



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Rickson Simioni Pereira

VEITRA - Serviços de Gerenciamento de Recursos para Redes Veiculares para Amparo na Gestão de Transportes Inteligentes

São José do Rio Preto

2021

Rickson Simioni Pereira

**VEITRA - Serviços de Gerenciamento de Recursos para
Redes Veiculares para Amparo na Gestão de Transportes
Inteligentes**

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Orientador: prof. Dr. Rodolfo Ipolito Meneguette
Coorientador: prof. Dr. Luis Hideo Vasconcelos Nakamura

São José do Rio Preto
2021

P436v	<p>Pereira, Rickson Simioni</p> <p>VEITRA - Serviços de Gerenciamento de Recursos para Redes Veiculares para Amparo na Gestão de Transportes Inteligentes / Rickson Simioni Pereira.</p> <p>-- São José do Rio Preto, 2021</p> <p>67 f. : il., tabs., mapas</p> <p>Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto</p> <p>Orientador: Rodolfo Ipolito Meneguette</p> <p>Coorientador: Luis Hideo Vasconcelos Nakamura</p> <p>1. Alocação de recursos. 2. Computação em Fog. 3. Computação em nuvem. 4. Transportes inteligentes. I. Título.</p>
-------	---

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Rickson Simioni Pereira

**VEITRA - Serviços de Gerenciamento de Recursos para Redes
Veiculares para Amparo na Gestão de Transportes Inteligentes**

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Comissão Examinadora

Prof. Dr. Rodolfo Ipolito Meneguette
UNESP – Câmpus de São José do Rio Preto
Orientador

Prof. Dr. Lourenço Alves Pereira
ITA – São José dos Campos

Profa. Dra. Renata Spolon Lobato
UNESP – Câmpus de São José do Rio Preto

São José do Rio Preto
05 de outubro de 2021

Dedico este trabalho a todos que proporcionaram o mesmo se tornar realidade.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer ao Ricardo e Tânia, meus pais, além de minha irmã, Julia, pelo suporte oferecido para que eu pudesse realizar esse trabalho, o amor concedido e a o apoio incondicional em toda minha trajetória nos estudos.

Aos meus amigos que encontrei durante esse trajeto de minha vida, aos que permanecem até hoje e aos que já foram.

Aos meus além de amigos, parceiros de pesquisa, quais sem eles, não seria possível o desenvolvimento desse trabalho (Douglão, Quessada e Thiago - a ordem é alfabética para evitar discussões).

Ao professor Robson, qual me concebeu a oportunidade de vivenciar um novo horizonte na pesquisa e auxiliou ativamente na trajetória.

Finalmente, aos meus orientadores Rodolfo Meneguette e Luis Nakamura, pelo encorajamento, apoio, broncas, orientações, e o mais importante, pela amizade proporcionada.

*“I am just a child who has never grown up.
I still keep asking these ‘how’ and ‘why’ questions.
Occasionally, I find an answer.”
(HAWKING 1998)*

RESUMO

Um conjunto de veículos dispostos a cooperarem entre si através do paradigma da computação em nuvem pode ser definido como *Cloud Veicular*. Com o intuito de compartilhar recursos e facilitar a transação de informações, as redes veiculares se unem aos sistemas de *Clouds* móveis. Além disso, a medida que avançamos mais profundamente em serviços e sistemas orientados à informação, observamos claramente a importância e o impacto dos veículos inteligentes e conectados para a computação urbana. Novos paradigmas habilitados para nuvem impulsionaram o compartilhamento de informações e serviços. No entanto, tais paradigmas dependem fortemente da camada de comunicação subjacente, herdando os desafios originados das redes veiculares. Vários trabalhos foram elaborados para lidar com ambientes veiculares altamente dinâmicos em apoio à gestão e alocação de recursos eficazes, onde discutiremos alguns nesse trabalho. Os desafios da *Cloud Veicular* citados acima originam-se devido a alta mobilidade dos veículos, desse modo, a qualidade de serviço pode ser prejudicada caso a nuvem não se adapte rapidamente a esse fluxo de entradas e saídas constantes. Ou seja, para criar e manter *Clouds* veiculares, a descoberta e o gerenciamento de recursos são elementos fundamentais, devendo ser tratados com cuidado. Com isso em mente, nesse trabalho discutiremos sobre métodos já existentes na literatura, realizaremos o levantamento do estado da arte da área de alocação de recursos computacionais em redes veiculares, proporemos nosso método de alocação de recursos computacionais em redes veiculares baseado no paradigma de *Fog*, denominado VEITRA - Serviços de Gerenciamento de Recursos para Redes Veiculares para Amparo na Gestão de Transportes Inteligentes, onde as decisões são tomadas a partir da utilização um método matemático chamado *Analytic Hierarchy Process*, proposto por Thomas Saaty. Por fim, desenvolvimento é apresentado utilizando a linguagem de programação Python, juntamente do método Pearson e do simulador SUMO, após isso, foi feita a análise e a realização das comparações entre os resultados obtidos com outros resultados de métodos da literatura já estabelecidos.

Palavras-chave: Alocação de recursos. Computação em Fog. Computação em nuvem. Transportes Inteligentes.

ABSTRACT

A set of vehicles willing to cooperate with each other through the cloud computing paradigm can be defined as Cloud Vehicle. In order to share resources and facilitate the transaction of information, vehicle networks join mobile cloud systems. In addition, as we move deeper into information-oriented systems and services, we clearly see the importance and impact of smart, connected vehicles for urban computing. New cloud-enabled paradigms have driven the sharing of information and services. However, such paradigms depend heavily on the underlying communication layer, inheriting the challenges arising from vehicular networks. Several works have been developed to deal with highly dynamic vehicular environments in support of effective resource management and allocation, where we will discuss some in this work. The challenges of the Vehicle Cloud mentioned above originate from the high mobility of vehicles, thus, the quality of service can be harmed if the cloud does not adapt quickly to this constant input and output flow. In other words, to create and maintain Vehicle Clouds, the discovery and management of resources are fundamental elements and must be treated with care. With this in mind, in this work we will discuss methods already existing in the literature, we will carry out a survey of the state of the art in the area of computational resource allocation in vehicular networks, we will propose our method of computational resource allocation in vehicular networks based on the Fog paradigm, called VEITRA - Resource Management Services for Vehicle Networks to Support Intelligent Transport Management, where decisions are made using a mathematical method called Analytic Hierarchy Process, proposed by Thomas Saaty. Finally, development is presented using Python programming language, together with the Pearson method and the SUMO simulator, after that, we analyzed and compared the results with others from methods in the literature already established.

Keywords: Resource allocation. Fog Computing. Cloud Computing. Intelligent Transport.

LISTA DE FIGURAS

Figura 1 – Arquitetura de <i>Cloud</i> Veicular. Fonte: (MENEQUETTE; GRANDE; LOUREIRO, 2018b) (Adaptada).	22
Figura 2 – Comunicação V2V onde os veículos estão emitindo ou retransmitindo requisições entre si. (MOSTAED; ALDABAS; OLAVERRI-MONREAL, 2018)	23
Figura 3 – Comunicação por Overlay, onde o veículo vermelho é o requerente, o verde é um <i>superpeer</i> e os azuis são <i>peers</i> comuns. Fonte: (MENEQUETTE; BOUKERCHE; GRANDE, 2016)	24
Figura 4 – Comunicação dentro de um cluster, onde o veículo A requisita um serviço ao CH e ele por sua vez busca os recursos dentro do cluster. Fonte: (ARKIAN et al., 2015a)	25
Figura 5 – Comunicação com o auxílio de uma RSU. O CH se comunica diretamente com a RSU com a finalidade de atender as solicitações dos membros do <i>cluster</i> . Fonte: (ARKIAN et al., 2015a)	26
Figura 6 – Fluxograma do PSO. Fonte: (REYNOLDS, 1987; HEPPNER; GRENDER, 1990). Adaptada.	32
Figura 7 – Representação do VEITRA. Fonte: Pereira et al. (2021a). Adaptada.	45
Figura 8 – Fluxograma do funcionamento da entidade Controlador. Fonte: Própria.	51
Figura 9 – Fluxograma do funcionamento da entidade <i>Fog</i> . Fonte: Própria.	52
Figura 10 – Fluxograma do funcionamento da entidade Veículo. Fonte: Própria.	52
Figura 11 – Mapa de Colônia. Fonte: Simulation of Urban MObility.	54
Figura 12 – Análise comparativa de desempenho dos serviços atendidos. Fonte: Própria	55
Figura 13 – Análise comparativa de desempenho de serviços negados. Fonte: Própria.	57
Figura 14 – Análise comparativa de desempenho de serviços bloqueados. Fonte: Própria.	58

LISTA DE TABELAS

Tabela 1 – Comparação entre os diferentes tipos de <i>CLOUD</i>	20
Tabela 2 – Estágios de importância recomendado por Saaty. Fonte: (JAISWAL, 1997; SAATY, 1980) (Adaptada).	34
Tabela 3 – Comparação entre trabalhos relacionados. Fonte: Própria	42
Tabela 4 – Julgamento comparativo e valor numérico para MAHP e Classic AHP (EGUTI; GOMES; BELDERRAIN, 2007; LOOTSMA, 1993)	47
Tabela 5 – Fator influência AHP.	48
Tabela 6 – Fator influência MAHP.	48
Tabela 7 – Parâmetros de simulação	55

LISTA DE ABREVIATURAS E SIGLAS

CC	<i>Cloud Computing</i>
MCC	<i>Mobile Cloud Computing</i>
VCC	<i>Vehicular Cloud Computing</i>
V2V	Veículo-a-Veículo
V2I	Veículo-a-Infraestrutura
RSU	<i>Roadside Unit</i>
ILP	<i>Integer Linear Program</i>
IaaS	<i>Infrastructure as a Service</i>
PaaS	<i>Platform as a Service</i>
SaaS	<i>Software as a Service</i>
VM	<i>Virtual Machine</i>
SMDP	<i>Semi-Markov Decision Process</i>
AHP	<i>Analytic Hierarchy Process</i>
NEP	<i>Nash Equilibrium Point</i>
MAHP	<i>Multiplicative Analytic Hierarchy Process</i>
PSO	<i>Particle Swarm Optimization</i>
SDP	<i>Stochastic Dynamic Programming</i>
KKT	<i>KarushKuhn-Tucker</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	16
1.2	Objetivos	16
1.3	Organização do trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Diferentes Tipos de Clouds	18
2.1.1	Cloud Computing (CC)	18
2.1.2	<i>Mobile Cloud Computing (MCC)</i>	19
2.1.3	<i>Vehicular Cloud Computing (VCC)</i>	19
2.2	Estruturas de Alocação	21
2.2.1	Veículo-a-Veículo (V2V)	23
2.2.2	<i>Overlay</i>	23
2.2.3	Clusterização	24
2.2.4	<i>Roadside Unit (RSU)</i>	25
2.3	Métodos de Decisão de Alocação de Recursos	26
2.3.1	<i>Integer Linear Program (ILP – Programação Linear Inteira)</i>	27
2.3.2	Teoria dos Jogos	28
2.3.3	<i>Semi-Markov Decision Process</i>	29
2.3.4	<i>Particle Swarm Optimization</i>	31
2.3.5	<i>Analytic Hierarchy Process</i>	32
2.3.6	<i>Multiplicative Analytic Hierarchy Process</i>	36
2.3.7	Considerações parciais	37
3	ESTADO DA ARTE	38
3.1	Trabalhos relacionados	38
3.1.1	Considerações parciais	41
4	SERVIÇOS DE GERENCIAMENTO DE RECURSOS PARA REDES VEICULARES PARA AMPARO NA GESTÃO DE TRANSPORTES INTELIGENTES	43
4.1	VEITRA	43
4.2	Simulação	51

4.2.1	Cenário	53
4.2.2	Métricas de avaliação	53
4.3	Resultados	55
4.4	Considerações parciais	59
5	CONCLUSÃO	60
5.1	Considerações finais	60
5.2	Principais contribuições	61
5.3	Produções científicas	61
5.4	Trabalhos futuros	62
	REFERÊNCIAS	63

1 INTRODUÇÃO

Como influenciador direto da vida e da rotina das pessoas, o transporte urbano é um fator essencial do dia a dia. E com essa característica, ele tende a crescer e impactar cada vez mais na vida de qualquer pessoa, e conseqüentemente, no trânsito. Analisando dados estatísticos (DENATRAN, 2019), houve um aumento de aproximadamente 55% na quantidade de automóveis no Brasil entre o período de agosto de 2010 (aproximadamente 36 milhões) a agosto de 2019 (aproximadamente 56 milhões), indiretamente essa informação revela uma maior densidade na circulação de veículos. O avanço da tecnologia no setor automotivo nos traz novos paradigmas e novos desafios. À medida que a indústria automotiva evolui, diversos avanços tecnológicos são inseridos nos veículos, tanto para a segurança dos passageiros quanto para a melhoria de sua experiência (MENEGUETTE; GRANDE; LOUREIRO, 2018a). Esses avanços podem ser observados em veículos inteligentes, como carros autônomos e até carros elétricos (QUALCOMM, 2018; DANQUAH; ALTILAR, 2020).

Com isso, espera-se que as tecnologias móveis estejam cada vez mais disponíveis em veículos, a área privada e estadual estão pesquisando com muitos recursos para dar segurança, entretenimento e interatividade aos usuários, veículos e infraestruturas, pretendendo melhorar o transporte rodoviário (MENEGUETTE; BOUKERCHE, 2017). Os recursos computacionais nos veículos, sejam eles de processamento, armazenamento ou comunicação, em constante evolução, fornecem análises e detecções através dos dados coletados *in loco*. Esses recursos veiculares, se agregados, caracterizam um *pool* (ou lago, na tradução literal), que organiza de modo dinâmico uma nuvem dando suporte a toda complexidade trazida pelas redes móveis veiculares.

As redes veiculares são responsáveis por um tráfego de rede que se intensifica a cada ano. Um estudo realizado (WASICEK et al., 2015) estimou que durante o ano de 2013 foram realizadas cerca de 26 milhões de conexões, responsáveis por trafegar aproximadamente 480 terabytes de informações, isso apenas com dispositivos embarcados. Outra estimativa trazida pelo estudo é de um tráfego de 30 terabytes ao dia através de cerca de 150 milhões de conexões até o ano de 2020. Isso demonstra a necessidade de ampliação da infraestrutura de rede, para que haja espaço ao desenvolvimento de uma nova gama de serviços e também tecnologias que irão prover um melhor gerenciamento das *Clouds* veiculares.

