuição de tráfego mais uniforme revelam que o algoritmo baseado na metaheurística GRASP supera o algoritmo guloso na maioria dos cenários, com ganhos de até 5,10%, sugerindo que a diversificação proporcionada pelo GRASP é mais eficaz em cenários com múltiplas regiões promissoras.

**Palavras-chave:** VANET; posicionamento de RSUs; heurísticas.

## ABSTRACT

The placement of communication infrastructure in Vehicular Ad Hoc Networks (VANETs) consists of determining the location of Roadside Units (RSUs) to maximize connectivity under deployment constraints. Traditional approaches evaluate

---

<sup>1</sup> Artigo científico elaborado para o trabalho de conclusão do curso de Engenharia de Computação do campus Nova Gameleira do Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG.)  
<sup>2</sup> Graduando em Engenharia de Computação no campus Nova Gameleira do Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG.). vhsilveira2@gmail.com  
<sup>3</sup> Orientador do trabalho. Docente do Departamento de Computação do campus Nova Gameleira do Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG.). joasarubbi@gmail.com

connectivity in a binary manner, disregarding its duration throughout the trip. The Delta metric provides a more comprehensive evaluation by relating the fraction of travel time during which vehicles remain connected ( $\rho_1$ ) to the fraction of vehicles that achieve this connectivity level ( $\rho_2$ ). In this work, the problem is addressed by maximizing the area under this curve, considering a fixed number of RSUs. A greedy algorithm and a GRASP-based metaheuristic algorithm are developed, and their results are compared against those obtained by the exact model solved using CPLEX. Results show that both algorithms achieve optimal solutions in small instances. For a realistic urban vehicular mobility instance, both algorithms produce equivalent solutions across all evaluated scenarios. Experiments on a synthetic instance with more uniform traffic distribution reveal that the GRASP-based metaheuristic algorithm outperforms the greedy algorithm in most scenarios, with gains of up to 5.10%, suggesting that the diversification provided by GRASP is more effective in scenarios with multiple promising regions.

**Keywords:** VANET; RSU placement; heuristics.

## 1 INTRODUÇÃO

As redes veiculares ad hoc, conhecidas como VANETs (*Vehicular Ad Hoc Networks*), desempenham um papel fundamental no contexto de sistemas inteligentes de transporte (*Intelligent Transportation Systems* – ITS), viabilizando aplicações como monitoramento de tráfego, segurança veicular e serviços de informação em tempo real (Blum *et al.*, 2004). Nessas redes, veículos são capazes de se comunicar entre si e com elementos de infraestrutura, formando um ambiente altamente dinâmico, caracterizado por elevada mobilidade e variações constantes na conectividade.

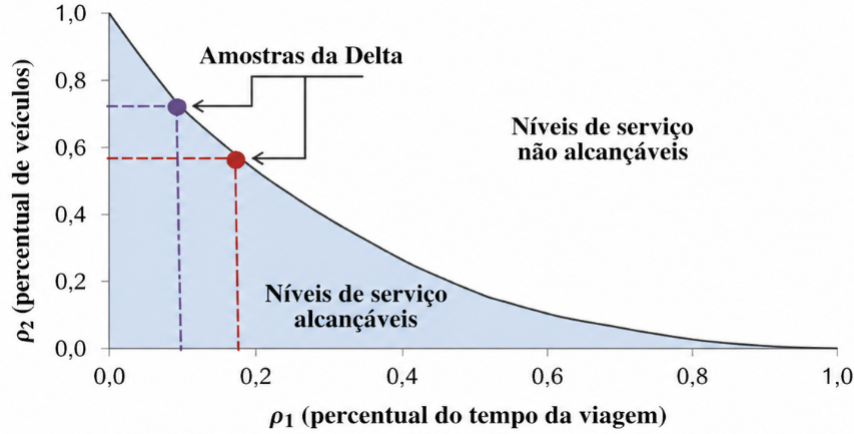
A comunicação em VANETs pode ocorrer de duas formas principais: *Vehicle-to-Vehicle* (V2V), na qual a troca de informações ocorre diretamente entre veículos, e *Vehicle-to-Infrastructure* (V2I), que envolve a comunicação entre veículos e unidades fixas de infraestrutura, denominadas *Roadside Units* (RSUs). Embora a comunicação V2V permita a disseminação descentralizada de informações, a abordagem V2I apresenta maior robustez, especialmente em cenários com baixa densidade veicular ou elevada dispersão espacial (Hartenstein; Laberteaux, 2008; Garcia *et al.*, 2021). No entanto, a implantação de RSUs envolve custos significativos, o que impõe restrições ao número de unidades que podem ser instaladas.

Diante dessas limitações, surge o problema de posicionamento de RSUs, cujo objetivo é determinar a melhor distribuição dessas unidades de forma a maximizar a qualidade da comunicação na rede. Tradicionalmente, esse problema tem sido abordado sob a perspectiva de cobertura, frequentemente modelado como uma variação do Problema de Máxima Cobertura (Church; ReVelle, 1974), no qual se busca atender o maior número possível de veículos a partir de um conjunto limitado de pontos candidatos. Entretanto, tais abordagens tendem a considerar a conectividade de forma binária, verificando apenas a existência de contato entre veículos e infraestrutura, sem levar em conta a duração ou a continuidade dessa conexão ao longo da viagem.

Com o intuito de avaliar a qualidade de serviço em redes veiculares de forma abrangente, foi proposta a métrica Delta (Silva; Meira Jr., 2015a), que considera simultaneamente dois parâmetros:  $\rho_1$ , representando a fração do tempo de viagem em que um veículo permanece conectado à infraestrutura, e  $\rho_2$ , correspondente à fração de veículos

que atendem a esse nível de conectividade. A partir desses parâmetros, a qualidade global da rede pode ser representada por uma curva no plano  $(\rho_1, \rho_2)$ , cuja área associada fornece uma medida agregada do desempenho da rede.

Figura 1 – Representação do gráfico da rede Delta  $\Delta_{\rho_1}^{\rho_2}$ , ilustrando a relação entre o percentual de tempo de cobertura dos veículos ( $\rho_1$ ) e o percentual de veículos atendidos ( $\rho_2$ ).



Fonte: Adaptado de Silva e Meira Jr. (2015b).

Na literatura, trabalhos que utilizam a métrica Delta abordam o problema de posicionamento de RSUs sob a ótica da minimização da infraestrutura necessária para atender requisitos específicos de qualidade de serviço, como os de (Sarubbi; Silva, 2016), (Silva *et al.*, 2016) e (Sarubbi *et al.*, 2017), por meio de abordagens heurísticas e metaheurísticas.

Diferente das abordagens anteriores, este trabalho adota a maximização da área sob o gráfico da rede Delta como critério central para o problema de posicionamento de RSUs, considerando uma quantidade fixa de unidades de infraestrutura. Essa abordagem permite avaliar de maneira mais abrangente o desempenho da rede, ao considerar simultaneamente diferentes níveis de conectividade, sem a necessidade de definir previamente requisitos específicos de serviço.

Este trabalho se posiciona como uma continuação do trabalho de conclusão de curso de Marques (2020), que abordou o mesmo problema de maximização da área do gráfico da rede Delta utilizando um algoritmo genético. Os algoritmos desenvolvidos no presente trabalho são avaliados sobre a mesma instância utilizada naquele trabalho, o *Vehicular Mobility Trace* da cidade de Colônia, Alemanha (Uppoor *et al.*, 2014), uma base de dados de mobilidade veicular urbana de larga escala amplamente utilizada na literatura de VANETs, permitindo uma comparação direta entre as abordagens. Adicionalmente, experimentos sobre uma instância sintética são conduzidos para investigar o comportamento dos algoritmos em cenários distintos.

Para a resolução do problema, são exploradas diferentes estratégias. O modelo matemático de programação inteira mista proposto por (Silva *et al.*, 2023), resolvido com o auxílio do solver CPLEX, é utilizado como referência para instâncias de pequeno porte. Para instâncias de maior escala, são desenvolvidas abordagens heurísticas, incluindo um algoritmo guloso baseado no ganho incremental da função objetivo e uma metaheurística GRASP (Feo; Resende, 1995), que combina construção aleatorizada com procedimentos de busca local, permitindo explorar diferentes regiões do espaço de soluções.

Os resultados experimentais demonstram que ambos os algoritmos produzem soluções equivalentes em todos os cenários avaliados. A análise visual do posicionamento das RSUs confirma que ambos os algoritmos convergem para padrões espacialmente equivalentes, preenchendo progressivamente as regiões de maior densidade viária à medida que o número de RSUs aumenta.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve a representação das malhas rodoviárias adotada. A Seção 4 formaliza o problema e a métrica Delta, incluindo o cálculo da área. A Seção 5 apresenta os algoritmos desenvolvidos. A Seção 6 discute os resultados computacionais obtidos. Por fim, a Seção 7 apresenta as considerações finais.

## 2 TRABALHOS RELACIONADOS

O problema de posicionamento de RSUs em VANETs tem sido abordado sob diferentes perspectivas ao longo do tempo. Abordagens iniciais consideram métricas baseadas em garantias de conectividade, como a *alpha-coverage* (Zheng *et al.*, 2009). Posteriormente, abordagens baseadas em cobertura passaram a ser adotadas, mantendo uma representação binária da conectividade (Trullols *et al.*, 2010).

A partir da métrica Delta (Silva; Meira Jr., 2015a), diversos trabalhos exploraram o problema de posicionamento de RSUs por meio de abordagens heurísticas e metaheurísticas. Estratégias gulosas, como a heurística Delta-r (Sarubbi; Silva, 2016), e variantes baseadas em GRASP como DL\_GRASP, Delta-g\_GRASP e Delta-r\_GRASP (Silva *et al.*, 2016) foram propostas para melhorar a qualidade das soluções, incorporando mecanismos de aleatoriedade e busca local. Sarubbi *et al.* (2017) também apresentaram uma extensão dessas abordagens, incorporando uma etapa de busca em vizinhança variável (VNS) (Mladenović; Hansen, 1997) com o objetivo de refinar as soluções obtidas. De forma geral, esses trabalhos mantiveram a formulação baseada na minimização da infraestrutura sob restrições de QoS.