À medida que os veículos se comunicam, um cenário totalmente novo pode ser abor-

dado quando o *Vehicle Cloud Computing* (VCC) entra em cena. O VCC é uma proposta que traz o paradigma da nuvem por meio das VANETS. Em uma *Vehicle Cloud* (VC), um grupo de veículos possui um *pool* de recursos que podem ser memória, processamento, armazenamento e compartilham seus recursos computacionais por meio do paradigma da nuvem. Desta forma, aspectos como Qualidade de Serviço (QoS), recursos sob demanda e escalabilidade são aplicáveis neste cenário (MENEGUETTE; GRANDE; LOUREIRO, 2018a; ZHANG; GRANDE; BOUKERCHE, 2015; DANQUAH; ALTILAR, 2020; MENEGUETTE; BOUKERCHE; De Grande, 2016). Portanto, a necessidade de algoritmos de alocação de recursos para gerenciar esses recursos e seu compartilhamento são fundamentais (ZHANG; SHEN; YANG, 2021; DANQUAH; ALTILAR, 2020). Logo, a *Cloud Veicular*, deve oferecer descoberta e alocação dinâmica de recursos para entregar de forma confiável esses serviços. A base para que isso ocorra é dada pela comunicação, que irá capacitar os veículos a cooperar entre si. A dinamicidade dessa comunicação é um dos desafios, já que os veículos devem se agregar para formar um sistema computacional, principalmente enquanto se movimentam (MENEGUETTE; BOUKERCHE; PIMENTA, 2019a).

Para auxiliar a nuvem veicular no gerenciamento dos recursos disponíveis e oferecer uma gama mais ampla de serviços, sem impactar a rede e a experiência do usuário, utiliza-se o paradigma de *Fog* (GAI et al., 2016). Além disso, a *Fog* pode agregar mais recursos computacionais, auxiliando no processamento das solicitações, auxiliando também na alocação e na aquisição de recursos para um melhor ajuste da solicitação do serviço em nuvem (LIU et al., 2018; WU et al., 2019).

O uso de Redes Ad Hoc Veiculares (VANETs) vem auxiliar em problemas como segurança de tráfego, eficiência de transporte, congestionamento de tráfego e vários outros (ZHANG et al., 2017; GOMIDES et al., 2020). As VANETs são apresentadas como uma tecnologia de comunicação sem fio para ajudar os motoristas com sua segurança e melhorar a gestão do tráfego (TAHERKHANI; PIERRE, 2012; MENEGUETTE; GRANDE; LOUREIRO, 2018a). Esta rede possibilita a comunicação entre os veículos e a troca de informações. A comunicação das VANETs pode ser em dois tipos principais, veículo a veículo (V2V) e veículo a infraestrutura (V2I) (DANQUAH; ALTILAR, 2020; MENEGUETTE et al., 2016). As VANETs possuem características particulares como topologia dinâmica, sem estrutura definida onde nós (veículos) se conectam e compartilham informações entre si sem controle centralizado (ZHANG; SHEN; YANG, 2021; MENEGUETTE; GRANDE; LOUREIRO, 2018a).

1.1 Motivação

A busca da melhora no transporte das pessoas juntamente da otimização desses serviços é a justificativa para a realização deste trabalho, por meio desse estudo será possível contribuir e entregar melhorias às tecnologias e aos sistemas computacionais no cenário de transporte inteligente.

Já estabelecidos na literatura, os trabalhos de Meneguette, Boukerche e Grande (2016), Arkian et al. (2015a), Liu e Fan (2018), Yu et al. (2015a), Zheng et al. (2015), Meng et al. (2018), Yang et al. (2017), Lin et al. (2018), Bhuvaneshwari, Paramasivan e Kandasamy (2018), Lieira et al. (2021) buscam resolver problemas relacionados a alocação de recursos computacionais, a variação entre eles vai de métodos mais tradicionais e complexos como Yu et al. (2015a), onde os autores utilizam o processo de decisão Semi-Markov, até métodos não tão tradicionais, como Lieira et al. (2021), onde os autores utilizaram um método meta-heurístico baseado no algoritmo do Lobo Cinzento. Essa variedade também acontece no objetivo principal do trabalho, podendo ser como Yang et al. (2017), onde os autores pesam o sigilo das informações no processo de alocação de recursos, ou também como Meng et al. (2018), onde os autores tem como base a realocação de recursos.

Motivados pela topologia dinâmica dos veículos por estarem em recorrente movimentação, propomos uma ferramenta de alocação e gerenciamento de recursos altamente adaptável e de empregabilidade de baixa complexidade em *Clouds* veiculares utilizando paradigmas de *Fog* baseado em um processo analítico hierárquico para a tomada de decisões, e assim, criar possibilidades de melhora no gerenciamento desses recursos dentro de uma nuvem veicular, proporcionando a maior utilização desses recursos e agilidade na alocação.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver um mecanismo de alocação e gerenciamento de recursos, tais como; armazenamento, processamento e largura de banda com intuito da utilização máxima desses recursos em uma nuvem veicular através dos paradigmas de *Fog*, um dos desafios que torna o gerenciamento complexo é a movimentação dos veículos em uma nuvem veicular, por conta de sua variação de movimentação de forma rápida e podendo um veículo entrar e sair de tal nuvem em questão de segundos.

Já os objetivos específicos desse trabalho são definidos com as seguintes contribuições:

1. Introdução e comparação sobre *Clouds* tradicionais, *Clouds* móveis e *Clouds* veiculares;

2. Investigação, discussão e comparação sobre as estruturas e métodos de alocação já estabelecidos na literatura;
3. Proposição e desenvolvimento de um mecanismo chamado VEITRA, qual realizará a alocação e o gerenciamento de recursos em uma nuvem veicular.
4. Implementação e simulação de dois algoritmos baseados no método Analytic Hierarchy Process e Multiplicative Hierarchy Process utilizando a linguagem de programação Python e o simulador de mobilidade urbana SUMO.
5. Otimizar e alcançar a utilização máxima dos recursos computacionais dentro de uma *Cloud* veicular utilizando paradigmas de *Fog*.

1.3 Organização do trabalho

A dissertação deste trabalho de mestrado está estruturada da seguinte forma:

O **Capítulo 2** traz a fundamentação teórica do trabalho. Envolvendo os tipos de *Cloud*, sendo eles: *Cloud Computing*, *Mobile Cloud Computing* e *Vehicular Cloud Computing*. Além disso, são apresentadas algumas estruturas de alocação de recursos e alguns métodos de decisão de alocação de recursos já existentes na literatura.

O **Capítulo 3** apresenta e compara o estado da arte da área, envolvendo trabalhos relacionados com foco em alocação de recursos computacionais.

O **Capítulo 4** apresenta a metodologia de pesquisa do trabalho, exibindo a modelagem do cenário de simulação, as métricas utilizadas, o funcionamento do mecanismo desenvolvido denominado VEITRA e a avaliação dos resultados alcançados durante as simulações realizadas.

O **Capítulo 5** apresenta as conclusões dessa dissertação, suas produções bibliográficas, contribuições e ideias para o desenvolvimento de trabalhos futuros promovendo a escalabilidade do mecanismo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Diferentes Tipos de Clouds

A flexibilidade que a *Cloud Computing* (CC) introduziu para o fornecimento sob demanda de recursos e serviços pela Internet permitiu o seu amplo uso tanto por usuários domésticos quanto por empresas. Os interesses acelerados em sua versatilidade fizeram da computação em nuvem uma das principais tendências tecnológicas, com investimentos maciços e esforços corporativos para migrar seus negócios para esse novo paradigma.

Juntamente com a ascensão da CC, os dispositivos móveis cresceram em quantidade, recursos, acessibilidade e popularidade. Esse crescimento permitiu uma série de avanços tecnológicos e, conseqüentemente, entrega de serviços e aplicativos de alta qualidade por meio deles, o que promoveu uma nova área de pesquisa chamada *Mobile Cloud Computing* (MCC) (SANAIE et al., 2014), (DINH et al., 2013). Os projetos neste novo campo concebem dispositivos móveis como terminais de entrada e saída, e assumem que os servidores remotos na nuvem processem tarefas intensivas de computação.

Os benefícios e desafios no campo da *Cloud* motivaram vários trabalhos que promoveram a introdução do *framework* de *Cloud* veicular (STEPHAN, 2011). Esse trabalho inicial abstrai os problemas atuais e classifica as soluções como uma questão de aproveitar os recursos veiculares subutilizados, como conectividade de rede, energia computacional, armazenamento e capacidade de detecção, que podem ser compartilhados com os proprietários dos veículos.

2.1.1 Cloud Computing (CC)

Nos últimos anos, foram feitos grandes avanços na *Cloud Computing*, graças aos avanços tecnológicos na computação e na rede (KIM, 2009). A computação em nuvem apresentou uma mudança de paradigma na indústria global de tecnologia da informação. A CC é definida como um modelo para permitir o acesso onipresente e conveniente da rede sob demanda a um *pool* compartilhado de recursos de computação configurável, como redes, servidores, armazenamento, aplicativos e serviços, que podem ser rapidamente provisionados e disponibilizados com o mínimo esforço de gerenciamento ou interação do provedor de serviços (MELL; GRANCE, 2011). A *Cloud Computing* introduziu várias características novas para soluções de computação distribuída quando comparadas aos sistemas locais tradicionais como: provisionamento sob demanda e alocação de serviços e

recursos de acordo com a necessidade e por tempo determinado.

Os serviços em nuvem atingiram um determinado nível de maturidade e são oferecidos em três classes principais: Infraestrutura como serviço (IaaS – *Infrastructure as a Service*), Plataforma como serviço (PaaS – *Platform as a Service*) e *Software* como serviço (SaaS – *Software as a Service*) (KIM, 2009).

2.1.2 *Mobile Cloud Computing (MCC)*

A MCC é definida como uma infraestrutura que fornece a possibilidade dos dispositivos móveis não processarem e não armazenarem dados dentro deles; essas tarefas são executadas remotamente por outras entidades (DINH et al., 2013). Portanto, aplicativos da *Mobile Cloud Computing* transferem o estresse do armazenamento de dados e do poder de computação, dos dispositivos móveis para a nuvem.

Espera-se que a MCC ofereça um potencial significativo para aplicativos de tecnologia da informação e apresente características únicas, como mobilidade, portabilidade e comunicação. Como recursos únicos dela, podem ser notados a extensão da vida útil da bateria, melhoria na capacidade de processamento e armazenamento e aumento na confiabilidade (DINH et al., 2013).

Devido a evolução direta da *Cloud Computing* e da computação móvel, várias vantagens da *Mobile Cloud Computing* são expostas na literatura, como provisionamento dinâmico, facilidade de acesso e integração (DINH et al., 2013; MELL; GRANCE, 2011).

2.1.3 *Vehicular Cloud Computing (VCC)*

A indústria automotiva vem construindo veículos consideravelmente inteligentes, que já contêm recursos robustos de computação, comunicação, armazenamento e energia. Esses recursos diferenciam veículos modernos de dispositivos móveis, os quais assumem recursos limitados. O VCC, nesse caso, trabalha na direção oposta ao paradigma do MCC. A MCC tenta aproveitar os recursos dos dispositivos, enquanto a VCC permite que recursos subutilizados ou disponíveis do veículo, sejam coletados. Ela lida com a extrema volatilidade do ambiente e dos recursos, o movimento distinto de veículos, a demanda de aplicativos e também da disponibilidade esporádica.

Na literatura, já estão estabelecidos alguns trabalhos direcionados à alocação de recursos em nuvem veicular (YU et al., 2013; MENEGUETTE et al., 2017; TAO et al., 2015; ZHENG et al., ; MENG et al., 2015; ARKIAN et al., 2015b; DAI et al., 2015; PYUN; LEE; CHO, 2016; MIAO et al., 2016). Zheng et al. (ZHENG et al.,) e Meng et al. (MENG et al., 2015) consideram o Processo de Decisão Semi-Markov (SMDP) para alocar recursos

em uma nuvem veicular. Arkian et al. (ARKIAN et al., 2015b) e Miao et al. (MIAO et al., 2016) usavam a lógica Fuzzy para auxiliar no processo de alocação de recursos, bem como na seleção de veículos que seriam usados na comunicação entre diferentes clusters. Dai et al. (DAI et al., 2015) e Pyun et al. (PYUN; LEE; CHO, 2016) introduziram modelos matemáticos para lidar com o problema de alocação na VANET, que consideram a comunicação direta entre o veículo e a RSU. Yu et al. (YU et al., 2015b) estabeleceu uma teoria dos jogos com o objetivo de ajudar no processo de alocação de recursos. Todos esses trabalhos usam uma infraestrutura de borda para realizar a alocação de recursos na nuvem veicular. Além disso, grande parte deste trabalho não considera que o veículo tenha recursos que possam ser agregados na nuvem, bem como a mobilidade do veículo no momento da alocação do serviço.

O compartilhamento de recursos entre os veículos pode ser realizada através da cooperação entre os mesmos, com o objetivo de criar uma nuvem veicular (MENEQUETTE; BOUKERCHE; De Grande, 2016). O conjunto de tais recursos cria um grupo de serviços que são cedidos para outros veículos. Em vista disso, a nuvem veicular precisa suprir a entrega confiável de serviços, descoberta de recursos e alocação dinâmica de recursos. Para esse fim, a nuvem veicular pode acessar a infraestrutura através da *Roadside Unit* (RSU), por uma rede celular ou qualquer outro suporte externo (GERLA, 2012), por conta das características da VANET, como alta mobilidade de veículos e constante mudança de topologia, o gerenciamento e a descoberta de recursos para a nuvem veicular se tornam uma tarefa desafiadora.

A nuvem, os motoristas, os passageiros e usuários se beneficiam da agregação eficiente de todos os recursos circundantes e da sua disponibilização efetiva como serviços em nuvem. Conforme definido, a VCC refere-se a um grupo de veículos que contêm recursos físicos, de computação, comunicação e sensoriamento que podem ser coordenados e alocados dinamicamente, além de atribuídos a usuários autorizados (OLARIU; HRISTOV; YAN, 2013). A Tabela 1 apresenta uma breve comparação das principais características de CC, MCC e VCC.

Tabela 1 – Comparação entre os diferentes tipos de *CLOUD*

Características	Recursos Móveis	Processamento	Limitação Energética	Armazenamento	Formação Autônoma	Flexibilidade de Recursos	Arquitetura de Rede	Recursos físicos
CC	Não	Alta	Não	Alto	Não	Estático	Cliente-Servidor	Servidores Locais ou Remoto
MCC	Sim	Baixa	Sim	Baixo	Não	Estático	Cliente-Servidor	Dispositivos móveis locais ou servidor remoto
VCC	Sim	Mediana	Não	Mediano	Sim	Dinâmico	Ponto-a-Ponto ou Cliente-Servidor	Veículos locais ou servidor remoto

Em relação à capacidade de processamento e capacidade de armazenamento, é perceptível que os servidores em nuvem convencionais apresentam os mais poderosos recursos de computação local, e um veículo moderno geralmente inclui um computador de bordo,

que possui recursos de computação menores, mas muito maiores do que os de um dispositivo móvel. Além disso, a mobilidade é uma parte valiosa e inerente dos recursos físicos da MCC e da VCC, enquanto os recursos de uma CC tradicional geralmente estão localizados no *datacenter*. O CC também não é restringido pelo consumo de energia devido ao fornecimento constante no *datacenter*; da mesma forma, o VCC não é limitado pela energia, pois um veículo contém uma bateria de grande capacidade e o motor em funcionamento que a recarrega continuamente. No entanto, a duração da bateria de dispositivos móveis limita a MCC. Por fim, o VCC é o único tipo de nuvem que se baseia em recursos físicos altamente dinâmicos e pode ser formado autonomamente, dependendo da situação do tráfego.