Silva *et al.* (2019) consideraram a análise global da conectividade por meio da curva completa, propondo uma estratégia baseada em *scores* para priorizar regiões de cobertura, permitindo a customização de diferentes padrões de conectividade. Nessa linha, essa estratégia passa a ser utilizada não apenas como instrumento de avaliação, mas também como base para a construção de soluções. Posteriormente, Silva *et al.* (2023) apresentaram uma estratégia baseada na geração aleatória de soluções e redução incremental do espaço de busca, aplicada a problemas não polinomiais, incluindo o posicionamento de RSUs. Além disso, esse trabalho propôs o modelo matemático adotado neste artigo, já considerando a maximização da área sob a curva da rede Delta como indicador de qualidade de serviço.

Além da métrica Delta, outras perspectivas para o problema de posicionamento de RSUs também foram exploradas. Huo *et al.* (2024) propuseram uma estratégia multiobjetivo considerando simultaneamente cobertura confiável e custo de implantação como critérios de otimização, mantendo, contudo, uma representação binária da conectividade.

Mais diretamente relacionado ao presente trabalho, Marques (2020) abordou o problema de maximização da área do gráfico da rede Delta considerando uma quantidade fixa de RSUs, utilizando um algoritmo genético com seleção por torneio e elitismo. Naquele trabalho, o algoritmo genético foi comparado com três algoritmos gulosos que selecionam as  $\eta$  melhores células com base em métricas estáticas: quantidade de veículos (Guloso I), tempo total de permanência (Guloso II) e tempo médio de permanência (Guloso III). Os

experimentos foram conduzidos sobre a instância de Colônia com 1.000 e 5.000 veículos, com  $\eta$  variando de 20 a 200 RSUs. O algoritmo genético superou os algoritmos gulosos em todos os cenários, com ganhos de até 33% em relação ao melhor guloso.

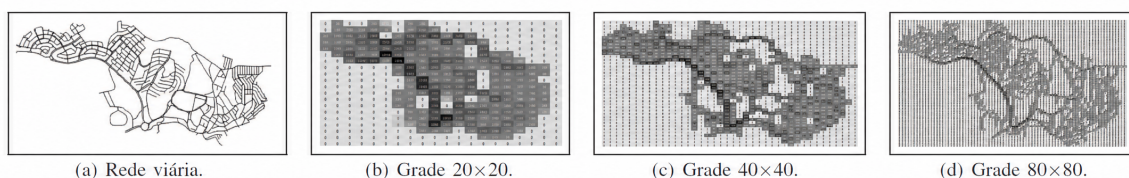
Diferente da maioria dos trabalhos citados, que minimizam a infraestrutura necessária sob restrições de QoS ou adotam representações binárias de conectividade, este trabalho investiga a resolução do problema de maximização da área da curva Delta. Embora Silva *et al.* (2023) também adote a maximização da área sob a curva Delta como critério de qualidade e proponha o modelo matemático utilizado como referência neste trabalho, a abordagem ali desenvolvida baseia-se na geração aleatória de soluções com redução incremental do espaço de busca, sem uma fase construtiva gulosa nem uma etapa de busca local estruturada. O presente trabalho difere ao propor um algoritmo guloso baseado em ganho incremental e um algoritmo baseado na metaheurística GRASP, que combina construção aleatorizada com busca local, tendo como referência o modelo exato resolvido via CPLEX. Em relação ao trabalho de Marques (2020), avança-se ao substituir os algoritmos gulosos estáticos utilizados como baseline por uma abordagem gulosa que avalia explicitamente o impacto de cada decisão sobre a função objetivo e ao propor um algoritmo baseado na metaheurística GRASP, em contraste com o algoritmo genético adotado naquele trabalho.

### 3 REPRESENTAÇÃO DE MALHAS RODOVIÁRIAS

As malhas rodoviárias reais apresentam topologias complexas e irregulares, o que dificulta sua modelagem direta em problemas de otimização. Para lidar com essa complexidade, neste trabalho adota-se uma abordagem baseada na discretização espacial da região de interesse, conforme proposto por (Silva; Meira Jr., 2015a).

A estratégia consiste em particionar a área urbana em uma *grade bidimensional* composta por células de mesmo tamanho, formando uma malha regular. Cada célula da grade representa uma região geográfica elementar, permitindo abstrair a malha viária original e representar o problema de forma estruturada e tratável computacionalmente.

Figura 2 – Representação da malha rodoviária da cidade de Ouro Branco (MG) por meio de discretização em grade com diferentes resoluções.



Fonte: Adaptado de Silva e Meira Jr. (2015a).

A principal vantagem dessa modelagem é permitir um equilíbrio entre precisão espacial e esforço computacional, uma vez que a resolução da grade pode ser ajustada conforme necessário. Além disso, a discretização transforma o problema contínuo de posicionamento de RSUs em um problema combinatório sobre um conjunto finito de células.

Neste trabalho, adota-se a premissa de que cada RSU cobre exatamente a célula em que está instalada. Assim, um veículo é considerado conectado à infraestrutura somente durante o intervalo de tempo em que ocupa a mesma célula que uma RSU. Essa premissa

implica que a resolução da grade influencia diretamente a granularidade da cobertura modelada.

## 4 DEFINIÇÃO DO PROBLEMA

Esta seção formaliza o problema de posicionamento de RSUs abordado neste trabalho. A Seção 4.1 apresenta a métrica Delta, utilizada para avaliar a qualidade de serviço da rede. A Seção 4.2 descreve o cálculo da área sob o gráfico da rede Delta. A Seção 4.3 apresenta o modelo matemático utilizado como referência.

### 4.1 Definição da métrica Delta $\Delta_{\rho_1}^{\rho_2}$

A métrica Delta  $\Delta_{\rho_1}^{\rho_2}$ , proposta por (Silva; Meira Jr., 2015a), define requisitos de qualidade de serviço em redes veiculares a partir de dois parâmetros:  $\rho_1$ , que representa a fração do tempo de viagem em que um veículo permanece conectado à infraestrutura, e  $\rho_2$ , que indica a fração de veículos que atendem a esse requisito.

**Definição 1.** *Seja  $\mathbf{M}$  a representação da malha rodoviária, e  $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$  um conjunto de veículos viajando por  $\mathbf{M}$ . Seja  $\mathbf{T} = \{\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_n\}$  o conjunto que representa a trajetória de cada veículo  $v \in \mathbf{V}$ . Logo, cada  $v_k \in \mathbf{V}$  possui uma trajetória específica  $\mathbf{U}_k \in \mathbf{T}$ . Cada  $\mathbf{U} \in \mathbf{T}$  representa uma trajetória de células  $\mathbf{U}_k = \{u_1^k, u_2^k, \dots, u_m^k\}$  percorrida por um veículo  $v_k$  durante a viagem. Seja  $\mathbf{C} \subset \mathbf{V}$  o conjunto de veículos  $v_k$  que estejam conectados durante um percentual da viagem  $\geq \rho_1 \forall u \in \mathbf{U}_k$ . Essa rede veicular é considerada  $\Delta_{\rho_1}^{\rho_2}$  sempre que  $\frac{|\mathbf{C}|}{|\mathbf{V}|} \geq \rho_2$ .*

### 4.2 Cálculo da área do gráfico da rede Delta

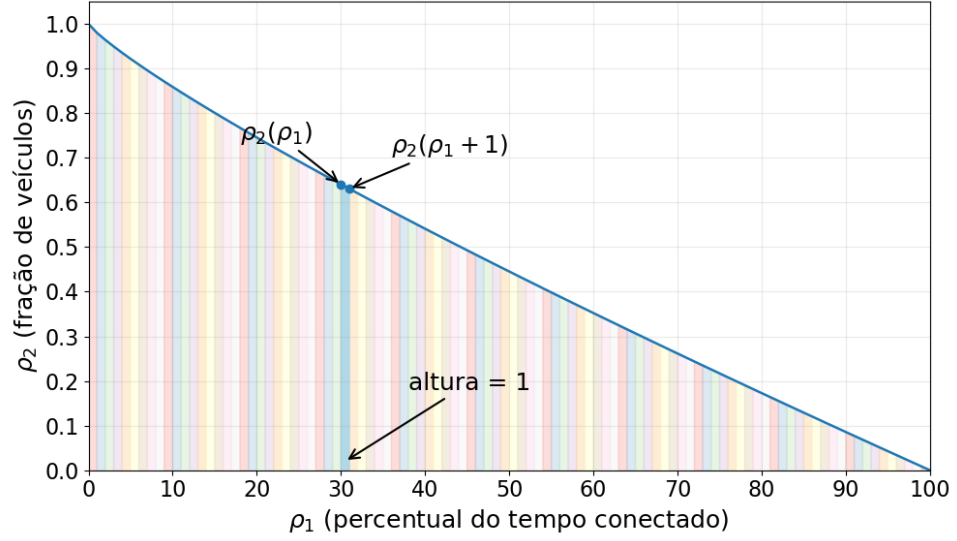
Para avaliar a qualidade global da rede, considera-se a área sob o gráfico da rede Delta. Essa área representa, de forma agregada, os níveis de serviço obtidos para diferentes combinações dos parâmetros  $\rho_1$  e  $\rho_2$ .

O gráfico da rede Delta, ilustrado na Figura 1, é construído a partir da relação entre  $\rho_1$ , que representa o percentual do tempo de viagem em que os veículos permanecem conectados, e  $\rho_2$ , que indica a fração de veículos que atendem a esse nível de conectividade. Neste trabalho, o eixo horizontal é discretizado no intervalo de 0 a 100, com passo unitário, enquanto o eixo vertical é normalizado no intervalo  $[0, 1]$ .