2.2 Estruturas de Alocação

Para que a locação de recursos possa efetivamente acontecer, deve-se primeiramente realizar a descoberta dos recursos ociosos. Para que isso ocorra, os veículos devem se organizar de algum modo, dada a circunstância em que eles se encontram. Por exemplo, se um veículo não consegue detectar nenhum outro ao seu alcance, logo ele será sua própria *Cloud*, sem ninguém para compartilhar seus recursos ociosos. Porém, ao detectar outro veículo, ele poderá decidir se inicia uma comunicação veículo-a-veículo ou através de um *overlay*, ou até mesmo se dará início à formação de um *cluster*.

Várias abordagens foram discutidas na literatura sobre o problema da alocação de recursos na nuvem veicular (ZHENG et al., ; MENG et al., 2015; TAO et al., 2015; YU et al., 2013; MENEGUETTE et al., 2017; MENEGUETTE; BOUKERCHE; PIMENTA, 2019b). Neste trabalho, listamos alguns deles. Tao et. al (TAO et al., 2015) propôs um esquema de alocação de recursos em nuvem não cooperativo, que utilizava um método de iteração Gauss-Seidel (G-S) para reduzir o tempo de cálculo do NEP. Esse esquema considerou os processos de junção e saída dos nós e modelou um controle de precisão para melhorar a iteração do fluxo. Dessa forma, ele explorou a teoria dos jogos para modelar os comportamentos de transmissão dos veículos, também conhecido como jogo de alocação de recursos na nuvem. Além disso, esse esquema também utilizou a RSU para aumentar o tempo de comunicação entre o veículo e o servidor de dados. Nesse esquema, cada veículo tenta minimizar seu custo ou maximizar sua utilidade e usou o método de iteração G-S para encontrar a utilidade e taxa de fluxo ideais dos nós. Para lidar com o problema de convergência da alocação de recursos em redes veiculares, foi utilizada uma convergência do método de iteração G-S.

Na base de tudo isso está uma arquitetura em comum, desenvolvida para criar e ge-

reenciar uma *Cloud Veicular*. A maioria desses trabalhos divide a arquitetura em camadas *Onboard*, *Communication* e *Cloud* (WHAIDUZZAMAN et al., 2014) e (BITAM; MEL-LOUK; ZEADALLY, 2015), conforme ilustrado na Figura 1.

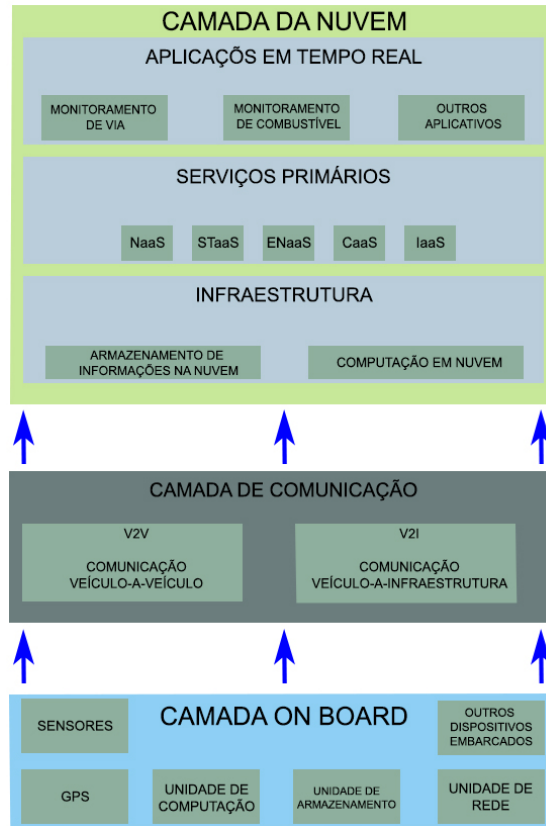


Figura 1 – Arquitetura de *Cloud Veicular*. Fonte: (MENEQUETTE; GRANDE; LOUREIRO, 2018b) (Adaptada).

A primeira camada (*Onboard*) é responsável por abstrair informações de sensores incorporados como GPS, câmera e outros dispositivos que os veículos possam conter.

A segunda camada (*Communication*) garante a conexão entre a camada interna e a nuvem. Para isso, os veículos contêm dispositivos que utilizam protocolos de comunicação sem fio, como 802.11p, 3G e 4G. Esses dispositivos permitem a classificação dessa camada em duas partes: V2V e a V2I. O V2V refere-se a veículos que estão no mesmo alcance, portanto, as informações podem ser propagadas pelos veículos até que cheguem à *Cloud*. O V2I consiste na troca de informações entre veículos, infraestruturas e a nuvem.

A terceira camada (*Cloud*) responde pela agregação de recursos e fornece serviços de aplicativos aos usuários. Essa camada contém três partes: aplicativos em tempo real, serviços primários e infraestrutura.

A subcamada de aplicativos é capaz de oferecer serviços e aplicativos adequados para restrições em tempo real e que suportam os principais serviços da nuvem, como informações sobre a estrada. A subcamada de infraestrutura apresenta componentes de

armazenamento e computação que suportam a análise dentro da própria nuvem veicular. Já a subcamada de serviços primários é alimentada pelo conteúdo da subcamada de infraestrutura e fornece serviços para a subcamada de aplicativos, como por exemplo IaaS e CaaS.

2.2.1 Veículo-a-Veículo (V2V)

Cada veículo apresenta um conjunto de recursos que podem estar disponíveis para outros veículos na rede veicular e cada veículo deve investigar o ambiente ao redor para identificar os serviços e recursos disponíveis através de conexões V2V, propagando mensagens de controle (SIBAÏ et al., 2015).

Quando se considera o veículo como uma própria nuvem, os protocolos de disseminação de dados podem ser a estratégia mais comum para disseminar uma requisição entre os veículos propagando mensagens até chegarem a uma nuvem que contém o recurso procurado, como exemplificado através da Figura 2. No entanto, essa técnica gera sobrecarga na rede devido a uma tempestade de *broadcast*.

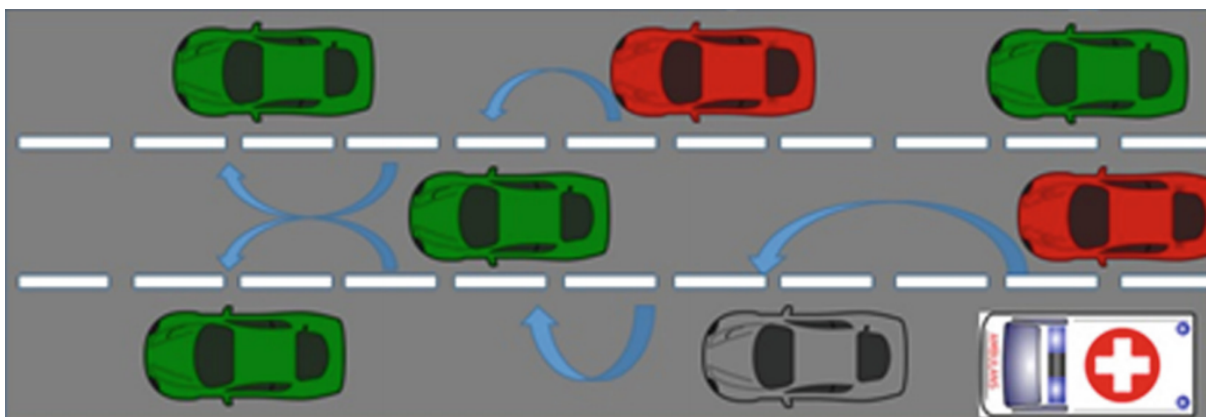


Figura 2 – Comunicação V2V onde os veículos estão emitindo ou retransmitindo requisições entre si. (MOSTAED; ALDABAS; OLAVERRI-MONREAL, 2018)

2.2.2 Overlay

Esse tipo de estrutura cria uma camada de abstração entre os veículos na rede para facilitar a comunicação e o gerenciamento de recursos. Cada um dos nós da rede funciona como cliente e servidor, permitindo o compartilhamento de serviços e dados sem a necessidade de um servidor central, parecido com a arquitetura P2P (MENEQUETTE; BOUKERCHE; GRANDE, 2016). Essa arquitetura pode ser descentralizada ou híbrida.

Na arquitetura descentralizada não existe um elemento central para executar o controle do recurso e todos os nós têm a mesma capacidade. Nela, quando um par precisa de um

serviço, ele transmite uma mensagem para seus vizinhos. Se o vizinho tiver o recurso solicitado, ele envia uma resposta ao solicitante. Caso contrário, os vizinhos transmitem a solicitação aos seus vizinhos.

No híbrido, os nós podem ser divididos em *superpeers* e *peers*. Os *Superpeers* são nós que possuem maior capacidade de processamento, memória e outros recursos. Os *peers* consistem em nós que possuem uma capacidade comum a todos os outros nós. Os *superpeers* têm alguns recursos disponíveis, portanto, eles gerenciam apenas parte dos recursos disponíveis na nuvem. Logo, uma vez que um par precisa de um determinado recurso, ele envia uma solicitação ao *superpeer* ao qual está anexado. O *superpeer*, ao receber essas informações, verifica se pode responder à solicitação. Caso puder, ele envia uma resposta ao mesmo nível. Caso contrário, verifica se outros *superpeers* possuem o recurso solicitado. A Figura 3 ilustra esse exemplo.

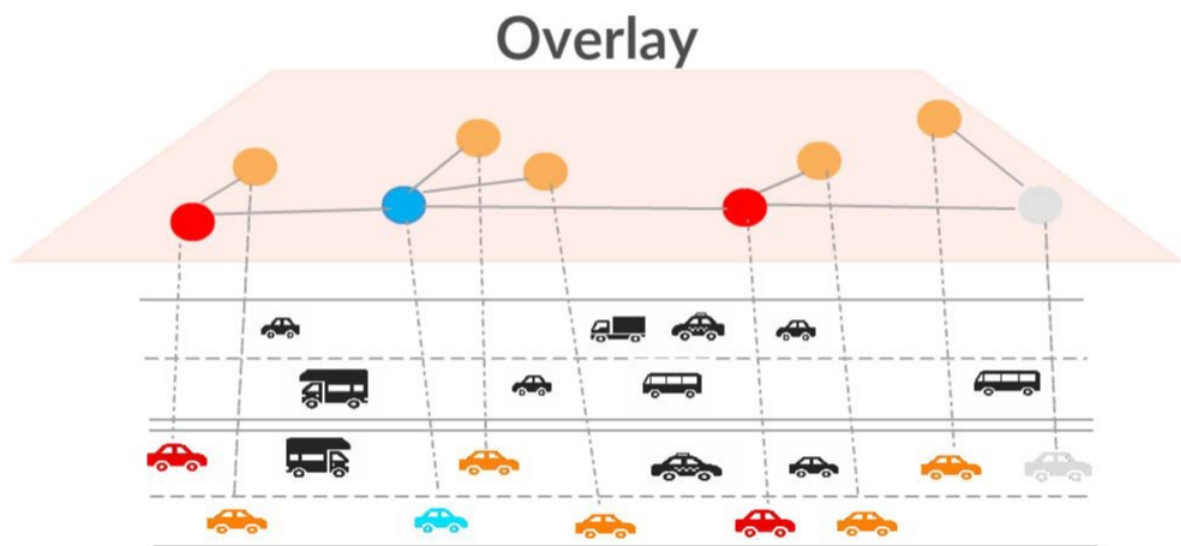


Figura 3 – Comunicação por Overlay, onde o veículo vermelho é o requerente, o verde é um *superpeer* e os azuis são *peers* comuns. Fonte: (MENEGUETTE; BOUKERCHE; GRANDE, 2016)

Essa técnica é usada não apenas para facilitar a comunicação entre os veículos, mas também para criar uma camada de conexão. A técnica também permite a distribuição de recursos entre os veículos, permitindo a busca de recursos sem sobrecarregar as redes com mensagens de controle (MENEGUETTE; BOUKERCHE; GRANDE, 2016).

2.2.3 Clusterização

O *cluster* é uma técnica para agrupar veículos para facilitar a comunicação. Existem vários critérios envolvidos no agrupamento de veículos, como velocidade, direção, número

de veículos e veículos de acompanhamento. A maneira mais simples de agrupar veículos em uma cidade é através da localização em que os veículos estão; portanto, todos os veículos que estão na mesma rua, seguindo um ao outro ou em um quarteirão são colocados no mesmo *cluster* (MENEQUETTE; BOUKERCHE; PIMENTA, 2019a).

Um *cluster* simples é composto pelos seguintes elementos (ARKIAN et al., 2015a):

- **Cabeça de *Cluster* (CH):** que gerencia o *cluster* e seus recursos;
- **Veículos Membros:** que são aqueles que participam de um *cluster* específico;
- **Veículos Desconhecidos:** que são veículos que ainda não participam do *cluster*.

O CH é escolhido através de um processo de eleição entre os veículos pertencentes ao *cluster*. Após essa seleção, essas informações são disseminadas para todos os membros do *cluster* para que eles possam saber quem é o controlador desse *cluster*. A Figura 4 exemplifica essa estrutura.

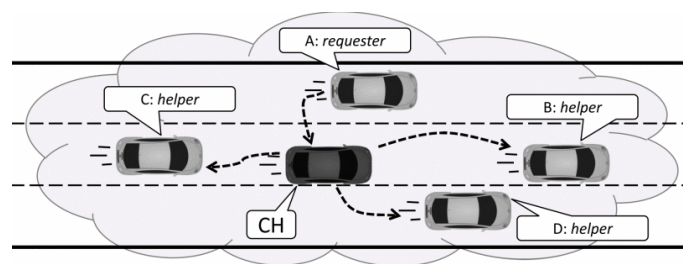


Figura 4 – Comunicação dentro de um cluster, onde o veículo A requisita um serviço ao CH e ele por sua vez busca os recursos dentro do cluster. Fonte: (ARKIAN et al., 2015a)

Essa técnica é usada para criar uma estrutura de comunicação entre veículos para reduzir o número de mensagens enviadas pela rede. Qualquer solicitação ou comunicação no *cluster* passa primeiro pelo CH para encaminhar posteriormente ao veículo de destino.

2.2.4 Roadside Unit (RSU)

Uma RSU é uma estrutura física com poder de processamento, armazenamento e comunicação que é estrategicamente alocada em um ponto para prover uma melhor comunicação e compartilhamento de recursos entre veículos.

Como as RSUs estão potencialmente fisicamente conectadas à *Cloud* tradicional, elas precisam de novos aspectos e componentes para lidar com a alta mobilidade dos veículos (ARKIAN et al., 2015a).

Portanto, Uma técnica ideal pode fazer uso de elementos de armazenamento de informações em uma nuvem tradicional e auxiliar os intermediários (RSUs) na detecção e coordenação de recursos (NING; HUANG; WANG, 2019).