Para cada valor de  $\rho_1$ , calcula-se o valor correspondente de  $\rho_2$ , resultando em uma curva que descreve o desempenho da rede. Como essa curva é obtida de forma discreta, a área sob o gráfico é aproximada numericamente por meio do método dos trapézios, em que cada trapézio é formado a partir de dois pontos consecutivos correspondentes a  $\rho_1$  e  $\rho_1 + 1$ , com altura unitária e bases dadas pelos valores de  $\rho_2$  nesses dois pontos.

A área total é obtida pela soma das áreas de todos os trapézios ao longo do intervalo considerado. Dessa forma, quanto maior a área sob o gráfico da rede Delta, melhor é a qualidade global da rede, pois isso indica maior cobertura dos veículos ao longo de suas trajetórias.

Figura 3 – Aproximação da área do gráfico da rede Delta por meio do método dos trapézios.



Fonte: elaborado pelo autor (2026).

Como o eixo  $\rho_2$  é normalizado e o eixo  $\rho_1$  varia de 0 a 100, a área máxima possível do gráfico da rede Delta é igual a 100.

### 4.3 Modelo Matemático

Neste trabalho, utiliza-se o modelo matemático apresentado por (Silva *et al.*, 2023), que tem como objetivo maximizar a área do gráfico da rede Delta em uma malha rodoviária, considerando uma quantidade fixa  $\eta$  de RSUs.

Considerando os seguintes conjuntos:

U: conjunto de células urbanas onde é possível instalar uma RSU onde  $U = \{u_1, u_2, \dots, u_n\}$

P: conjunto de  $\rho_1$  onde  $P = \{0\%, 1\%, \dots, 100\%\}$

V: conjunto de veículos  $\{v_1, v_2, \dots, v_n\}$  viajando pela malha rodoviária.

E sendo as variáveis:

$\vartheta_{\rho_1}$ : fração de veículos que permanecem conectados à infraestrutura por pelo menos  $\rho_1\%$  do tempo de sua viagem

$$a_u = \begin{cases} 1, & \text{se a célula urbana } u \text{ recebe uma RSU} \\ 0, & \text{caso contrário} \end{cases}$$

$$v_k^{\rho_1} = \begin{cases} 1, & \text{se o veículo } k \text{ permanece } \rho_1\% \text{ da sua viagem conectado} \\ 0, & \text{caso contrário} \end{cases}$$

E os seguintes parâmetros:

$t_{uk}$ : tempo que o veículo  $k$  permanece na célula  $u$

$tv_k$ : tempo total de viagem do veículo  $k$

$\eta$ : número de RSUs a serem posicionadas

$m$ : passo  $\rho$  em que o gráfico da rede Delta é dividido

Dessa forma o modelo matemático é:

$$\max \sum_{\rho_1 \in P} \frac{(\vartheta_{\rho_1} + \vartheta_{\rho_1+m}) \times m}{2} \quad (1)$$

$$\sum_{u \in U} a_u \leq \eta \quad (2)$$

$$\sum_{u \in U} \frac{t_{uk}}{tv_k} \times a_u \geq \rho_1 v_k^{\rho_1} \quad \forall k \in V, \forall \rho_1 \in P \quad (3)$$

$$\frac{\sum_{k \in V} v_k^{\rho_1}}{|V|} = \vartheta_{\rho_1} \quad \forall \rho_1 \in P \quad (4)$$

$$a_u \in \{0, 1\} \quad \forall u \in U \quad (5)$$

$$v_k^{\rho_1} \in \{0, 1\} \quad \forall k \in V, \forall \rho_1 \in P \quad (6)$$

Com  $\vartheta_{\rho_1}$  representando a porcentagem de veículos sendo atendidos por  $\rho_1\%$  de sua viagem. A função objetivo (1) calcula o somatório das áreas do trapézio formado para cada  $\rho_1$  sendo  $\vartheta_{\rho_1}$  a base maior e  $\vartheta_{\rho_1+m}$  a base menor do trapézio e a altura desse trapézio sendo  $m$ . A restrição (2) garante que não haverá mais que  $\eta$  células com RSUs alocadas. A restrição (3) define quais veículos estão conectados às RSUs por pelo menos  $\rho_1\%$  do tempo de sua viagem. A restrição (4) calcula a porcentagem de veículos que estão conectados por pelo menos  $\rho_1\%$  do tempo da sua viagem. As restrições (5) e (6) são restrições de integralidade.

## 5 ALGORITMOS DESENVOLVIDOS

Esta seção apresenta os algoritmos desenvolvidos para a resolução do problema de posicionamento de RSUs. A Seção 5.1 descreve o algoritmo guloso baseado em ganho incremental, e a Seção 5.2 apresenta o algoritmo baseado na metaheurística GRASP.

### 5.1 Algoritmo Guloso

Neste trabalho apresentamos um algoritmo guloso que constrói a solução de forma iterativa, selecionando, a cada passo, a célula que produz o maior ganho incremental na

área do gráfico da rede Delta. A estratégia adotada avalia diretamente o impacto de cada decisão sobre a função objetivo, considerando explicitamente a variação da área resultante da inserção de novas unidades de comunicação.

Inicialmente, é realizado um pré-processamento no qual se calcula o tempo total de viagem de cada veículo. Em seguida, para cada célula da malha, calcula-se o tempo total acumulado de permanência dos veículos, sendo as células ordenadas de forma decrescente segundo esse critério. Com o objetivo de reduzir o custo computacional, a busca é restrita a um subconjunto das  $K$  células com maior tempo total acumulado de permanência dos veículos, definido pelo parâmetro Top-K. Esse subconjunto é fixo e determinístico, calculado uma única vez no pré-processamento com base em uma métrica estática, sem considerar o ganho incremental na função objetivo.

A partir de uma solução inicialmente vazia, o algoritmo avalia, em cada iteração, a adição de cada célula candidata ainda não selecionada. Para isso, calcula-se o impacto da inserção de cada célula sobre o tempo de cobertura dos veículos e recalcula-se a área do gráfico Delta por meio do método dos trapézios. O ganho incremental é obtido pela diferença entre a nova área e a área da solução atual. A célula que apresenta o maior ganho incremental positivo é incorporada à solução, atualizando-se os tempos cobertos e o valor da função objetivo. O processo é repetido até que seja atingido o número máximo de RSUs, denotado por  $\eta$ .

---

**Algoritmo 1** Guloso com Top-K

---

```

1: procedure GULOSO_TOPK( $\eta, TopK$ )
2:    $candidatos \leftarrow$  TopK células mais promissoras
3:    $S \leftarrow \emptyset$ 
4:   while  $|S| < \eta$  do
5:     selecionar  $c \in candidatos \setminus S$  com maior ganho
6:     if não houver ganho positivo then
7:       interromper
8:      $S \leftarrow S \cup \{c\}$ 
9:   return  $S$ 

```

---

## 5.2 Algoritmo baseado na metaheurística GRASP

Com o intuito de explorar de forma mais abrangente o espaço de soluções e tentar melhorar a solução do algoritmo guloso, foi implementado um algoritmo baseado na metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) (Feo; Resende, 1995). O método combina uma fase construtiva com aleatoriedade controlada e uma fase de busca local, sendo executado por um número fixo de iterações. A cada iteração, uma nova solução é gerada e refinada, sendo mantida a melhor solução encontrada ao longo do processo.

A construção da solução inicia-se a partir de um conjunto vazio, no qual novas RSUs são adicionadas iterativamente até atingir o limite  $\eta$ . A cada iteração, para cada célula candidata  $c$  ainda não selecionada, é avaliado o ganho incremental de área  $g(c) = A(S \cup \{c\}) - A(S)$ , em que  $A(S)$  representa a área do gráfico da rede Delta associada à solução corrente  $S$ . O cálculo do ganho é realizado conforme descrito na Seção 5.1.

A partir dos valores de ganho obtidos, são identificados o maior e o menor incremento, denotados por  $g_{\max}$  e  $g_{\min}$ . Com base nesses valores, é construída a Lista Restrita de

Candidatos (RCL), composta pelas células que satisfazem  $g(c) \geq g_{\max} - \alpha(g_{\max} - g_{\min})$ , em que  $\alpha \in [0, 1]$  controla o nível de aleatoriedade do processo. A escolha do próximo elemento a ser inserido na solução é realizada de forma aleatória entre os elementos da RCL, permitindo equilibrar intensificação e diversificação no processo de busca.

Após a construção da solução inicial, é aplicada uma etapa de busca local com o objetivo de refinamento. A estratégia adotada consiste em um procedimento de realocação (*relocate*), no qual cada RSU da solução é temporariamente removida e todas as células vizinhas dentro de um raio predefinido  $r$  são avaliadas como possíveis novas posições para a RSU removida, sendo selecionado o movimento que proporciona o maior ganho na função objetivo, caracterizando uma estratégia do tipo *best improvement*. O processo é repetido até que uma passada completa sobre todas as RSUs não produza melhorias adicionais, caracterizando a convergência para um ótimo local.

Em cada rodada, para cada uma das  $\eta$  RSUs da solução, são avaliados até  $(2r+1)^2 - 1$  vizinhos candidatos, e para cada candidato percorre-se o conjunto completo de  $|V|$  veículos para recalcular a cobertura acumulada. O custo por rodada é, portanto,  $O(\eta \cdot (2r+1)^2 \cdot |V|)$ , crescendo linearmente com o número de veículos e quadraticamente com o raio de vizinhança  $r$ . O número de rodadas até a convergência varia conforme a instância, o que torna o custo total da busca local maior para instâncias de grande escala, conforme evidenciado pelos tempos de execução reportados na Seção 6.3.

O algoritmo é executado por um número máximo de iterações, sendo que, ao final, a melhor solução encontrada ao longo de todas as execuções é retornada. Os principais parâmetros do método são  $\alpha$ , que controla o nível de aleatoriedade,  $r$ , que define o raio da vizinhança na busca local, e  $MaxIter$ , que estabelece o número de iterações do GRASP.

Os Algoritmos 2 e 3 apresentam, de forma resumida, a estrutura geral do método.

---

**Algoritmo 2** GRASP

---

```

1: procedure GRASP( $\eta, \alpha, MaxIter$ )
2:    $melhor \leftarrow \emptyset$ 
3:   for  $i \leftarrow 1$  até  $MaxIter$  do
4:      $S \leftarrow \text{Fase\_Construcao}(\eta, \alpha)$ 
5:      $S \leftarrow \text{Busca\_Local}(S)$ 
6:     if  $S$  melhor que  $melhor$  then
7:        $melhor \leftarrow S$ 
8:   return  $melhor$ 

```

---



---

**Algoritmo 3** Busca Local

---

```

1: procedure BUSCA_LOCAL( $S$ )
2:   while existir melhoria na solução do
3:     explorar vizinhança de  $S$  por realocação de RSUs
4:     atualizar  $S$  com a melhor solução encontrada
5:   return  $S$ 

```

---

## 6 RESULTADOS COMPUTACIONAIS

Esta seção apresenta os resultados computacionais obtidos para o problema de posicionamento de RSUs. Os experimentos são conduzidos sobre o *Vehicular Mobility Trace*

da cidade de Colônia, Alemanha (Uppoor *et al.*, 2014), uma base de dados de mobilidade veicular urbana de larga escala amplamente utilizada na literatura de VANETs. A malha rodoviária é representada por uma grade de  $100 \times 100$  células. Os algoritmos heurísticos são executados em um computador equipado com processador Intel Core i5-8350U (1,70 GHz) e 28 GB de memória RAM, sob o sistema operacional Ubuntu 22.04.5 LTS. O modelo exato é resolvido com o solver CPLEX versão 12.7.1.0. A Seção 6.1 apresenta os resultados do modelo exato resolvido via CPLEX, utilizado como referência para instâncias de pequeno porte. A Seção 6.2 apresenta os resultados do algoritmo guloso. A Seção 6.3 apresenta os resultados da metaheurística GRASP. A Seção 6.4 apresenta os resultados obtidos sobre uma instância sintética com distribuição de tráfego mais uniforme. A Seção 6.5 compara os resultados obtidos pelos dois algoritmos, e a Seção 6.6 analisa visualmente o posicionamento das RSUs obtido por cada abordagem.

## 6.1 Resultados Exatos

Os resultados do modelo exato, obtidos para uma instância com 100 veículos e quantidades de RSUs variando de 1 a 10, são apresentados na Tabela 1. Observa-se que a área da rede Delta cresce de forma consistente com o aumento do número de RSUs, partindo de 1,36 para 1 RSU e atingindo 6,47 para 10 RSUs. Esse comportamento é ilustrado na Figura 4, que mostra as curvas Delta correspondentes. À medida que mais RSUs são adicionadas, as curvas se deslocam progressivamente para cima, indicando maior cobertura dos veículos ao longo de suas trajetórias.

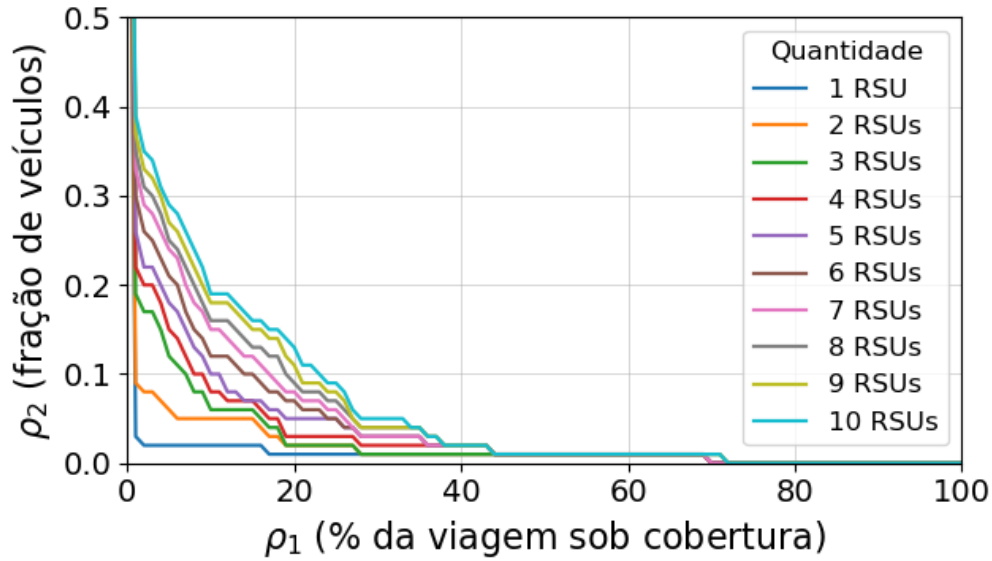
Os resultados mostram que o modelo exato encontra soluções ótimas para instâncias de pequeno porte, porém com tempo de execução que cresce exponencialmente com o número de RSUs. Para ilustrar essa limitação, no cenário com 20 RSUs, após aproximadamente 31,8 horas de execução, o gap relativo obtido foi de 12,23%. Vale ressaltar que, no contexto de planejamento estrutural de infraestrutura, no qual as decisões de posicionamento de RSUs tendem a ser tomadas com antecedência e permanecem válidas por longos períodos, tempos de execução dessa ordem podem ser aceitáveis. Ainda assim, para instâncias de maior escala, a abordagem exata torna-se inviável, motivando o uso de algoritmos heurísticos.

Tabela 1 – Resultados do modelo exato para diferentes quantidades de RSUs (100 veículos)

| RSUs  | 1     | 2      | 3      | 4       | 5       | 6       | 7    | 8    | 9    | 10    |
|-------|-------|--------|--------|---------|---------|---------|------|------|------|-------|
| Área  | 1,36  | 2,08   | 2,76   | 3,36    | 3,93    | 4,49    | 5,03 | 5,56 | 6,02 | 6,47  |
| Tempo | 0,73s | 15,54s | 65,58s | 20,9min | 19,4min | 49,7min | 2,9h | 4,9h | 8,0h | 19,0h |

Fonte: elaborado pelo autor (2026).

Figura 4 – Curvas Delta para diferentes quantidades de RSUs (100 veículos).



Fonte: elaborado pelo autor (2026).

## 6.2 Algoritmo Guloso

Para validar a qualidade da solução produzida pelo algoritmo guloso, são realizadas comparações com o modelo exato em instâncias de menor escala, considerando de 1 a 10 RSUs e 100 veículos. Os resultados são apresentados na Tabela 2. Observa-se que, para todos os casos avaliados, o algoritmo guloso atinge exatamente os mesmos valores de função objetivo obtidos pelo modelo exato, em tempo computacional inferior a 0,09 segundos em todos os cenários, indicando que, nessas instâncias, a heurística é capaz de encontrar soluções ótimas de forma eficiente.

Tabela 2 – Resultados do algoritmo guloso (Top-K = 500) para 100 veículos

| RSUs      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Área      | 1,3600 | 2,0800 | 2,7600 | 3,3600 | 3,9300 | 4,4900 | 5,0300 | 5,5600 | 6,0200 | 6,4700 |
| Tempo (s) | 0,0813 | 0,0507 | 0,0506 | 0,0517 | 0,0520 | 0,0533 | 0,0546 | 0,0551 | 0,0554 | 0,0560 |

Fonte: elaborado pelo autor (2026).

A partir desse resultado, a análise é estendida para instâncias de maior escala, considerando cenários com 1.000 e 5.000 veículos, nos quais a abordagem exata se torna computacionalmente inviável.

Tabela 3 – Resultados do algoritmo guloso (Top-K = 500) para 1.000 veículos

| RSUs      | 20     | 40      | 60      | 80      | 100     | 120     | 140     | 160     | 180     | 200     |
|-----------|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Área      | 8,2410 | 13,4525 | 17,8380 | 21,5840 | 24,9335 | 27,9920 | 30,8100 | 33,4025 | 35,8335 | 38,1725 |
| Tempo (s) | 1,18   | 1,45    | 1,76    | 2,06    | 2,55    | 2,61    | 2,89    | 3,14    | 3,37    | 3,60    |

Fonte: elaborado pelo autor (2026).

Tabela 4 – Resultados do algoritmo guloso (Top-K = 500) para 5.000 veículos

| RSUs      | 20     | 40      | 60      | 80      | 100     | 120     | 140     | 160     | 180     | 200     |
|-----------|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Área      | 7,8425 | 12,7956 | 16,8881 | 20,3776 | 23,5705 | 26,5447 | 29,3031 | 31,8870 | 34,2947 | 36,5599 |
| Tempo (s) | 3,91   | 5,86    | 7,71    | 9,69    | 11,51   | 13,21   | 14,89   | 16,29   | 17,84   | 18,68   |

Fonte: elaborado pelo autor (2026).

### 6.3 Algoritmo baseado na metaheurística GRASP

Nesta subseção apresentamos os resultados do algoritmo GRASP para as mesmas instâncias testadas pelo algoritmo guloso, considerando cenários com 1.000 e 5.000 veículos e quantidades de RSUs variando de 20 a 200. Assim como o algoritmo guloso, o GRASP também foi validado em instâncias de pequeno porte. Os resultados são apresentados na Tabela 5. Em todos os cenários de 1 a 10 RSUs com 100 veículos, a melhor solução encontrada pelo GRASP iguala o valor ótimo obtido pelo modelo exato, com tempo de execução inferior a 18 segundos em todos os casos.

Tabela 5 – Resultados do algoritmo baseado na metaheurística GRASP ( $\alpha = 0,2$ , iterações = 100) para 100 veículos

| RSUs      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Melhor    | 1,3600 | 2,0800 | 2,7600 | 3,3600 | 3,9300 | 4,4900 | 5,0300 | 5,5600 | 6,0200 | 6,4700 |
| Mediana   | 1,3600 | 2,0800 | 2,6800 | 3,3000 | 3,8800 | 4,4400 | 4,9650 | 5,4700 | 5,9300 | 6,4250 |
| Média     | 1,2928 | 2,0436 | 2,6863 | 3,3024 | 3,8725 | 4,4323 | 4,9685 | 5,4658 | 5,9595 | 6,4110 |
| Pior      | 1,2200 | 1,9600 | 2,6400 | 3,2000 | 3,8100 | 4,2800 | 4,8100 | 5,2600 | 5,9000 | 6,3000 |
| DP        | 0,0699 | 0,0503 | 0,0470 | 0,0550 | 0,0432 | 0,0461 | 0,0557 | 0,0707 | 0,0420 | 0,0451 |
| Tempo (s) | 1,73   | 3,58   | 5,41   | 7,31   | 8,82   | 10,53  | 12,06  | 14,01  | 15,39  | 17,54  |

Fonte: elaborado pelo autor (2026).