Uma RSU pode servir como intermediária entre as nuvens para identificar e coordenar recursos em uma rede veicular. Portanto, quando um veículo solicita um serviço à RSU, ele pode procurar a melhor nuvem para atender à solicitação de serviço, alocando os recursos em caso de aceitação.

Ela também pode servir de intermediária entre *Clouds* veiculares formadas com a estrutura de clusterização, por exemplo. Nesse caso o *cluster* irá definir o CH e ele será o responsável por gerenciar as solicitações e transmiti-las à RSU, conforme exibido na Figura 5.

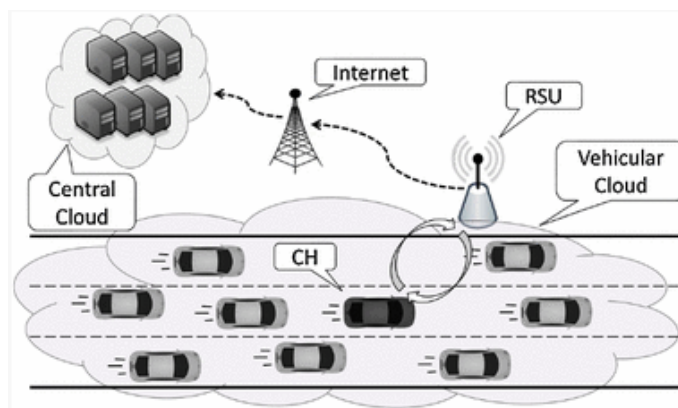


Figura 5 – Comunicação com o auxílio de uma RSU. O CH se comunica diretamente com a RSU com a finalidade de atender as solicitações dos membros do *cluster*. Fonte: (ARKIAN et al., 2015a)

2.3 Métodos de Decisão de Alocação de Recursos

Para o gerenciamento de recursos em *Cloud* deve-se considerar uma série de fatores como eficiência, Qualidade do serviço e disponibilidade (YU et al., 2015a).

O gerenciamento de recursos em uma infraestrutura baseada em *Cloud* veicular precisa lidar com a alocação de recursos, que normalmente é obtida com a criação de VMs. Além disso, um mecanismo de gerenciamento de recursos deve permitir a transferência de dados entre RSUs, caso elas estejam previstas, e evitar falhas de recursos resultantes da falta de atendimento aos requisitos de QoS ou da alta mobilidade do veículo. Outro aspecto a considerar é a análise de recursos, que indica a ordem em que as solicitações são atendidas.

Como esses gerenciadores de recursos baseados em infraestrutura e dependem dos aspectos da alocação de recursos, a transferência e o agendamento de recursos entre RSUs,

também devem ser considerados o gerenciamento de recursos realizado pela comunicação direta entre os veículos. Como resultado, o método no qual a comunicação é estabelecida entre os veículos em trânsito é um elemento essencial nesse tipo de gerenciamento.

2.3.1 *Integer Linear Program* (ILP – Programação Linear Inteira)

Uma das técnicas usadas para gerenciar recursos em *Clouds* veiculares é a programação linear inteira (ILP), que consiste em uma otimização matemática na qual todas as variáveis devem ser inteiras. O ILP busca a distribuição eficiente de recursos limitados para atingir um objetivo específico, geralmente maximizando lucros ou minimizando custos (LIU; FAN, 2018). Portanto, esse objetivo é expresso através de uma função linear, chamada função objetivo. Além disso, é necessário definir quais atividades consomem recursos e em que proporções os recursos são consumidos. Esta informação é apresentada na forma de inequações lineares, uma para cada recurso. Para o conjunto dessas inequações e desigualdades, isso é chamado de restrições de modelo.

Geralmente, existem muitas maneiras de distribuir recursos escassos entre várias atividades, desde que essas distribuições sejam consistentes com as restrições de um determinado modelo. No entanto, o que é investigado é a função objetivo, com foco na maximização do lucro ou na minimização de custos. Essa solução é chamada de solução ideal. Assim, a programação linear é usada para encontrar a solução ideal para um problema, uma vez que o modelo linear é definido, delimitando a função objetivo e as restrições lineares.

Em uma *Cloud* veicular, o ILP é usado para minimizar o custo das operações de gerenciamento de recursos, para reduzir a necessidade de migrar uma VM de uma nuvem para outra ou de um elemento de rede para outro (LIU; FAN, 2018). Para isso, a estratégia usa parâmetros de rede, como sobrecarga, largura de banda e tamanho da VM, entre outros elementos de redes veiculares, e os elementos em questão como modelos de restrição.

Para exemplificar, considere uma *Cloud* veicular com capacidade T para atender solicitações de serviço. A *Cloud* fornece os serviços de gerenciamento de tráfego, alerta de evento ou uma combinação desses dois. Essa nuvem também tem uma capacidade limitada para recursos de processamento (R_p) e de armazenamento (R_a) disponíveis que podem ser usados para atender esses serviços. Esses recursos podem ser alocados em quantidades diferentes, dependendo se a solicitação é para gerenciamento de tráfego (R_{pt} , R_{at}) ou para alerta de evento (R_{pe} , R_{ae}). Suponha que a alocação do recurso de processamento tenha um custo para o sistema (C_p) e um custo para alocar os recursos de armazenamento (C_a). Portanto, se assumirmos que os veículos fazem solicitações de serviços de gerenciamento

de tráfego (gt) ou de alerta de eventos (ae), o número ideal de serviços fornecidos pelo sistema pode ser descrito como um problema de programação linear.

$$C_p \cdot gt + C_s \cdot ea \quad (2.1)$$

$$gt + ea \leq T \quad (2.2)$$

$$R_{pt} \cdot gt + R_{pe} \cdot ae \leq R_p \quad (2.3)$$

$$R_{at} \cdot gt + R_{ae} \cdot ae \leq R_a \quad (2.4)$$

$$gt \geq 0 \quad ea \geq 0 \quad (2.5)$$

Dado o exemplo, a equação 2.1 é a função objetivo, que irá calcular custos mínimos. A equação 2.2 expressa a quantidade máxima de requisições que podem ser atendidas, por isso devem ser menor ou igual à capacidade total de oferta de serviços (T). A expressão 2.3 representa o limite de recursos de processamento, enquanto a expressão 2.4 representa o limite de recursos de armazenamento. Por fim, a equação 2.5 mostra a condição matemática tornando verdadeira a requisição de serviços por algum veículo.

2.3.2 Teoria dos Jogos

A teoria dos jogos é uma estratégia matemática projetada para modelar fenômenos que podem ser observados quando dois ou mais tomadores de decisão interagem entre si (YU et al., 2015a). Esse esquema é usado em situações em que as decisões de um agente ou jogador dependem ou influenciam as escolhas de outros.

Portanto, um jogo consiste em um conjunto de jogadores representados por $J = j_1, j_2, \dots, j_n$. Cada jogador j_i tem um conjunto finito E_i de opções chamado de estratégia de jogador que ele pode usar. Denota-se por E um conjunto de todos os perfis de estratégia, definidos da seguinte forma (YU et al., 2015a):

$$E = \prod_{i=1}^n E_i = E_1 \times E_2 \times \dots \times E_n \quad (2.6)$$

Também é exposto que cada elemento $e \in E$ e representa um resultado do jogo, também chamado de vetor de estratégia ou perfil de estratégia do jogo. Além disso, cada jogador deve ter uma ordem de preferência, que deve ser uma relação completa, transitiva, reflexiva e binária para os possíveis resultados do jogo. Essa relação indica que o jogador prefere mudar sua estratégia e , com isso, o resultado do jogo muda de um vetor de estratégias para outro. Uma maneira simples de representar isso é uma função de utilidade ou benefício $u_i : E \rightarrow R$ para cada jogador i , que retorna um valor numérico para cada vetor de estratégia. Portanto, se i mudar sua estratégia, migrando de um vetor

de estratégia e para um vetor e' , essa ação ocorrerá apenas porque $u_i(e') > u_i(e)$ (YU et al., 2015a).

Quando os jogadores atingem um resultado em que cada jogador não tem interesse em mudar sua estratégia, um resultado estável é alcançado, então esse vetor de estratégias que representa esse resultado é um equilíbrio de Nash.

Para ilustrar como essa estratégia funciona, considere uma *Cloud* veicular no qual dois serviços (jogadores) foram solicitados pelo usuário. Esses serviços podem ser atendidos ou bloqueados. Se ambos forem atendidos, eles terão um custo de 4 para cada serviço. Se apenas um serviço (A) for atendido e o outro (B) for bloqueado, o serviço A custará 1 e o outro custará 5. Da mesma forma, se B for atendido e A for bloqueado, B custará 1 e A terá um custo de 5. Se ambos estiverem bloqueados, o custo será 2.

Então, define-se um conjunto J com dois jogadores A e B que tem os seguintes conjuntos de estratégias para cada jogador:

$$S_A = (\textit{Atendido}, \textit{Bloqueado})$$

$$S_B = (\textit{Atendido}, \textit{Bloqueado})$$

$$u_A = (\textit{Atendido}, \textit{Atendido}) = 4$$

$$u_A = (\textit{Atendido}, \textit{Bloqueado}) = 1$$

$$u_A = (\textit{Bloqueado}, \textit{Atendido}) = 5$$

$$u_A = (\textit{Bloqueado}, \textit{Bloqueado}) = 2$$

$$u_B = (\textit{Atendido}, \textit{Atendido}) = 4$$

$$u_B = (\textit{Atendido}, \textit{Bloqueado}) = 1$$

$$u_B = (\textit{Bloqueado}, \textit{Atendido}) = 5$$

$$u_B = (\textit{Bloqueado}, \textit{Bloqueado}) = 2$$

Assim, para este exemplo, o resultado em que ambos os status são atendidos é um equilíbrio de Nash porque, para atingir esse resultado, nenhum deles minimiza o custo do sistema alterando suas escolhas individualmente.

2.3.3 *Semi-Markov Decision Process*

O SMDP, ou na tradução, Processo de Decisão Semi-Markov visa modelar a tomada de decisão em situações em que os resultados são parcialmente aleatórios e parcialmente sob o controle de um tomador de decisão (BAYKAL-GÜRSOY, 2011). O SMDP fornecem uma estrutura matemática para modelar processos em que as transições entre estados são

probabilísticas; é possível observar em que estado está um processo e interferir periodicamente nos tempos de decisão, executando ações (PUTERMAN, 2014). Cada ação tem uma recompensa ou custo, dependendo do estado do processo.

Portanto, um SMDP é definido como uma tupla (S, A, T, R) (ZHENG et al., 2015), onde:

- S é um conjunto de estados em que o processo pode estar presente;
- A é um conjunto de ações que podem ser executadas em diferentes períodos de decisão;
- $T : SXAXS \mapsto [0, 1]$ é uma função que fornece a probabilidade de que o sistema passe para um estado $s' \in S$. Essa função assume que o processo estava em um estado $s \in S$ e o agente decidiu adotar ação $a \in A$ (denota-se por: $T(s' | s, a)$);
- $R : SXA \mapsto R$ é uma função que fornece o custo (ou recompensa) para tomar uma decisão $a \in A$ quando o processo está em um estado $s \in S$.

Para ilustrar o uso de SMDP, é assumido que uma *Cloud* veicular pode oferecer uma *pool* de recursos que contém R unidades de recursos que suportam uma classe de serviço K . Além disso, a classe de serviço $i, i \in \{1, 2, \dots, K\}$ requer recursos b_i para atender a seus requisitos de serviço. Outro ponto é o horário de chegada de uma nova solicitação, que é modelado como um processo de Poisson a uma taxa de λ_n (ZHENG et al., 2015).

Portanto, o estado do sistema pode ser descrito pela ocupação de recursos na *Cloud*. Assim, o espaço de estados pode ser indicado da seguinte maneira:

$$S = \{s \mid s = (n, e)\}, \text{ onde } e \in E = \{Req, Teq\} \text{ e } n = \{n_1, n_2, \dots, n_i\} \quad (2.7)$$

A variável n_i representa o número de veículos em uma *Cloud* que solicita a classe de serviço i . Da mesma forma, os recursos alocados na nuvem local é dado por $\sum_{k=1}^K b_k * n_k \leq R$.

Enquanto e representa um evento que ocorre no sistema, dentro do conjunto de eventos E temos *Req* que representa a chegada de uma solicitação, e *Ter* que representa o termino de um serviço solicitado da classe k que a requisição vem da nuvem.

Nesse ambiente SMDP, a nuvem precisa tomar uma decisão com base em seu espaço de ação após um período conhecido (ZHENG et al., 2015). Assim, em cada época de decisão, a nuvem escolhe uma ação a no espaço de ação A_s , que é definido da seguinte maneira:

$$A_s = \{-1\}, e = Teq \quad A_s = \{0, 1, 2, \dots, K\} \quad e = Req \quad (2.8)$$

Ou seja, $a = -1$ indica que o controlador não precisa tomar decisões, apenas atualizar o consumo de recursos no sistema já que o processo está sendo encerrado. Para $a = 0$ tem-se a indicação de que a solicitação de serviço da classe k foi rejeitada pelo controlador. Por fim, quando $a = k$, sendo $k \neq \{-1, 0\}$, temos que a solicitação foi aceita e o serviço da classe k foi alocado no sistema.

Em (ZHENG et al., 2015) também é definido a equação da probabilidade de transição de estados (s para s'), assim como a representação do tempo de espera até a próxima época de decisão. O autor também demonstra a descoberta da receita ou da penalidade do sistema dada uma ação a em um estado s e inclui nos cálculos os custos do sistema.

A modelagem proposta visa controlar a aceitação ou rejeição de solicitações de serviço no sistema, buscando maximizar o uso desses recursos na nuvem. Existem alguns modelos propostos para considerar não apenas serviços, mas também características da entrada e saída de veículos na nuvem (ZHENG et al., 2015).

2.3.4 Particle Swarm Optimization

Seguindo uma linha menos convencional, existem também os métodos meta-heurísticos para a tomada de decisões, Particle Swarm Optimization (PSO) sendo um dos mais conceituados da literatura, proposto por Kennedy e Eberhart em 1995, o método é desenvolvido em um conceito simples, podendo ser implementado em poucas linhas de código, sendo necessário para a sua aplicação operações básicas matemáticas e recursos computacionais básicos em termos de memória ou velocidade. Uma das razões para o desenvolvimento do método é o comportamento humano, possuindo uma abstração importante quando comparada a outros métodos desenvolvidos previamente por Reynolds em 1987, Heppner e Grenander em 1990 (REYNOLDS, 1987; HEPPNER; GREANDER, 1990), onde respectivamente o núcleo para o desenvolvimento foram o comportamento de revoadas e cardumes.

A principal diferença entre o comportamento humano e os outros dois grupos são as limitações dos peixes e pássaros, onde seus movimentos físicos se ajustam baseados em evitar predadores, procurar parceiros e comida, ou otimizar parâmetros corporais, como por exemplo, temperatura. Por outro lado, os humanos não se limitam aos movimentos físicos, porém, aos cognitivos ou variáveis de experiência também, o exemplo dos autores é que nós não andamos no mesmo ritmo e viramos em uníssono, como as revoadas ou cardumes, ao invés disso, tendemos a ajustar nossas atitudes e nosso modo para se adaptar com as de nossos colegas sociais.