Os experimentos são conduzidos com os parâmetros  $\alpha = 0,2$ , número de iterações igual a 100 e raio de vizinhança  $r = 2$ . Experimentos adicionais com raios maiores também são considerados, porém o GRASP não supera o algoritmo guloso em nenhum dos cenários avaliados. Para cada cenário, o algoritmo é executado múltiplas vezes, sendo registradas métricas como melhor solução encontrada, mediana, média, pior solução e desvio padrão.

Tabela 6 – Resultados do algoritmo baseado na metaheurística GRASP ( $\alpha = 0,2$ , iterações = 100) para 1.000 veículos

| RSUs      | 20     | 40      | 60      | 80      | 100     | 120     | 140     | 160     | 180     | 200     |
|-----------|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Melhor    | 8,2410 | 13,4525 | 17,8340 | 21,5840 | 24,9335 | 27,9920 | 30,8080 | 33,3985 | 35,8225 | 38,1590 |
| Mediana   | 8,1815 | 13,3765 | 17,7973 | 21,5335 | 24,8745 | 27,9280 | 30,7240 | 33,3275 | 35,7575 | 38,0888 |
| Média     | 8,1758 | 13,3727 | 17,7946 | 21,5316 | 24,8690 | 27,9278 | 30,7245 | 33,3249 | 35,7542 | 38,0828 |
| Pior      | 8,0380 | 13,2285 | 17,7065 | 21,4600 | 24,7690 | 27,7880 | 30,5925 | 33,2410 | 35,6765 | 37,9530 |
| DP        | 0,0419 | 0,0479  | 0,0265  | 0,0252  | 0,0390  | 0,0367  | 0,0447  | 0,0336  | 0,0346  | 0,0451  |
| Tempo (s) | 547,30 | 1091,80 | 1645,56 | 2199,23 | 2757,72 | 3292,55 | 3841,02 | 4394,56 | 4974,44 | 5580,01 |

Fonte: elaborado pelo autor (2026).

Tabela 7 – Resultados do algoritmo baseado na metaheurística GRASP ( $\alpha = 0,2$ , iterações = 100) para 5.000 veículos

| RSUs             | 20      | 40      | 60      | 80       | 100      | 120      | 140      | 160      | 180      | 200      |
|------------------|---------|---------|---------|----------|----------|----------|----------|----------|----------|----------|
| <b>Melhor</b>    | 7,8425  | 12,7956 | 16,8881 | 20,3776  | 23,5621  | 26,5436  | 29,2937  | 31,8837  | 34,2837  | 36,5400  |
| <b>Mediana</b>   | 7,7775  | 12,7290 | 16,8382 | 20,3312  | 23,4973  | 26,4552  | 29,2421  | 31,8129  | 34,2080  | 36,4775  |
| <b>Média</b>     | 7,7770  | 12,7296 | 16,8360 | 20,3294  | 23,4887  | 26,4510  | 29,2385  | 31,8104  | 34,2012  | 36,4754  |
| <b>Pior</b>      | 7,6727  | 12,6548 | 16,7390 | 20,2532  | 23,3634  | 26,3597  | 29,1128  | 31,6678  | 34,1007  | 36,3745  |
| <b>DP</b>        | 0,0375  | 0,0329  | 0,0291  | 0,0293   | 0,0412   | 0,0432   | 0,0337   | 0,0428   | 0,0454   | 0,0356   |
| <b>Tempo (s)</b> | 2875,73 | 5848,67 | 8881,07 | 11771,11 | 14747,57 | 18706,44 | 22218,52 | 25023,43 | 28374,73 | 31751,56 |

Fonte: elaborado pelo autor (2026).

## 6.4 Experimento com instância sintética

Com o objetivo de investigar o comportamento dos algoritmos em cenários com distribuição de tráfego mais uniforme, realiza-se um experimento exploratório sobre uma instância sintética gerada artificialmente sobre a mesma grade de  $100 \times 100$  células, com trajetórias geradas por caminhadas aleatórias multidirecionais, resultando em uma distribuição espacial significativamente mais homogênea do que a observada na instância de Colônia.

As Tabelas 8 e 9 apresentam os resultados para 1.000 veículos, e as Tabelas 10 e 11 para 5.000 veículos.

Tabela 8 – Resultados do algoritmo guloso (Top-K = 500) para a instância sintética com 1.000 veículos

| RSUs             | 20     | 40     | 60     | 80     | 100    | 120    | 140    | 160    | 180    | 200    |
|------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| <b>Área</b>      | 1,5540 | 2,4050 | 3,1690 | 3,8930 | 4,5830 | 5,2380 | 5,8720 | 6,4930 | 7,0970 | 7,6830 |
| <b>Tempo (s)</b> | 0,63   | 0,98   | 1,22   | 1,67   | 1,99   | 2,28   | 2,55   | 2,65   | 3,10   | 3,12   |

Fonte: elaborado pelo autor (2026).

Tabela 9 – Resultados do algoritmo baseado na metaheurística GRASP ( $\alpha = 0,2$ , iterações = 100) para a instância sintética com 1.000 veículos

| RSUs             | 20     | 40      | 60      | 80      | 100     | 120     | 140     | 160     | 180     | 200     |
|------------------|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| <b>Melhor</b>    | 1,5610 | 2,4300  | 3,2390  | 3,9820  | 4,7100  | 5,4230  | 6,1080  | 6,7810  | 7,4300  | 8,0750  |
| <b>Mediana</b>   | 1,5395 | 2,4120  | 3,1970  | 3,9525  | 4,6790  | 5,3900  | 6,0790  | 6,7420  | 7,3965  | 8,0370  |
| <b>Média</b>     | 1,5384 | 2,4093  | 3,1974  | 3,9496  | 4,6774  | 5,3867  | 6,0738  | 6,7401  | 7,3913  | 8,0323  |
| <b>Pior</b>      | 1,5090 | 2,3710  | 3,1410  | 3,9050  | 4,6340  | 5,3250  | 6,0160  | 6,6850  | 7,3420  | 7,9510  |
| <b>DP</b>        | 0,0115 | 0,0135  | 0,0203  | 0,0179  | 0,0180  | 0,0225  | 0,0200  | 0,0215  | 0,0219  | 0,0240  |
| <b>Tempo (s)</b> | 569,56 | 1140,34 | 1714,73 | 2308,93 | 2860,74 | 3461,13 | 4018,10 | 4663,52 | 5326,35 | 5894,46 |

Fonte: elaborado pelo autor (2026).

Tabela 10 – Resultados do algoritmo guloso (Top-K = 500) para a instância sintética com 5.000 veículos

| RSUs             | 20     | 40     | 60     | 80     | 100    | 120    | 140    | 160    | 180    | 200    |
|------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| <b>Área</b>      | 0,9646 | 1,3972 | 1,8168 | 2,2294 | 2,6358 | 3,0304 | 3,4164 | 3,7950 | 4,1678 | 4,5338 |
| <b>Tempo (s)</b> | 3,34   | 5,03   | 7,46   | 9,36   | 11,27  | 13,08  | 14,48  | 16,16  | 17,52  | 18,50  |

Fonte: elaborado pelo autor (2026).

Tabela 11 – Resultados do algoritmo baseado na metaheurística GRASP ( $\alpha = 0,2$ , iterações = 100) para a instância sintética com 5.000 veículos

| RSUs             | 20      | 40      | 60      | 80       | 100      | 120      | 140      | 160      | 180      | 200      |
|------------------|---------|---------|---------|----------|----------|----------|----------|----------|----------|----------|
| <b>Melhor</b>    | 0,9638  | 1,4040  | 1,8346  | 2,2504   | 2,6568   | 3,0600   | 3,4604   | 3,8508   | 4,2340   | 4,6144   |
| <b>Mediana</b>   | 0,9500  | 1,3927  | 1,8176  | 2,2329   | 2,6427   | 3,0369   | 3,4289   | 3,8196   | 4,2037   | 4,5778   |
| <b>Média</b>     | 0,9498  | 1,3918  | 1,8179  | 2,2324   | 2,6388   | 3,0358   | 3,4289   | 3,8168   | 4,2024   | 4,5776   |
| <b>Pior</b>      | 0,9372  | 1,3680  | 1,8000  | 2,2116   | 2,6070   | 3,0044   | 3,3856   | 3,7796   | 4,1652   | 4,5482   |
| <b>DP</b>        | 0,0052  | 0,0071  | 0,0082  | 0,0076   | 0,0111   | 0,0129   | 0,0176   | 0,0169   | 0,0166   | 0,0138   |
| <b>Tempo (s)</b> | 3026,46 | 6083,50 | 9167,75 | 12295,37 | 15129,00 | 18413,03 | 21951,02 | 26212,83 | 29762,25 | 32878,06 |

Fonte: elaborado pelo autor (2026).

## 6.5 Comparação entre algoritmos

Os resultados obtidos pelo algoritmo guloso e pelo GRASP são comparados nas Tabelas 3 a 7. Para o GRASP, são reportadas a melhor solução encontrada ao longo das 100 iterações, a mediana, a média, a pior solução e o desvio padrão. Em todos os cenários avaliados, a melhor solução encontrada pelo GRASP é equivalente à solução produzida pelo algoritmo guloso.