Proposto há quase duas décadas, o algoritmo do PSO passou por melhorias em sua implementação desde então. Para o entendimento do funcionamento, a figura 6 exibe

suas etapas, i) onde há o início do *swarm*, após isso, ii) há avaliação *fitness* das partículas da simulação, então, iii e iv) o cálculo da posição ideal histórica individual e *swarm* das mesmas, depois disso, v) é realizada a atualização das velocidades e posições das partículas de acordo com a equação de atualização de velocidade e posição, por fim vi) há a verificação se a condição final foi alcançada, se sim, o processo é finalizado, se não, há o reinício no segundo passo.

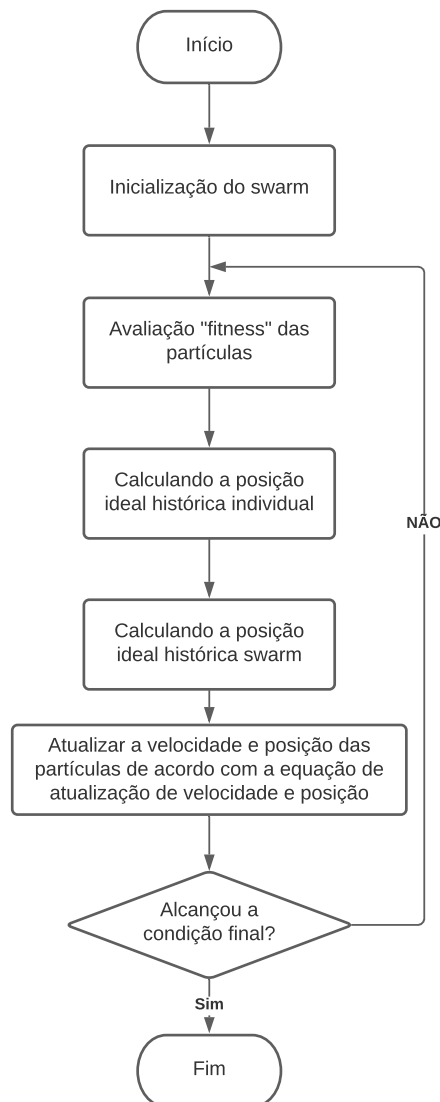


Figura 6 – Fluxograma do PSO. Fonte: (REYNOLDS, 1987; HEPPNER; GREANDER, 1990). Adaptada.

2.3.5 Analytic Hierarchy Process

Na década de 1970 o matemático Thomas L. Saaty desenvolveu uma técnica estruturada com objetivo de organizar e analisar decisões complexas, com base em matemática e

psicologia, denominado processo hierárquico analítico (AHP). O processo é amplamente estudado e aprimorado em diversas áreas de estudo. Isso se dá conta pela sua aplicação particular em diferentes situações de tomada de decisão, como em áreas do governo, da saúde, na educação, etc.

O objetivo central da teoria introduzida por Saaty é a otimização no estudo dos sistemas de tomada de decisão. O AHP fornece uma técnica estruturada para a tomada de decisão de problemas com vários critérios envolvidos, usando comparação por pares entre os valores numéricos de cada parâmetro e seus graus relativos de importância, a fim de ajustar seus pesos em tempo de execução. Como resultado, quanto maior o peso, mais importante é o critério correspondente. É recomendado por Saaty a utilização de uma escala com 9 níveis, com o nível 1 expondo a indiferença de importância de um critério em relação a outro, já no nível 9 significa a extrema importância de um critério sobre o outro, exibindo ainda estágios intermediários entre eles.

Além disso, desconsiderando as comparações entre os próprios critérios, que representam 1 na escala, apenas metade das comparações precisam ser feitas, porque a outra metade constitui-se das comparações recíprocas na matriz de comparações, que são os valores recíprocos já comparados.

Na Tabela 2 toda a exposição realizada acima pode ser visualizada, desde o primeiro estágio de importância até o último, com suas definições e explicações, respectivamente.

A empresa Woods Gordon utilizou o método AHP para tomada de decisões na área da saúde na década de 80, no Canadá. Eles utilizavam-o para quatro aplicações, para o programa de construção e reforma do hospital, para o planejamento estratégico de necessidades de sistemas de informação, seleção de contratantes e alocação de trabalho de manutenção militar (MITCHELL; WASIL, 1989).

A tomada de decisão era definida em 10 passos, são eles:

1. Definição do problema.

Inicialmente, é necessário definir o problema que o AHP resolverá, podendo ser alocação de recursos, escolha da melhor alternativa, etc. Após a exposição, é possível o fornecimento de conselhos sobre a estrutura do problema, abrangendo todos os fatores envolvidos e relacionando os objetivos e as alternativas existentes.

2. Seleção do grupo de decisão

O grupo de decisão pode incluir a equipe técnica do hospital, consultores (podendo ser externos, para uma perspectiva mais holística da situação). É interessante a escolha de um grupo que coletivamente possua conhecimento e informações necessárias para confrontar diretamente o problema.

Tabela 2 – Estágios de importância recomendado por Saaty. Fonte: (JAISWAL, 1997; SAATY, 1980) (Adaptada).

Intensidade de Importância	Definição	Explicação
1	Mesma importância	As duas atividades contribuem igualmente para o objetivo.
3	Importância pequena de uma sobre a outra	A experiência e o julgamento favorecem levemente uma atividade em relação à outra.
5	Importância grande ou essencial	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra.
7	Importância muito grande ou demonstrada	Uma atividade é muito fortemente favorecida em relação à outra; sua dominação de importância é demonstrada na prática.
9	Importância absoluta	A evidência favorece uma atividade em relação à outra com o mais alto grau de certeza.
2, 4, 6, 8	Valores intermediários entre os valores adjacentes	Quando se procura uma condição de compromisso entre duas definições.
Recíprocos dos valores acima de zero	Se a atividade i recebe uma das designações diferentes acima de zero, quando comparada com a atividade j , então j tem o valor recíproco quando comparada com i .	Uma designação razoável
Racionais	Razões resultantes da escala	Se a consistência tiver de ser forçada para obter valores numéricos n , somente para completar a matriz

3. Identificação dos problemas e dos objetivos

A primeira tarefa do grupo de decisão é a identificação dos problemas e dos objetivos que deverão ser considerados na decisão. Com isso, a energia pode ser focalizada na geração de matrizes de comparação de pares.

4. Desenvolvimento da estrutura de hierarquia

A estrutura é organizada, possui uma hierarquia e uma descrição do problema de decisão em termos das escolhas a serem feitas. Levando em consideração os objetivos que devem ser atingidos, os interesses e todos os critérios de decisão.

5. Julgamento da importância dos fatores de decisão

Após a estrutura de tomada de decisão ser estabelecida por completa, a próxima etapa é catalogar a importância dos fatores de decisão, esses fatores são avaliados em pares, julgando suas importâncias para o objetivo qual eles contribuem.

6. Avaliação alternativas

As alternativas que são expostas no nível inferior da hierarquia são comparadas pelo grupo de tomada de decisão.

7. Relatório dos resultados

A classificação dos fatores de decisão são sintetizados e a média das prioridades das alternativas são calculadas.

8. Checagem de razoabilidade

O grupo de tomada de decisão deve considerar a razoabilidade que essas decisões vão implicar nos resultados, considerando que alguns resultados podem não ser adequados, e caso seja necessário, haja a revisão do processo.

9. Finalizar escolhas

Após a checagem minuciosa da razoabilidade dos resultados, há a reunião do grupo de tomada de decisão para a finalização das decisões.

10. Documentação

O passo final do processo é a documentação de tudo que foi realizado anteriormente, realizado de forma detalhada, podendo conter de 20 a 200 páginas dependendo da complexidade do projeto.

2.3.6 Multiplicative Analytic Hierarchy Process

Sendo uma evolução do já existente AHP, o Multiplicative Analytic Hierarchy Process (MAHP) (LOOTSMA, 1993; RAMANATHAN, 1997) foi proposto com o objetivo de preenchimento de algumas lacunas de seu antecessor, onde as principais críticas edificam-se na montagem da matriz de decisão, qual é baseada em um vetor Perron-Frobenius, limitando a matriz a números positivos, além disso, o cálculo de pontuação final das alternativas para a tomada de decisão é realizado baseado na regra de agregação da média aritmética. O método de decisão baseado em MAHP proposto por Lootsma considera dois métodos de avaliação em seu trabalho. No primeiro método, inicialmente deve-se coletar a informação de preferência (indiferente, fraco, rigoroso, forte ou muito forte) de duas alternativas A_j e A_k comparadas a partir de um critério C_i , após isso deve-se realizar a conversão da declaração verbal do tomador de decisão d em um valor numérico $r_{jkd}^{(i)}$ em uma escala geométrica, como por exemplo, uma escala discreta com escalões constituindo uma série com progressão geométrica. Após isso, os autores utilizam uma regressão logarítmica para calcular o critério único de fator de impacto $\bar{v}_i(A_j), j = 1, \dots, n$, aproximando os valores subjetivos das alternativas sob o critério C_i . O autor considera que os fatores de impactos não são únicos, portanto, eles devem ser normalizados como mostra a 2.9 por conta de seu grau multiplicativo de liberdade.

$$\sum_{j=1}^n \bar{v}_i(A_j) = 1, i = 1, \dots, m \quad (2.9)$$

Considerando seu segundo caso de avaliação, onde dois critérios são comparados mutuamente, o método ganha um grau a mais de complexidade. Os autores sugerem que o tomador de decisão considere duas alternativas reais ou imaginárias, onde a propriedade de preferência da primeira decisão sobre o primeiro critério é igual a sua propriedade de preferência da outra alternativa sobre o segundo critério. Após isso, é requisitado que haja a indicação se há indiferença entre as duas alternativas e os dois critérios simultaneamente, ou, um dos dois critérios possui uma decisiva preferência por uma das duas alternativas, sendo elas: i) fraca, ii) rigorosa, iii) forte, iv) muito forte. Em seguida isso as declarações de julgamentos são convertidas em valores numéricos em uma escala geométrica particular. Regressões logarítmicas nesse caso produzem pesos normalizados $\bar{w}(C_i), i = 1, \dots, m$ para cada respectivo critério, a seguir, a Equação 2.10 exhibe que finalmente é realizado o passo de agregação gerando as pontuações finais multi-criteriosas $\bar{f}(A_j)$ por meio de uma regra de agregação média geométrica onde \bar{c}_i destaca o peso $\bar{w}(C_i)$ e a é o fator normalização para garantir que as pontuações finais somem à unidade. Portanto, consi-

derando essas quantidades, as alternativas são classificadas de forma inequívoca em um ranqueamento subjetivo de preferência quando operamos com escalas geométricas.

$$\bar{f}(A_j) = a \prod_{i=1}^m [\bar{v}_i(A_j)]^{\bar{c}_i} \quad (2.10)$$

2.3.7 Considerações parciais

Baseado nos métodos apresentados previamente, fica-se decidido que o método selecionado para o desenvolvimento da ferramenta de gerenciamento de recursos será o AHP. Por conta de sua capacidade de decisão em situações complexas, incluindo e medindo todos os fatores que caracterizam-se importantes, qualitativa e quantitativamente mensuráveis, sejam eles tangíveis ou intangíveis, para aproximar-se de um modelo realista (SAATY, 1980). Entretanto, ao decorrer do desenvolvimento dessa dissertação, foi-se estabelecido também o aprimoramento do mecanismo com a utilização do método MAHP, onde suas alterações sobre o método tradicional surtem efeito direto na alocação e no gerenciamento dos recursos, devido sua alteração na modelagem da matriz de decisão.

3 ESTADO DA ARTE

Nesse capítulo serão apresentados trabalhos já estabelecidos na literatura na área de alocação de recursos computacionais, cada qual com sua especificidade e seus objetivos, utilizando diferentes métodos e cenários de aplicação.

3.1 Trabalhos relacionados

Vários trabalhos discutem de modo avançado sobre os métodos de descoberta e alocação de recursos em *Clouds* veiculares (MENEGUETTE; BOUKERCHE; GRANDE, 2016; ARKIAN et al., 2015a; LIU; FAN, 2018; YU et al., 2015a; ZHENG et al., 2015; MENG et al., 2018; YANG et al., 2017; LIN et al., 2018; BHUVANESWARI; PARAMASIVAN; KANDASAMY, 2018; LIEIRA et al., 2021). Cada um com seu ambiente de trabalho e buscando a solução de um problema específico, mas tangenciando a questão de gerenciamento e alocação recursos em diferentes estruturas de uma nuvem veicular.

Há a utilização do protocolo baseado em P2P com objetivo de busca e gerenciamento de recursos em uma *Cloud* veicular (MENEGUETTE; BOUKERCHE; GRANDE, 2016). Este protocolo é baseado nos conceitos de Gnutella e introduz a técnica de *overlay* para ajudar na descoberta e gerenciamento de recursos na nuvem. O protocolo é dividido em dois componentes principais: O primeiro é responsável pelo gerenciamento de recursos, que cria e controla uma sobreposição e ajuda na descoberta e gerenciamento de recursos; O segundo é responsável pelo roteamento, que trata da propagação de mensagens de solicitação e resposta.

Também existe na literatura uma *Cloud* Veicular chamada COHORT (ARKIAN et al., 2015a). Ela emprega RSUs para prestar serviços aos veículos. A COHORT também auxilia RSUs no gerenciamento e descoberta de novos recursos em redes veiculares. Como uma RSU é acessível apenas para os veículos localizados dentro do seu alcance de transmissão, o COHORT utiliza de CaaS para aumentar a cobertura do serviço disponível.

O gerenciamento de recursos em nuvem veicular também pode ser realizado através do ILP (LIU; FAN, 2018), que minimiza a sobrecarga de reconfiguração. O método visa reduzir atrasos na infraestrutura e replicações de serviços. O modelo ILP consiste em minimizar conjuntamente a sobrecarga de reconfiguração, migrações de VM, modificações no plano de controle, número de *hosts* de serviço e atrasos na infraestrutura da nuvem. Para conseguir isso, os autores usaram pesos para controlar a prioridade da sobrecarga

de reconfiguração, para que a minimização das migrações da VM tenha prioridade sobre as modificações do plano de controle.

A teoria de jogos (YU et al., 2015a) também é utilizada no gerenciamento de recursos em nuvens veiculares e compartilhamento cooperativo de recursos ociosos. Essa abordagem é dividida em duas etapas. Primeiro, o provedor de serviços em nuvem analisa a receita e verifica se funciona sozinho ou se integra a um mercado em nuvem. Em seguida, o provedor de serviços aluga recursos de outros provedores de serviços. Nessa abordagem, os autores usaram a otimização de Pareto para auxiliar os prestadores de serviços em suas ações, o que garantiu que sua utilidade aumentasse ou pelo menos não diminuísse. A coleção ideal de Pareto consolida-se gradualmente a cada rodada.