Acredita-se que esse comportamento pode ser explicado pelas características da instância utilizada. Como evidenciado pelo mapa de calor presente no fundo da Figura 8, o tráfego da instância de Colônia é extremamente concentrado em uma região central da malha, com intensidade consideravelmente superior às demais regiões. Nesse cenário, o espaço de soluções apresenta uma estrutura bem definida, na qual as células mais vantajosas são claramente identificáveis, o que favorece abordagens determinísticas como o algoritmo guloso. O GRASP, por sua natureza estocástica, foi projetado para explorar espaços de soluções com múltiplas regiões promissoras de qualidade similar. Em instâncias com tráfego mais distribuído espacialmente, espera-se que a diversificação proporcionada pelo GRASP produza soluções superiores às obtidas pelo algoritmo guloso. Na instância avaliada, entretanto, o tráfego altamente concentrado faz com que poucas células se destaquem com ganhos significativamente superiores às demais, levando o processo construtivo aleatorizado a convergir para as mesmas regiões identificadas deterministicamente pelo guloso.

Os resultados obtidos na instância sintética, apresentados na Seção 6.4, corroboram essa hipótese. Nesse cenário, com distribuição de tráfego significativamente mais uniforme, o GRASP supera o algoritmo guloso em todos os valores de  $\eta$  avaliados para 1.000 veículos, com ganhos crescentes à medida que o número de RSUs aumenta, variando de +0,45% para  $\eta = 20$  até +5,10% para  $\eta = 200$ . Para 5.000 veículos, os resultados confirmam essa tendência, com o GRASP superando o guloso em 9 dos 10 valores de  $\eta$  avaliados, com ganhos de até +1,78% para  $\eta = 200$ . A Tabela 12 sintetiza a comparação entre guloso e GRASP na instância sintética para ambos os volumes de veículos.

Tabela 12 – Comparação entre guloso e GRASP (melhor solução) na instância sintética

| RSUs | 1.000 veículos |        |          | 5.000 veículos |        |          |
|------|----------------|--------|----------|----------------|--------|----------|
|      | Guloso         | GRASP  | Dif. (%) | Guloso         | GRASP  | Dif. (%) |
| 20   | 1,5540         | 1,5610 | +0,45    | 0,9646         | 0,9638 | −0,08    |
| 40   | 2,4050         | 2,4300 | +1,04    | 1,3972         | 1,4040 | +0,49    |
| 60   | 3,1690         | 3,2390 | +2,21    | 1,8168         | 1,8346 | +0,98    |
| 80   | 3,8930         | 3,9820 | +2,29    | 2,2294         | 2,2504 | +0,94    |
| 100  | 4,5830         | 4,7100 | +2,77    | 2,6358         | 2,6568 | +0,80    |
| 120  | 5,2380         | 5,4230 | +3,53    | 3,0304         | 3,0600 | +0,98    |
| 140  | 5,8720         | 6,1080 | +4,02    | 3,4164         | 3,4604 | +1,29    |
| 160  | 6,4930         | 6,7810 | +4,43    | 3,7950         | 3,8508 | +1,47    |
| 180  | 7,0970         | 7,4300 | +4,69    | 4,1678         | 4,2340 | +1,59    |
| 200  | 7,6830         | 8,0750 | +5,10    | 4,5338         | 4,6144 | +1,78    |

Fonte: elaborado pelo autor (2026).

Os resultados obtidos neste trabalho também são comparados com os reportados por Marques (2020), que abordou o mesmo problema de maximização da área da rede Delta sobre a instância de Colônia utilizando um algoritmo genético. As Tabelas 13 e 14 apresentam a comparação entre os resultados obtidos pelos algoritmos deste trabalho e a maior área reportada pelo algoritmo genético daquela autora, para as instâncias de 1.000 e 5.000 veículos, respectivamente. Como demonstrado na Seção 6.5, o algoritmo guloso e o GRASP produziram resultados equivalentes em todos os cenários avaliados, sendo reportado o valor do guloso como representativo de ambos. Em todos os cenários avaliados, os algoritmos deste trabalho produzem áreas superiores, com a vantagem crescendo consistentemente com o aumento do número de RSUs.

Tabela 13 – Comparação entre os resultados deste trabalho e os de (Marques, 2020) para 1.000 veículos

| RSUs | Guloso  | Marques (2020) | Diferença (%) |
|------|---------|----------------|---------------|
| 20   | 8,2410  | 7,0700         | +16,6         |
| 40   | 13,4525 | 12,3270        | +9,1          |
| 60   | 17,8380 | 16,1770        | +10,3         |
| 80   | 21,5840 | 19,3870        | +11,3         |
| 100  | 24,9335 | 22,3210        | +11,7         |
| 120  | 27,9920 | 24,9230        | +12,3         |
| 140  | 30,8100 | 27,3130        | +12,8         |
| 160  | 33,4025 | 29,5620        | +13,0         |
| 180  | 35,8335 | 31,6540        | +13,2         |
| 200  | 38,1725 | 33,6220        | +13,5         |

Fonte: elaborado pelo autor (2026) com base em (Marques, 2020).

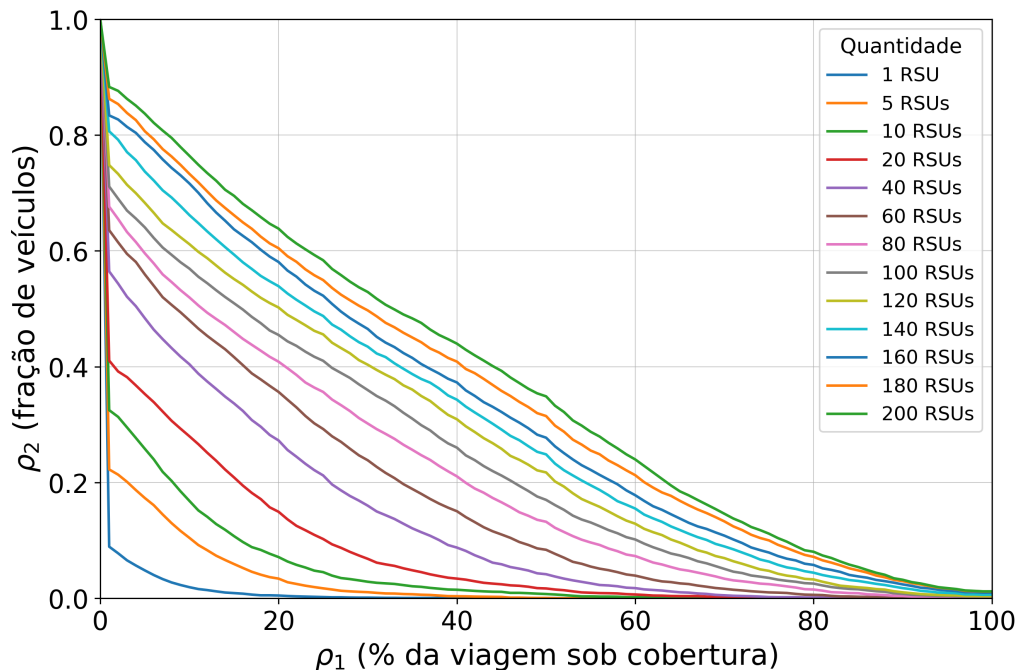
Tabela 14 – Comparação entre os resultados deste trabalho e os de (Marques, 2020) para 5.000 veículos

| RSUs | Guloso  | Marques (2020) | Diferença (%) |
|------|---------|----------------|---------------|
| 20   | 7,8425  | 7,1780         | +9,3          |
| 40   | 12,7956 | 11,5110        | +11,2         |
| 60   | 16,8881 | 15,0670        | +12,1         |
| 80   | 20,3776 | 18,0440        | +12,9         |
| 100  | 23,5705 | 20,7630        | +13,5         |
| 120  | 26,5447 | 23,2820        | +14,0         |
| 140  | 29,3031 | 25,6160        | +14,4         |
| 160  | 31,8870 | 27,7960        | +14,7         |
| 180  | 34,2947 | 29,8090        | +15,1         |
| 200  | 36,5599 | 31,6740        | +15,4         |

Fonte: elaborado pelo autor (2026) com base em (Marques, 2020).

A Figura 5 ilustra as curvas Delta obtidas pela melhor solução do GRASP para os diferentes cenários avaliados com 5.000 veículos. Observa-se que, à medida que o número de RSUs aumenta, as curvas se deslocam progressivamente para cima, indicando maior cobertura dos veículos. O comportamento é consistente com o esperado: cada RSU adicional contribui para aumentar o tempo de cobertura dos veículos, elevando tanto  $\rho_1$  quanto  $\rho_2$  para níveis maiores.

Figura 5 – Curvas Delta para diferentes quantidades de RSUs obtidas pelo algoritmo GRASP (5.000 veículos). À medida que o número de RSUs aumenta, as curvas se deslocam progressivamente para cima, indicando maior cobertura dos veículos ao longo de suas trajetórias.

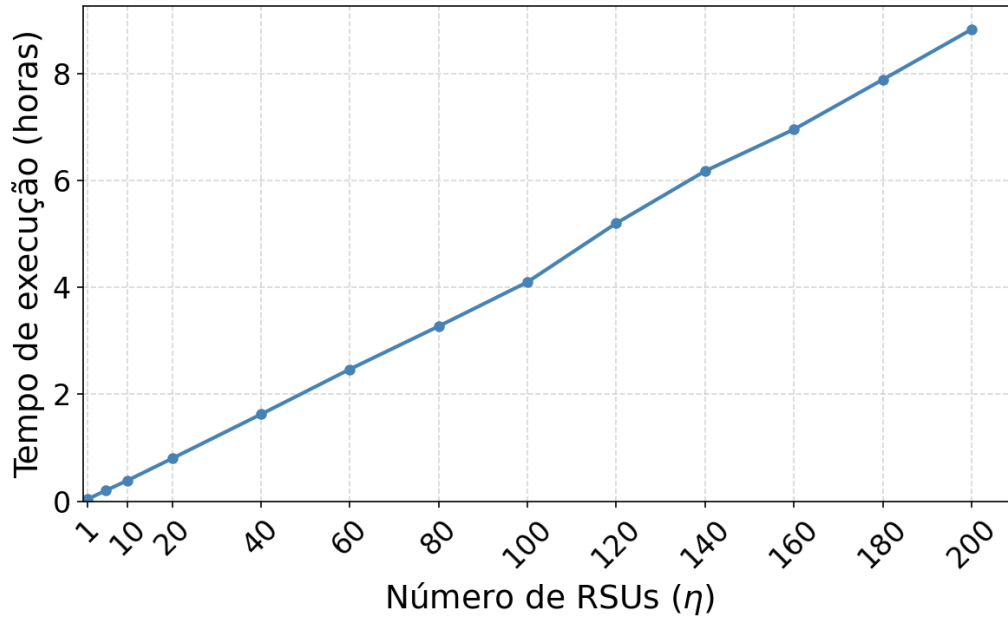


Fonte: elaborado pelo autor (2026).

A Figura 6 apresenta o tempo de execução do algoritmo GRASP em função do número de RSUs para a instância de Colônia com 5.000 veículos. Observa-se um crescimento

aproximadamente linear do tempo com o aumento de  $\eta$ , atingindo cerca de 8,8 horas para  $\eta = 200$ , o que reforça a análise de custo computacional discutida na Seção 5.2.

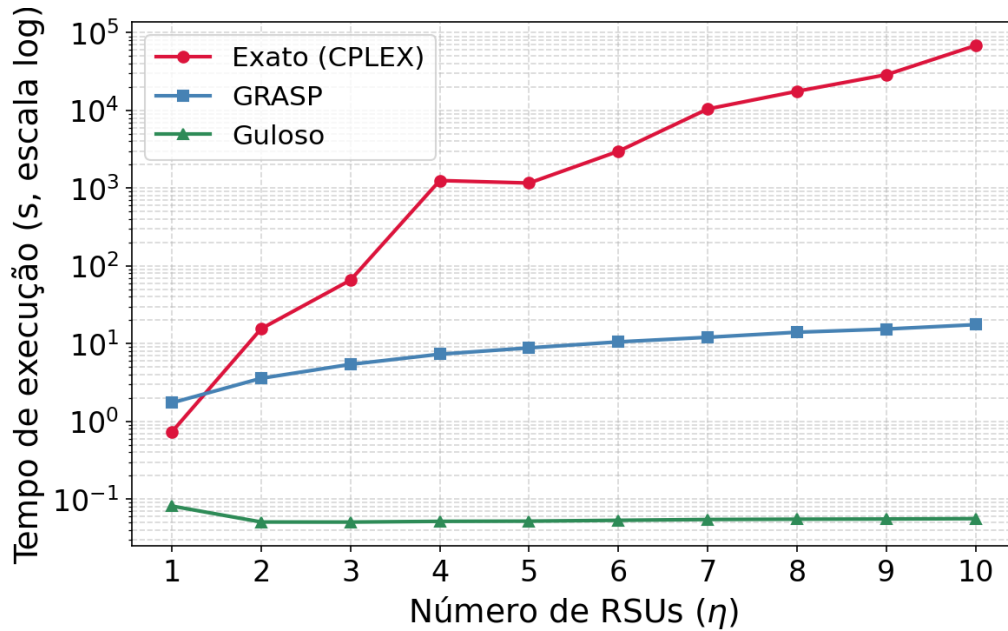
Figura 6 – Tempo de execução do algoritmo GRASP em função do número de RSUs para a instância de Colônia com 5.000 veículos.



Fonte: elaborado pelo autor (2026).

A Figura 7 ilustra a diferença de tempo de execução entre as três abordagens para a instância com 100 veículos. A escala logarítmica evidencia o crescimento exponencial do tempo do modelo exato em contraste com os tempos reduzidos do algoritmo guloso e do GRASP, que permanecem na faixa de milissegundos e dezessete segundos, respectivamente.

Figura 7 – Comparação do tempo de execução entre o modelo exato (CPLEX), o algoritmo guloso e o GRASP para a instância com 100 veículos e quantidades de RSUs variando de 1 a 10. O eixo vertical está em escala logarítmica.



Fonte: elaborado pelo autor (2026).

## 6.6 Análise do posicionamento das RSUs

A análise visual do posicionamento das RSUs permite complementar os resultados numéricos apresentados nas seções anteriores. A Figura 8 apresenta o posicionamento das RSUs obtido pelo algoritmo guloso e pelo algoritmo baseado na metaheurística GRASP para os cenários de 1, 10, 20, 40, 80, 120, 140, 160 e 200 RSUs, sobrepostos ao mapa de calor das trajetórias dos veículos.

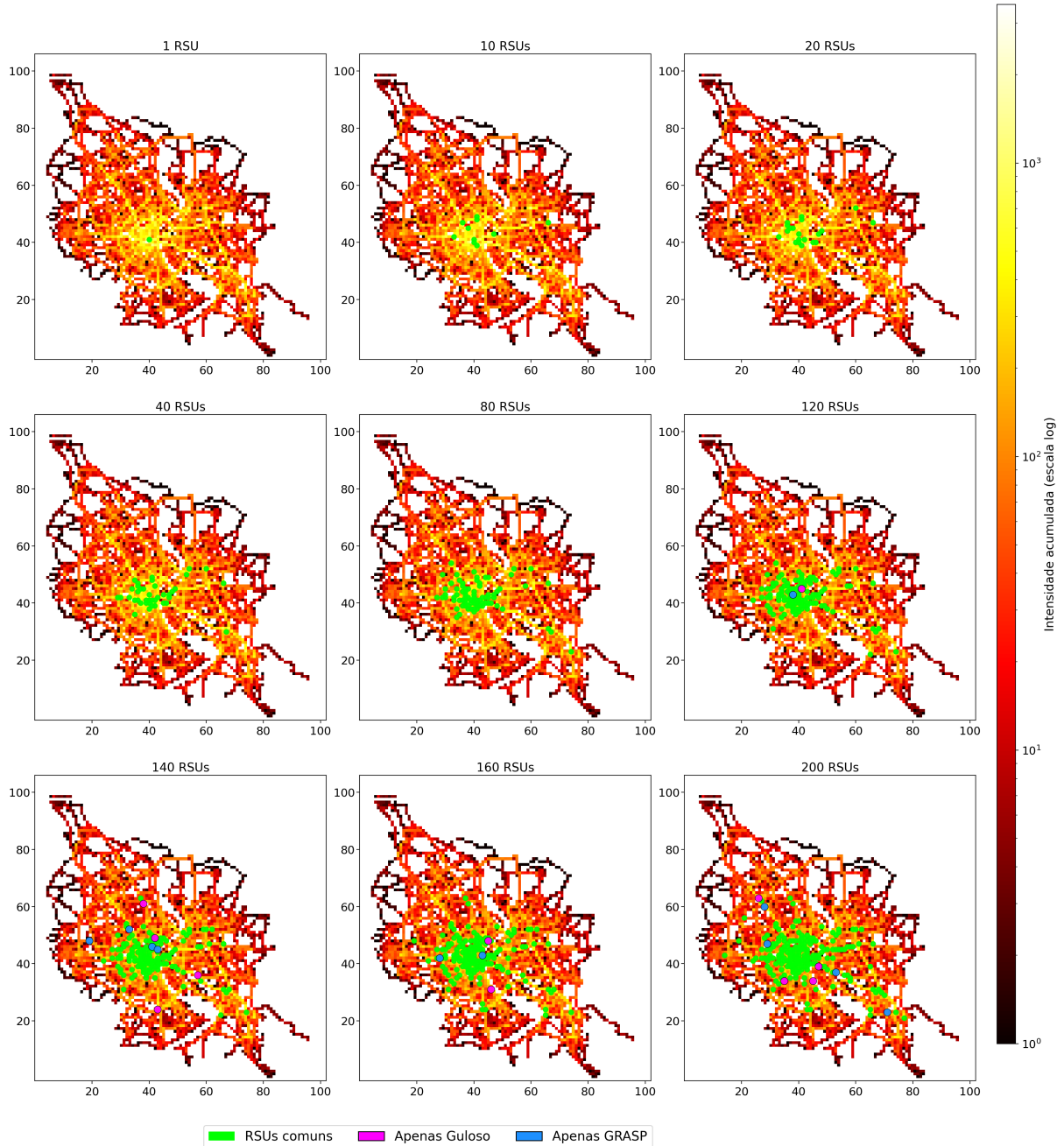
Em todos os cenários, ambos os algoritmos concentram as RSUs nas regiões de maior intensidade de tráfego, destacadas em amarelo no mapa de calor. À medida que o número de RSUs aumenta, o posicionamento expande progressivamente a partir do epicentro de tráfego, preenchendo gradualmente as regiões de maior densidade viária em um padrão circular. Esse comportamento é consistente com resultados reportados na literatura para a mesma instância de Colônia (Silva *et al.*, 2019).

Como evidenciado pela Figura 8, a grande maioria das RSUs é alocada nas mesmas posições pelos dois algoritmos, representadas em verde. As diferenças de posicionamento, destacadas em magenta para o guloso e em azul para o GRASP, são pontuais e restritas às regiões periféricas, reforçando a análise quantitativa da Seção 6.5: ambos os algoritmos convergem para soluções espacialmente equivalentes, o que explica a equivalência dos valores de área obtidos.

## 7 CONSIDERAÇÕES FINAIS

Este trabalho investigou o problema de posicionamento de RSUs em VANETs sob a ótica da maximização da área do gráfico da rede Delta, considerando uma quantidade fixa de unidades de infraestrutura. Para sua resolução, foram desenvolvidas e avaliadas duas

Figura 8 – Posicionamento das RSUs para os cenários de 1, 10, 20, 40, 80, 120, 140, 160 e 200 RSUs (5.000 veículos). Em verde, RSUs posicionadas em comum pelos dois algoritmos; em magenta, RSUs exclusivas do algoritmo guloso; em azul, RSUs exclusivas do GRASP. O mapa de calor indica a intensidade acumulada de tráfego em escala logarítmica.