Na literatura também há a proposição de um esquema de alocação de recursos para maximizar a recompensa de uma *Cloud* veicular baseada em SMDP (ZHENG et al., 2015). Essa abordagem utiliza quatro estágios para formular o problema de alocação de recursos. (1) O espaço de estado representa os recursos atuais e solicita os estados na *Cloud* veicular. (2) O espaço de ação é um conjunto de ações que podem ser usadas por uma *Cloud* veicular com base em seu estado atual. (3) Uma recompensa consiste na receita e nos custos que permitem aplicar um modelo que calcula e analisa a soma das recompensas. (4) Uma probabilidade de transição calcula a probabilidade de que um estado do sistema mude para outro estado sob uma ação específica. O algoritmo visa maximizar a recompensa total esperada a longo prazo da *Cloud* veicular.

Meng et al. (2018) propôs um esquema de alocação de recursos baseado em uma estratégia de realocação usando a teoria de otimização de Lyapunov. Os autores pretendem aumentar o número de serviços garantidos, diminuindo o custo e reduzindo a frequência de realocação. Neste trabalho, os autores adaptam e ajustam a alocação de recursos ao ambiente dinâmico da rede veicular. A realocação frequente e descontrolada de recursos, por outro lado, pode resultar em um grande número de trocas de sinalização de controle. Para controlar o número de realocações, os autores empregaram uma geometria estocástica para analisar a interferência no pelotamento de veículos. Para diminuir a complexidade computacional, eles mudam ao longo do problema de otimização dinâmica de tempo proposto em um problema de otimização determinística utilizando a hipótese de otimização de Lyapunov para decidir o conjunto de serviços servidos com base nas mudanças dinâmicas dentro do arranjo em cada abertura.

Yang et al. (2017) além da alocação de recursos, os autores propõem uma política que aumenta a segurança na camada física assumindo uma informação de estado de canal desatualizada (CSI) que visa otimizar a taxa máxima e mínima de sigilo dos usuários com feedback restrições de atraso para comunicações V2I. Os autores empregaram multi-

plexação por divisão ortogonal de frequência (OFDM) multiusuário na camada física nesse cenário. Eles desenvolveram um algoritmo de duas etapas para formular a otimização em dois subproblemas: i) alocação de subportadora; ii) alocação de energia; que visa reduzir a complexidade computacional. Além disso, o trabalho inclui um algoritmo guloso para alocar os recursos da subportadora. Resolvendo as condições de KarushKuhn-Tucker (KKT), os autores podem obter a política de alocação de energia ótima em uma forma semifechada.

Lin et al. (2018) propuseram uma abordagem de alocação de recursos focada na oferta de serviços aos usuários finais, na qual os autores dividiram a otimização em duas etapas. Primeiramente, eles assumiram que a largura de banda total disponível está otimamente alocada, sendo possíveis oito alternativas no esquema de alocação, que é baseado no algoritmo Lagrangiano, portanto, os autores podem obter a faixa de largura de banda total disponível. No entanto, alguns deles podem ser sobrepostos. Em segundo lugar, os autores dividiram a largura de banda total disponível em pequenas faixas com base nas faixas obtidas na primeira etapa do processo, portanto, todas as possibilidades são calculadas nessas pequenas faixas. No longo prazo, todas as atribuições ideais são associadas, e a demonstração de utilidade ideal é obtida para a computação Fog veicular.

Bhuvaneswari, Paramasivan e Kandasamy (2018) propuseram um método de alocação de recursos baseado em um Modelo de Programação Dinâmica Estocástica (SDP). Neste trabalho, os autores consideram um projeto bifásico para alocação de recursos, consideram também que todos os veículos possuem a mesma quantidade de recursos computacionais, e uma configuração homogênea de sensores, subsistemas de comunicação e controle. Segundo os autores, se os veículos estão em movimento em uma única direção, não há necessidade de utilizar todos os recursos disponíveis, para isso, o veículo seleciona uma zona angular ótima para transmissão de dados, para isso há um cálculo a fazer uma zona angular para transmissão de dados. Esses cálculos são representados como vetor, considerando a zona angular ótima, a zona atendida e a zona fútil. A zona de atenção é representada por aquele que contém o ângulo dos recursos disponíveis. A zona fútil não pode ser considerada para alocação de recursos, pois seus recursos não são considerados no vetor.

Lieira et al. (2021) propuseram um mecanismo para otimização de tarefas em *Clouds* veiculares utilizando uma técnica meta-heurística chamada Lobo Cinzento. Os autores dividiram o processo de alocação de tarefas em três estágios, onde a rede veicular presente corresponde a comunicação entre os veículos, caracterizando uma V2V. Além disso os autores utilizam a técnica de clusterização dos veículos, qual é responsável por compartilhamento e agrupamento dos recursos, formando um leque de recursos e gerando uma

Cloud veicular. A clusterização é realizada utilizando o agrupamento espacial baseado em densidade (DBSCAN), identificando os *clusters* baseado na densidade espacial dos veículos, considerando suas coordenadas. O DBSCAN considera que a vizinhança deve ultrapassar um limite para cada ponto de um *cluster* e para determinado raio na *Cloud* veicular ele contém um determinado número de pontos. O principal objetivo dos autores é maximizar a disponibilidade dos recursos para performar tarefas dentro de uma *Cloud* veicular.

3.1.1 Considerações parciais

Na Tabela 3 podemos visualizar uma comparação entre os métodos, cenários e o problema tratado entre os trabalhos apresentados acima, com isso é possível verificar que na literatura já existem diversas formas de gerenciamento e de alocação dos recursos dentro de uma nuvem veicular ou ambientes móveis, criando desafios e abrindo possibilidades no desenvolvimento de novas propostas na busca de uma melhor eficiência. Como já exposto, o principal objetivo do desenvolvimento de um mecanismo de alocação de recursos nesse trabalho é o aumento da disponibilidade de recursos oferecidos dentro de uma rede veicular utilizando um método analítico hierárquico.

Tabela 3 – Comparação entre trabalhos relacionados. Fonte: Própria

Trabalhos	Métodos	Cenários	Problema tratado
Meneguette, Boukerche e Grande (2016)	P2P	VANETs	Descoberta e alocação de recursos
Arkian et al. (2015a)	Fuzzy	VANETs	Descoberta e alocação de recursos
Liu e Fan (2018)	ILP	Multi-Cloudlet	Redução de atrasos na infraestrutura e replicação de serviços
Yu et al. (2015a)	Teoria dos jogos	VANETs	Aumento de utilidade nos provedores de serviços
Zheng et al. (2015)	SMDP	VANETs	Maximização de recompensa a longo prazo em uma Cloud veicular
Meng et al. (2018)	Lyapunov	VANETs	Realocação de recursos
Yang et al. (2017)	KKT	VANETs	Alocação de recursos baseada em sigilo das informações
Lin et al. (2018)	Lagrangian	VANETs	Otimização do serviço aos usuários finais
Bhuvaneswari, Paramasivan e Kandasamy (2018)	SDP	VANETs	Descoberta e alocação de recursos
Lieira et al. (2021)	Lobo cinzento	VANETs	Otimização de tarefas
VEITRA	AHP e MAHP	VANETs	Maximização da quantidade de serviços atendidos

4 SERVIÇOS DE GERENCIAMENTO DE RECURSOS PARA REDES VEICULARES PARA AMPARO NA GESTÃO DE TRANSPORTES INTELIGENTES

Nesse capítulo serão apresentados todo o processo para o desenvolvimento dos serviços de gerenciamento de recursos para redes veiculares para amparo na gestão de transportes inteligentes aqui propostos. O processo vai desde sua modelagem de cenário urbanom, qual foi baseado na cidade de Colônia, na Alemanha, implementação dos algoritmos desenvolvidos baseados nos métodos *Analytic Hierarchy Process* e *Multiplicative Analytic Hierarchy Process*, sendo esses, já comprovados pela literatura quando o objetivo é a tomada de decisões. Além de simulações que foram realizadas dentro do SUMO dos algoritmos desenvolvidos e de outros já estabelecidos na literatura, além de exposição e comparação dos resultados alcançados.

4.1 VEITRA

A solução que propomos chama-se VEITRA - Serviços de Gerenciamento de Recursos para Redes **V**eiculares para Amparo na Gestão de **T**ransportes Inteligentes, baseado em paradigmas de *Fog*. VEITRA adota uma abordagem que conta com uma série de veículos inteligentes que estão em constante movimento, trocando mensagens entre si e com Roadside Units (RSUs) que estão distribuídas no centro urbano. Utilizamos um cenário baseado na densidade veicular de Colônia, uma cidade da Alemanha. O esquema tem f Fogs onde cobrem todo o cenário com recursos limitados rl .

O VEITRA é projetado para lidar com o gerenciamento e alocação de recursos do veículo, a fim de maximizar o uso dos recursos do veículo. Para isso, consideramos veículos com recursos computacionais ociosos, como processamento e armazenamento, nos quais esses recursos podem ser disponibilizados para a formação de uma nuvem veicular. Portanto, os veículos junto com a *Fog* formarão uma infraestrutura para fornecer serviços para outros veículos. Assim, o VEITRA criará uma nuvem veicular na qual os recursos são compartilhados por veículos e *Fogs* configurados para oferecer serviço em nuvem a outros veículos.

Portanto consideramos que a nuvem veicular será composta por um conjunto de *Fogs* e veículos, quais ambos possuirão recursos computacionais disponíveis. Esses recursos de veículos ociosos serão agregados aos recursos da *Fog*, onde serão gerenciados pela própria *Fog*. Podemos considerar que as *Fogs* são um conjunto de recursos computacionais em uma RSU. Assim, neste trabalho, uma *Fog* é uma RSU com recursos disponíveis. O intuito da agregação de recursos de ambos os dispositivos é atender os serviços solicitados pelos veículos, tais como, serviços de entretenimento, serviços de troca de rotas de tráfego, troca de mensagens, entre outros. Consideramos que os veículos neste cenário entram e saem das *Fogs*. Cada veículo tem sua própria identificação $v_{id} = [1, n]$ e recursos $v_r = (process_r, mem_r, armazen_r, largb_r)$ que podem ser processamento, RAM, armazenamento e largura de banda.

Cada serviço possui recursos computacionais, como: processamento, armazenamento e tempo de execução para ser executado com integralidade. Assim, as *Fogs* tomarão todas as decisões sobre a alocação de recursos e qual *Fog* aloca recursos para atender melhor ao serviço solicitado. A Figura 7 representa a abstração do sistema, que é dividido em uma visão de três camadas. A primeira camada consiste na Camada Física, onde veículos e RSUs compartilham o ambiente urbano. Os veículos se comunicam entre si e chegam à RSU por meio de comunicação multi-hop. RSUs, nesta perspectiva física, servem como pontos de acesso de conexão de referência onde os veículos utilizam para se agrupar para um compartilhamento eficiente de sua computação e comunicação. A segunda camada corresponde à Camada de Recursos, onde RSUs mapeiam os recursos disponíveis no ambiente através dos veículos que podem alcançá-los. Com base no contato bem-sucedido de cada veículo com a RSU, os recursos veiculares a bordo disponíveis ou indexados são indexados, referenciados e mapeados. A terceira camada representa a Camada de *Fog*, onde os recursos mapeados são agrupados para compor uma *Fog*. Este agrupamento permite o gerenciamento e alocação dos recursos.

A *Fog* aloca os recursos computacionais ociosos dos veículos, bem como aloca os recursos disponíveis para atender aos serviços exigidos pelos veículos. Para resolver o problema de alocação de recursos em *Fogs*, consideramos que as mesmas podem ser conectadas a veículos v e que esses veículos compartilham seus recursos r_{veh} ociosos com as *Fogs*. Esses recursos podem ser alocados para atender aos s serviços solicitados por veículos que circulam nas vias. Os serviços solicitados podem solicitar recursos disponíveis r_{serv} . Portanto, cada *Fog* f terá um conjunto de v_f veículos conectados a ele que compartilham um total de r_f recursos.

Assim, o veículo v_{id} que está percorrendo a rota solicita um serviço de *Fog* e aloca os recursos disponíveis para atender este serviço de forma a executar com eficiência o serviço

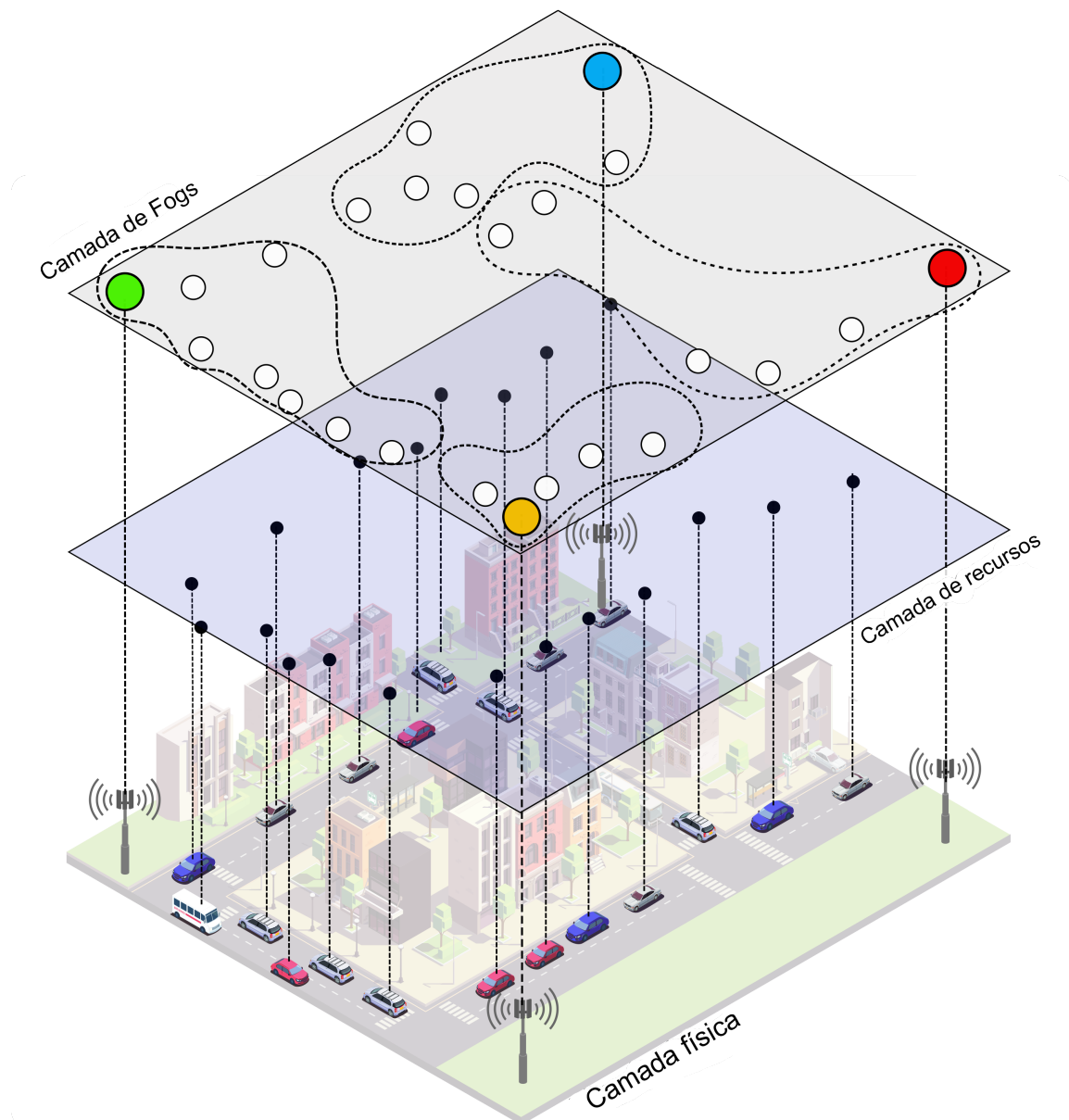


Figura 7 – Representação do VEITRA. Fonte: Pereira et al. (2021a). Adaptada.

solicitado. Devido à mobilidade do veículo, à disponibilidade de recursos ou mesmo à necessidade de balanceamento de carga, poderá haver a migração de *Fogs* para que se dê continuidade no atendimento dos veículos com a disponibilidade máxima dos recursos. Consideraremos que a comunicação já estará sendo realizada de forma eficiente, e priorizaremos o mecanismo somente na alocação e no gerenciamento de tais recursos.

Em um determinado momento, um veículo v precisa processar informações, mas seus recursos computacionais não oferecem suporte para processar tal tarefa. Para isso, ele envia uma mensagem de solicitação à infraestrutura formada para encontrar um conjunto de recursos disponíveis através do conjunto de *Fogs* que poderiam emprestar seus recursos para executar sua tarefa. Portanto, o grande desafio do VEITRA é alocar a maior

quantidade de recursos, atendendo com eficiência as demandas de cada serviço. Para isso, o VEITRA utiliza métodos matemáticos multicritério mais precisamente a estratégia de seleção elitista (SAATY, 1980; LOOTSMA, 1993), que possui um tempo de decisão curto devido à simplicidade de seus cálculos de decisão, conseqüentemente diminuindo o consenso de processamento e energia consumida.

O VEITRA visa o balanceamento de carga entre os serviços entre as diversas *Fogs*, maximizando o aproveitamento dos recursos computacionais disponibilizados pela nuvem veicular, ao mesmo tempo em que atende aos requisitos de recursos de cada serviço solicitado. Para tanto, consideramos as características de processamento, armazenamento e tempo de execução do serviço como parâmetros do método proposto, ou seja, os recursos computacionais para atender um serviço solicitado. O mecanismo de seleção será executado pelo controlador no conjunto de *Fogs*, onde selecionará qual *Fog* tem os recursos mais bem alocados para melhor atender às necessidades de um serviço. Portanto, há necessidade de troca de fluxo entre as *Fogs*, assim como é imprescindível a comunicação entre veículos.

Para dados de estado da rede, a *Fog* verifica as condições de comunicação entre os veículos e os RSUs, ou seja, a *Fog* monitora os fluxos sendo trafegados pela rede, de modo que a *Fog* pode capturar a largura de banda usada por cada fluxo de dados e verificar a largura de banda disponível para que possa ser usado para atender novos serviços. O tempo de execução do serviço é estabelecido pelo somatório dos tempos de execução dos serviços alocados na *Fog*, este somatório deve-se à concorrência dos recursos computacionais na *Fog*. A capacidade de armazenamento e processamento é dada pela alocação de serviços no *Fog*.

O VEITRA considera um conjunto de métricas para a tomada de decisão, tais como: (i) consumo de tempo de serviço, (ii) armazenamento e (iii) processamento. Especificamente, o consumo de tempo de serviço significa há quanto tempo o serviço está ativo. O armazenamento reconhece a quantidade de memória para executar o serviço. O processamento representa o tempo de processamento gasto. Esses parâmetros são fornecidos pelas *Fogs* e chamados de matrizes de decisão, uma vez que seus parâmetros são usados para decidir qual *Fog* realizará a alocação.

Inicialmente, o método selecionado para a criação da matriz de decisão foi o AHP, entretanto com o desenvolvimento da pesquisa, notamos que o método possui algumas lacunas onde deu-se a abertura para o aprimoramento do método, onde a principal estabeleceu-se na montagem da matriz, onde é utilizado o vetor Perron-Fobenius, qual estabelece o limite de números positivos à matriz, além disso, o cálculo da pontuação final das alternativas disponíveis é realizada baseado em agregação de média aritmética. Antes de iniciar

Tabela 4 – Julgamento comparativo e valor numérico para MAHP e Classic AHP (EGUTI; GOMES; BELDERRAIN, 2007; LOOTSMA, 1993)

Relação Semântica	Valor (δ_{ij})	
	MAHP	AHP
Importância muito forte de S_i versus S_j	-8	1/4
Importância moderada de S_i versus S_j	-6	1/3
Importância baixa de S_i versus S_j	-4	1/2
Indiferença de importância de S_i versus S_j	0	1
Importância baixa de S_j versus S_i	4	2
Importância moderada de S_j versus S_i	6	3
Importância muito forte de S_j versus S_i	8	4

a construção da matriz de decisão, utilizamos os métodos AHP e MAHP que calculam o fator de influência de cada parâmetro. A Tabela 4 mostra um julgamento comparativo e um valor numérico comparando o MAHP e o AHP clássico. Usamos essa tabela para construir nossa matriz de decisão, com base no valor de δ_{ij} que é i parâmetro versus j parâmetro. Neste trabalho, consideramos o processamento o parâmetro mais importante, seguido armazenamento, tempo de serviço e largura de banda. Portanto, esses valores serão usados como um peso para estabelecer a matriz de decisão, como podemos ver na equação 4.1.

O AHP e o MAHP são utilizados para auxiliar a tomada de decisão baseada em múltiplos critérios envolvidos. Os métodos calculam os graus de importância por meio da comparação de pares entre os valores de cada parâmetro para ajustar seus pesos em tempo de execução. Como resultado, quanto menor o peso, menos importante é o critério correspondente. Consideramos quatro níveis de importância para a comparação entre cada par de parâmetros (muito forte, moderada, baixa e indiferente). Este nível de importância indica a importância de um parâmetro sobre os outros.

As Tabelas 5 e 6 representam em suas linhas e colunas as comparações de pares realizadas nos métodos AHP e MAHP, respectivamente. Para construir essas matrizes, definimos os critérios mais importantes em um valor nominal, iniciando por processamento P , armazenamento A , tempo de serviço TS e por último a largura de banda LB . Após essa definição, calculamos os demais respectivos valores dependentes com base no inicial, que possui maior importância/peso. As matrizes determinam os elementos fundamentais que orientam a tomada de decisão de nosso mecanismo. Todo o processo de alocação depende diretamente dos valores estabelecidos nas matrizes, servindo de base para o cálculo dos escores de Fog e do peso dos recursos envolvidos. Este fator de influência indicará quais parâmetros têm maior prioridade sobre outros. Portanto, esses valores serão usados como

um peso para estabelecer a matriz de decisão.

Tabela 5 – Fator influência AHP.

FATOR	<i>P</i>	<i>A</i>	<i>TS</i>	<i>LB</i>
<i>P</i>	1	2	3	4
<i>A</i>	1/2	1	3	4
<i>TS</i>	1/3	1/3	1	4
<i>LB</i>	1/4	1/4	1/4	1

Tabela 6 – Fator influência MAHP.

FATOR	<i>P</i>	<i>A</i>	<i>TS</i>	<i>LB</i>
<i>P</i>	0.9242	0.9400	0.4961	0.3328
<i>A</i>	0.0577	0.0587	0.4961	0.3328
<i>TS</i>	0.0144	0.0009	0.0077	0.3328
<i>LB</i>	0.0036	0.0002	0.00003	0.0013

Equação 4.1 representa o cálculo do fator de influência, que é o valor que mede uma dada prioridade de *Fog* no recebimento de um novo recurso. Nesse sentido, dado uma *Fog*, Inf_i leva em consideração os recursos atualmente disponíveis e os multiplica por sua importância $peso_p$. Este valor é usado para saber a influência de cada *Fog* no cenário.

$$Inf_i = fat_i * peso_p \tag{4.1}$$

Equação 4.2 representa que cada serviço S_v possui um requisito diferente para o sistema, para isso utilizamos um novo peso para definir suas preferências gerenciadas pelo controlador.

$$Sv = \{Sv_1, Sv_2, Sv_3, Sv_4\} \tag{4.2}$$

$$where\ Sv_1 = 1, Sv_2 = 2, Sv_3 = 3, Sv_4 = 4 \tag{4.3}$$

A Equação 4.4 representa a combinação nos serviços S_v e o fator de influência Inf_i definindo a matriz como segue:

$$M = \begin{matrix} Sv_1 * Inf_i & Sv_2 * Inf_i & Sv_3 * Inf_i & Sv_4 * Inf_i \end{matrix} \tag{4.4}$$

$$\begin{matrix} \vdots & \vdots & \vdots & \vdots \end{matrix} \tag{4.5}$$

$$\begin{matrix} Sv_1 * Inf_n & Sv_2 * Inf_n & Sv_3 * Inf_n & Sv_4 * Inf_n \end{matrix}$$

(4.6)

onde n é o número de *Fogs*. A equação 4.7 descreve o processo após a construção da matriz, no qual realizamos uma normalização dos valores alcançados. Além disso, VEITRA captura os valores obtidos e calcula a distância euclidiana entre as *Fogs* do cenário, conforme mostrado em 4.8. Por último, representado pela Equação 4.9 o algoritmo escolhe o *Fog* com uma pontuação mais alta para alocação.

$$Matrix_{i,j} = (av_{i,j} - \overline{av}_j)/n \quad (4.7)$$

Qual \overline{d}_j é a média aritmética dos números de Fog.

$$e = \sqrt{\sum_{j=1}^n (av_{ij} - f_{ij})^2} \quad (4.8)$$

Onde $f_{i,j}$ são os números dos parâmetros das outras *Fogs*.

$$Fog = HighScore(Distance) \quad (4.9)$$

Nosso sistema está dividido em três entidades diferentes: i) controlador; ii) *Fogs* e iii) veículo. O controlador gerencia a comunicação entre *Fogs* e veículos e otimiza a alocação de recursos. O *Fog* tem a responsabilidade de receber as solicitações do controlador e alocar recursos computacionais aos veículos. Os veículos enviam mensagens ao controlador para solicitar recursos para executar serviços e também enviam mensagens sobre seu status.

O algoritmo 1 é estabelecido em uma função que verifica o tipo de mensagem recebida, se é um requisito ou não, então será criado um ranking de Fog com base no MAHP (linhas 1 a 3). Após a criação, VEITRA compara os recursos disponíveis no *Fog* selecionado com os recursos de que o veículo precisa. Então, se tiver recursos suficientes, os recursos serão alocados, o *Fog* enviará uma mensagem para ambos, e computará que atendeu (linhas 4 a 10). Caso não tenha recursos disponíveis, contará como um bloqueado e tentará em outro *Fog* até o final dos números de *Fog* (linhas 4 e 12). Porém, tentando todos as *Fogs* e não obtendo sucesso na alocação, VEITRA irá considerar aquele veículo como temporariamente negado, pedindo para ele tentar novamente mais tarde (linhas 14 e 15). Outra possibilidade da mensagem é sair do ambiente, conseqüentemente saindo do *Fog* que está alocado, o controlador recebe essa mensagem e comunica o *Fog* para retirar os recursos deste veículo (linhas 17 a 19).

O algoritmo 2 representa a entidade *Fog* em nosso sistema. A função exclusiva recebe uma mensagem do controlador e verifica se é uma mensagem de atendimento ou de saída.

Algorithm 1: F-controller

```

Input:  $f$ : Lista de  $fogs$ .
Function recebe (msg) is
    if msg = req (serviço) then
        rank_de_fogs  $\leftarrow$  criando_fogs_ranking (msg.v)
        while rank_de_fogs  $\neq$   $\emptyset$  do
             $i \leftarrow$  rank_de_fogs.pop_front()
            if  $f[i].r + \text{msg.v}_r \leq f[i].\text{cap}$  then
                 $f[\text{msg.c.fog}] \leftarrow f[\text{msg.c.fog}] + \text{msg.v}_r$ 
                envia (msg.vid,  $i$ )
                envia ( $f[i]$ , msg.vid)
                return atendidos++
            end
            else return bloqueados++
        end
        negados++
        envia (msg.vid,  $\emptyset$ , status = try_again_later)
    end
    else if msg = req(saída) then
         $f[\text{msg.c.fog}] \leftarrow f[\text{msg.c.fog}] - \text{msg.v}_r$ 
        envia(msg.c.fog, msg.vid, status = saindo)
    end
end

```

Algorithm 2: Fog

```

Input:  $f_i$ : Fog atual.
            $f_{i_r}$ : recursos de  $f_i$ .
Function recebe (msg) is
    // atualiza lista de recursos
    if msg = atendido then
         $f_{i_r} \leftarrow f_{i_r} + \text{msg.v}_r$ 
    end
    else
         $f_{i_r} \leftarrow f_{i_r} - \text{msg.v}_r$ 
    end
end
end

```

Neste caso, se for uma mensagem para alocar, o *Fog* irá receber os recursos e agregar para si. No entanto, se for uma mensagem de saída, o *Fog* irá remover os recursos de seu *pool* (linhas 1 a 4).

O algoritmo 3 mostra o lado do veículo de VEITRA. É dividido em três funções: i) recebe; ii) requisita e iii) requisita_saída. O primeiro verifica se a mensagem é um atendimento, então comunicará ao veículo qual *Fog* está alocado agora, caso não possa ser alocado, esperará um tempo t e solicitará novamente (linhas 1 a 5). A solicitação de função tem uma obrigação: enviar uma mensagem ao controlador quando o veículo precisar ser alocado (linhas 7 a 9). Caso contrário, a função requisita_saída envia uma

mensagem ao controlador quando o veículo precisa deixar o *Fog* (linhas 10 a 12).

As Figuras 8, 9 e 10 representam uma outra forma de visualização da entidades em nosso sistema, exibindo fluxogramas dos processos de cada uma, como explicados nos algoritmos anteriores.

4.2 Simulação

Esta seção descreve as ferramentas de simulação, parâmetros, algoritmos e cenários usados na avaliação da política de alocação de recursos proposta. Conduzimos uma série de análises simuladas comparando o desempenho de VEITRA com duas soluções de literatura e uma política de alocação de recursos.