Fonte: elaborado pelo autor (2026).

abordagens heurísticas: um algoritmo guloso baseado em ganho incremental e um algoritmo baseado na metaheurística GRASP, tendo como referência um modelo de programação inteira mista resolvido via CPLEX.

Os resultados experimentais demonstraram que o algoritmo guloso é capaz de encontrar soluções ótimas em instâncias de pequeno porte, igualando o modelo exato em todos os cenários de 1 a 10 RSUs avaliados com 100 veículos. Para instâncias de maior

escala, com 1.000 e 5.000 veículos, ambos os algoritmos produziram soluções equivalentes em todos os cenários avaliados. Acredita-se que esse comportamento está relacionado às características da instância de Colônia, cujo tráfego altamente concentrado em uma região central resulta em um espaço de soluções bem estruturado, favorecendo abordagens determinísticas.

A análise visual do posicionamento das RSUs revelou que ambos os algoritmos convergem para padrões espacialmente equivalentes, com as unidades preenchendo progressivamente as regiões de maior densidade viária à medida que o número de RSUs aumenta, comportamento consistente com resultados reportados na literatura para a mesma instância e que se mantém ao longo de todo o processo de otimização.

Os experimentos conduzidos sobre uma instância sintética com distribuição de tráfego mais uniforme corroboram a hipótese de que a diversificação proporcionada pelo GRASP é mais eficaz em cenários com múltiplas regiões promissoras de qualidade similar. Nesse cenário, o GRASP superou o algoritmo guloso na grande maioria dos valores de  $\eta$  avaliados, com ganhos de até +5,10% para 1.000 veículos e de até +1,78% para 5.000 veículos.

Como trabalho futuro, sugere-se a avaliação do algoritmo baseado na metaheurística GRASP em instâncias realísticas com distribuição de tráfego mais heterogênea, nas quais espera-se que a diversificação proporcionada pela metaheurística produza soluções superiores às obtidas pelo algoritmo guloso.

## REFERÊNCIAS

- Blum, J. J.; Eskandarian, A.; Hoffman, L. J. Challenges of intervehicle ad hoc networks. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 5, n. 4, p. 347–351, 2004. Citado na página 1.
- Church, R.; ReVelle, C. The maximal covering location problem. **Papers of the Regional Science Association**, Springer, v. 32, n. 1, p. 101–118, 1974. Citado na página 1.
- Feo, T. A.; Resende, M. G. C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, Kluwer Academic Publishers, v. 6, n. 2, p. 109–133, 1995. Citado 2 vezes nas páginas 2 e 8.
- Garcia, M. H. C.; Molina-Galan, A.; Boban, M.; Gozalvez, J.; Coll-Perales, B.; Şahin, T.; Kousaridas, A. A tutorial on 5G NR V2X communications. **IEEE Communications Surveys & Tutorials**, IEEE, v. 23, n. 3, p. 1972–2026, 2021. Citado na página 1.
- Hartenstein, H.; Laberteaux, K. P. A tutorial survey on vehicular ad hoc networks. **IEEE Communications Magazine**, IEEE, v. 46, n. 6, p. 164–171, 2008. Citado na página 1.
- Huo, Y.; Yang, R.; Jing, G.; Wang, X.; Mao, J. A multi-objective roadside units deployment strategy based on reliable coverage analysis in Internet of Vehicles. **Ad Hoc Networks**, Elsevier, v. 164, p. 103630, 2024. Citado na página 3.
- Marques, A. P. V. d. A. **Algoritmo Genético Aplicado ao Problema de Maximização da Área do Gráfico da Rede Delta**. 2020. Monografia (Trabalho de Conclusão de Curso), Belo Horizonte, 2020. Citado 5 vezes nas páginas 2, 3, 4, 15 e 16.

Mladenović, N.; Hansen, P. Variable neighborhood search. **Computers & Operations Research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997. Citado na página 3.

Sarubbi, J. F. M.; Silva, C. M. Delta-r: A novel and more economic strategy for allocating the roadside infrastructure in vehicular networks with guaranteed levels of performance. In: **IEEE/IFIP Network Operations and Management Symposium (NOMS 2016)**. Istanbul, Turquia: IEEE, 2016. p. 665–671. Citado 2 vezes nas páginas 2 e 3.

Sarubbi, J. F. M.; Silva, T. R.; Martins, F. V. C.; Wanner, E. F.; Silva, C. M. Allocating roadside units in VANETs using a variable neighborhood search strategy. In: **85. IEEE Vehicular Technology Conference (VTC Spring)**. Sydney, NSW, Austrália: IEEE, 2017. Citado 2 vezes nas páginas 2 e 3.

Silva, C.; Meira Jr., W. Design of roadside communication infrastructure with QoS guarantees. In: **IEEE Symposium on Computers and Communication (ISCC)**. Larnaca, Cyprus: IEEE, 2015. p. 439–444. Citado 4 vezes nas páginas 1, 3, 4 e 5.

Silva, C.; Meira Jr., W. Evaluating the performance of heterogeneous vehicular networks. In: **82. IEEE Vehicular Technology Conference (VTC2015-Fall)**. Boston, MA, EUA: IEEE, 2015. p. 1–5. Citado na página 2.

Silva, C. M.; Sarubbi, J. F. M.; Mokhtari, S.; Santos, L. A. L. d.; Silva, L. D.; Souza, F. S. H. d.; Guidoni, D. L.; Nogueira, J. M. S. RAGE: A novel strategy for solving non-polynomial problems through the random generation of solutions and incremental reduction of the number of candidates: A case study applied to the design of the network infrastructure for connected vehicles. **Expert Systems with Applications**, Elsevier, v. 213, p. 118900, 2023. Citado 4 vezes nas páginas 2, 3, 4 e 6.

Silva, C. M.; Silva, L. D.; Santos, L. A. L.; Sarubbi, J. F. M.; Pitsillides, A. Broadening understanding on managing the communication infrastructure in vehicular networks: Customizing the coverage using the delta network. **Future Internet**, MDPI, v. 11, n. 1, p. 1, 2019. Citado 2 vezes nas páginas 3 e 18.

Silva, T. R.; Sarubbi, J. F. M.; Martins, F. V. C.; Silva, C. M. Algoritmos baseados na metaheurística GRASP para implantação de unidades de comunicação em redes veiculares garantindo qualidade de serviço. In: **Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO)**. Vitória, ES, Brasil: SOBRAPO, 2016. Citado 2 vezes nas páginas 2 e 3.

Trullols, O.; Fiore, M.; Casetti, C.; Chiasserini, C.-F.; Barceló, J. Planning roadside infrastructure for information dissemination in intelligent transportation systems. **Computer Communications**, Elsevier, v. 33, n. 4, p. 432–442, 2010. ISSN 0140-3664. Citado na página 3.

Uppoor, S.; Trullols-Cruces, O.; Fiore, M.; Barcelo-Ordinas, J. M. Generation and analysis of a large-scale urban vehicular mobility dataset. **IEEE Transactions on Mobile Computing**, IEEE, v. 13, n. 5, p. 1061–1075, 2014. Citado 2 vezes nas páginas 2 e 10.

Zheng, Z.; Sinha, P.; Kumar, S. Alpha coverage: Bounding the interconnection gap for vehicular internet access. In: **28. IEEE International Conference on Computer Communications (INFOCOM)**. Rio de Janeiro, Brasil: IEEE, 2009. Citado na página 3.

## Agradecimentos

Agradeço ao Prof. Dr. João Fernando Machry Sarubbi pela excelente orientação e dedicação ao longo do desenvolvimento deste trabalho, ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) pelo suporte institucional, e aos professores Prof. Dr. Flávio Vinícius Cruzeiro Martins e Prof. Dr. Gustavo Campos Menezes pela disponibilidade e aceite do convite para avaliação deste trabalho. Agradeço também ao Prof. Dr. Gleiston Gonçalves Dias pela inspiração científica e pelo incentivo desde o início da graduação.



FOLHA DE APROVAÇÃO DE TCC Nº 15 / 2026 - DECOM (11.56.03)

Nº do Protocolo: 23062.032351/2026-06

Belo Horizonte-MG, 23 de junho de 2026.

**VINÍCIUS HENRIQUE SILVEIRA**

**POSICIONAMENTO DE INFRAESTRUTURAS DE COMUNICAÇÃO EM REDES  
VEICULARES: Algoritmos para a Maximização da Área da Rede Delta**

Trabalho de Conclusão de Curso apresentado  
ao Curso de **Engenharia de Computação** do  
Centro Federal de Educação Tecnológica de  
Minas Gerais, do Campus Nova Gameleira,  
como parte do requisito para obtenção do grau  
de **Engenheiro de Computação**.

Aprovado em 09 de JUNHO de 2026.

Banca Examinadora:

Prof. Dr. João Fernando Machry Sarubbi - Orientador  
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Flávio Vinicius Cruzeiro Martins  
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Gustavo Campos Menezes  
Centro Federal de Educação Tecnológica de Minas Gerais

**Belo Horizonte  
2026**

**(Assinado digitalmente em 24/06/2026 08:39 )**

FLAVIO VINICIUS CRUZEIRO MARTINS  
PROFESSOR ENS BASICO TECN TECNOLOGICO  
DECOM (11.56.03)  
Matrícula: 2630495

**(Assinado digitalmente em 23/06/2026 12:39 )**

GUSTAVO CAMPOS MENEZES  
PROFESSOR ENS BASICO TECN TECNOLOGICO  
DELCOM (11.58.04)  
Matrícula: 1550421

**(Assinado digitalmente em 23/06/2026 11:11 )**

JOAO FERNANDO MACHRY SARUBBI  
PROFESSOR ENS BASICO TECN TECNOLOGICO  
PPGLS (11.52.04)  
Matrícula: 1602873

Visualize o documento original em <https://sig.cefetmg.br/public/documentos/index.jsp>  
informando seu número: **15**, ano: **2026**, tipo: **FOLHA DE APROVAÇÃO DE TCC**, data de emissão:  
**23/06/2026** e o código de verificação: **4ed85ec159**