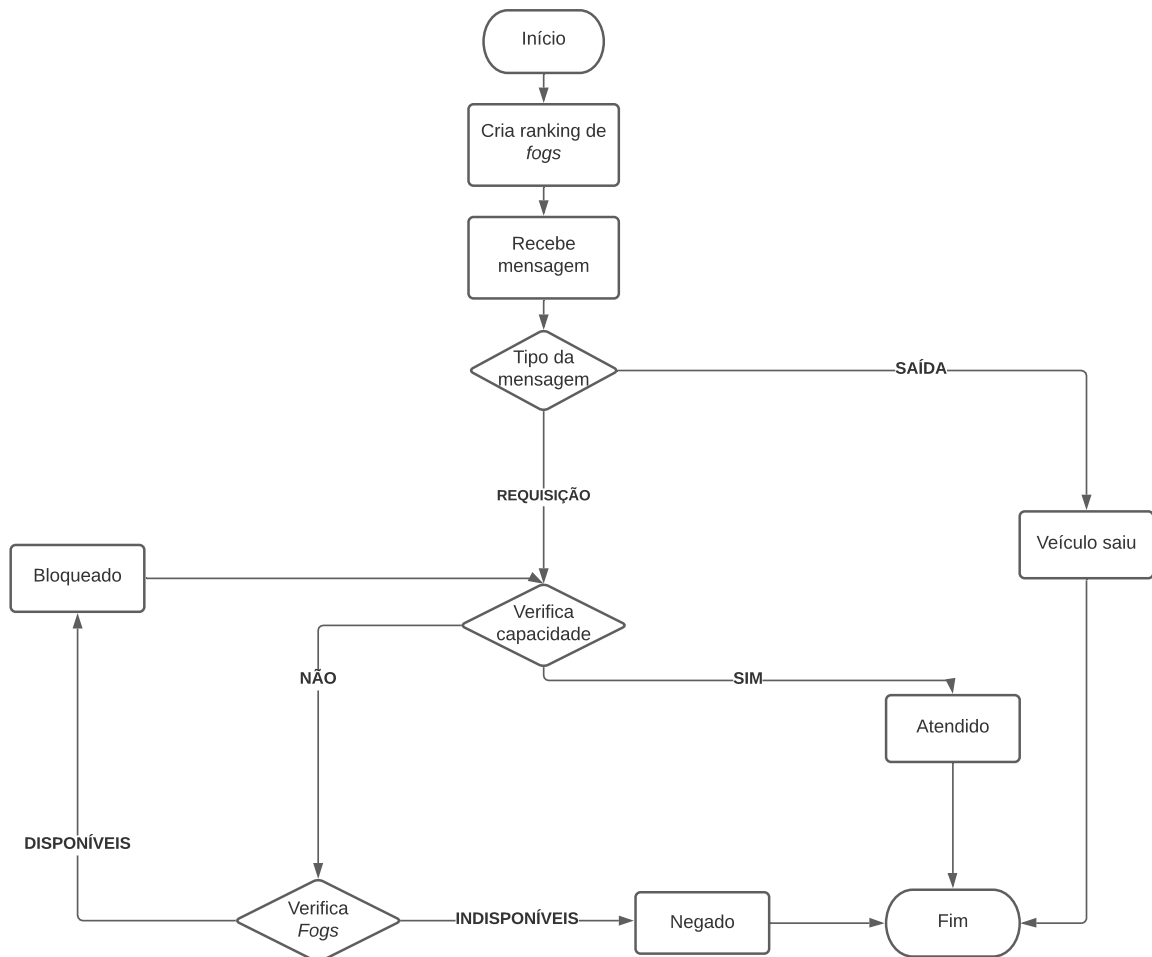


Figura 8 – Fluxograma do funcionamento da entidade Controlador. Fonte: Própria.

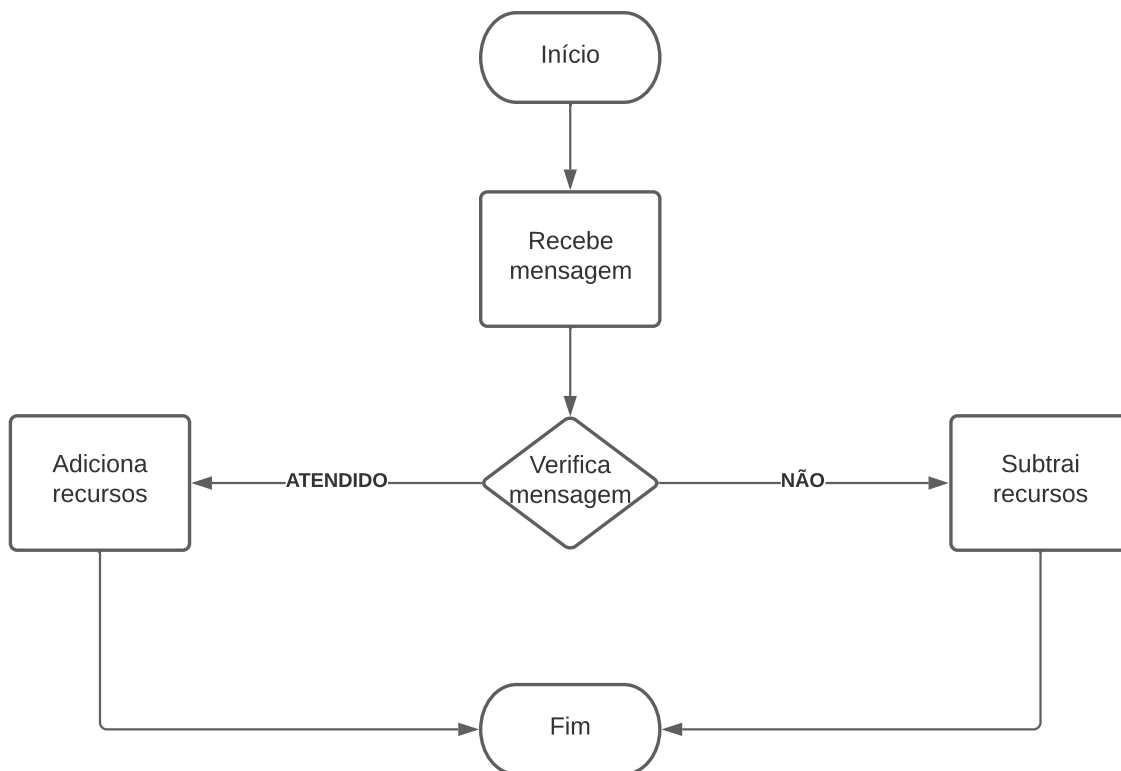


Figura 9 – Fluxograma do funcionamento da entidade *Fog*. Fonte: Própria.

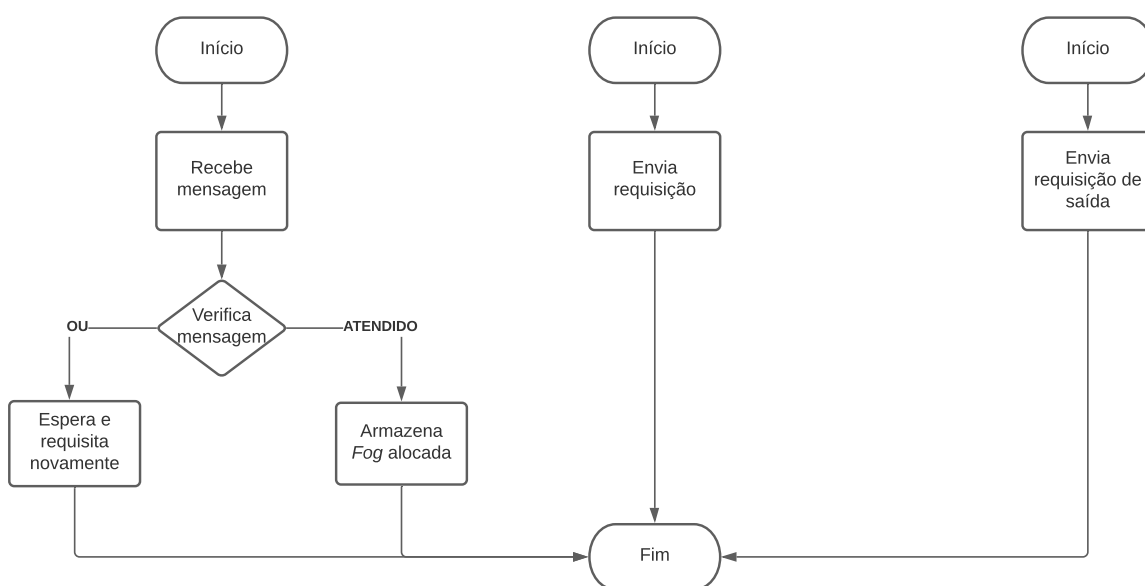


Figura 10 – Fluxograma do funcionamento da entidade Veículo. Fonte: Própria.

Algorithm 3: Veículo

Input: c_fog : Fog atual do veículo.
Function recebe (msg) **is**
 if $msg = \text{atendido}$ **then**
 | $c_fog \leftarrow msg.c_fog$
 end
 else
 | espera(t)
 | requisita(v_{id}, v_r)
 end
end
Function requisita (v_{data}) **is**
 | envia ($f_control, v_{data}, status = req(\text{serviço})$)
end
Function requisita_saída (v_{data}) **is**
 | envia ($f_control, v_{data}, status = req(\text{saída})$)
end

4.2.1 Cenário

Realizamos nossas simulações considerando as densidades de veículos de 3382, 5720, 8059, 10345, 12631 veículos obtidas a partir do traçado de mobilidade realista de TAPASCologne, mapa qual é exibido na Figura 11. Nesse cenário, RSU e veículos realizam a alocação de recursos, sendo capazes de processar, analisar, comunicar e tomar decisões distribuídas e colaborativas. Cada veículo recebe um valor aleatório para cada um de seus recursos, considerando o intervalo $Sv_* \in [1, 10]$. A alocação de recursos é executada em um ambiente Quatro-Fogs construído em quatro RSUs, onde cada Fog f_* pode atender simultaneamente até 100 unidades de cada recurso $f_{Sv_*} \in [0, 100]$. Cada f_* tem total cobertura de comunicação da área avaliada, podendo atender a qualquer solicitação durante a simulação. Cada uma das densidades avaliadas tem um tempo de simulação diferente: 10000 segundos (3382 veículos), 11600 segundos (8059 veículos), 13200 segundos (10345 veículos), 14800 segundos (10345 veículos) e 16400 segundos (12631 veículos). Mostramos todos os parâmetros e valores na Tabela 7. Todos os algoritmos consideram um intervalo de confiança de 95% obtido a partir de 33 Seeds simuladas 33 vezes.

4.2.2 Métricas de avaliação

Avaliamos nossas simulações considerando as seguintes métricas:

- **Serviços atendidos:** é uma métrica que representa o número de solicitações de serviço atendidas. Um elevado número de serviços atendidos indica que a política

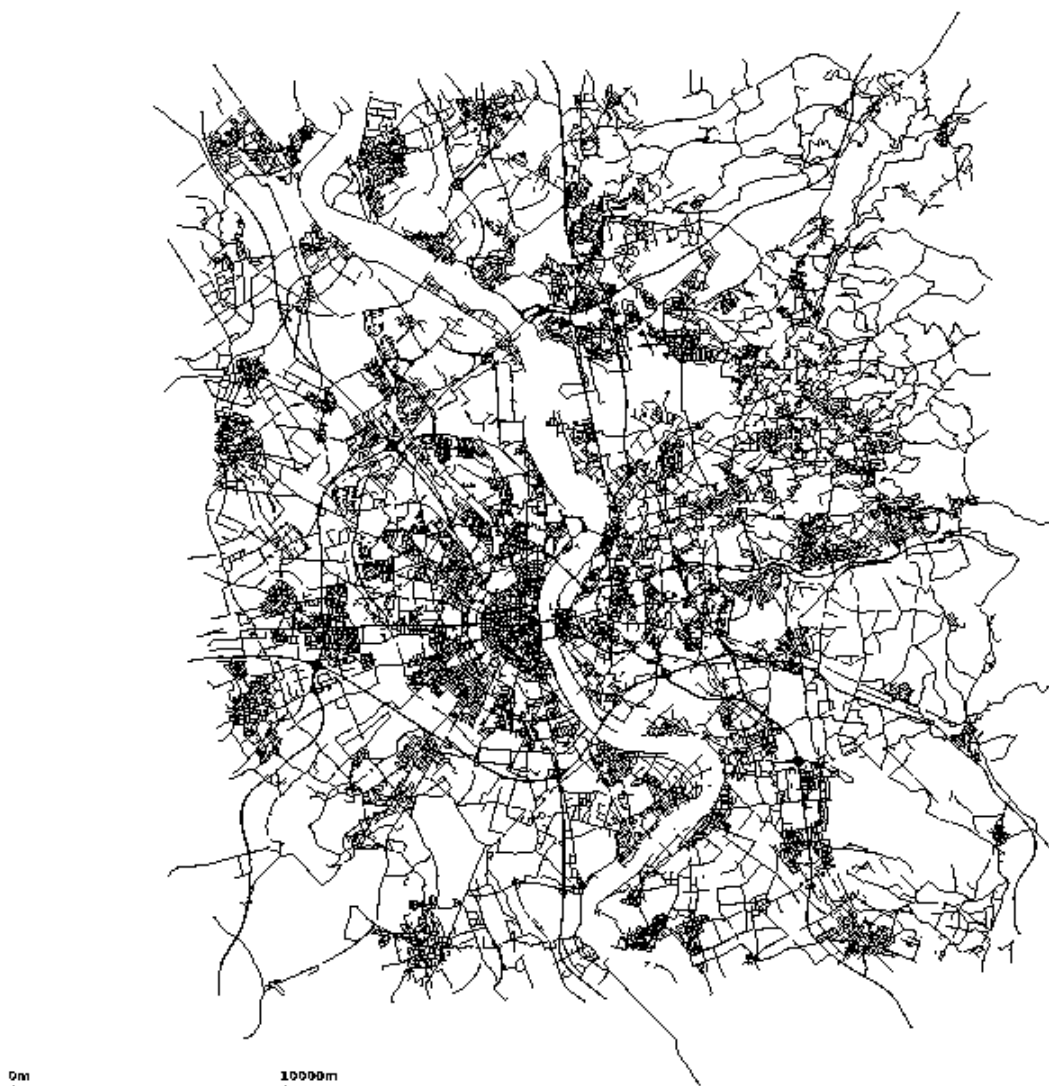


Figura 11 – Mapa de Colônia. Fonte: Simulation of Urban MObility.

de alocação é eficiente para otimizar a alocação de recursos considerando o intervalo avaliado.

- **Serviços bloqueados:** é uma métrica que calcula o número de vezes que um Fog recusa uma solicitação de serviço. Tal cenário ocorre quando um novo serviço v_r ultrapassa a capacidade máxima de um determinado Fog (f_i). Observe que v_r pode ser bloqueado por mais de um Fog.
- **Serviços negados:** é uma métrica que calcula quando um v_r ultrapassa a capacidade máxima de todos as Fogs disponíveis (f), sendo negado para alocar seus recursos. Um alto número de serviços negados indica que o algoritmo/política não pode gerenciar um grande número de solicitações em um curto período de tempo.

Tabela 7 – Parâmetros de simulação

Parâmetros	Valores
Veículos	3382, 5720, 8059, 10345, 12631
RSUs	4
Seeds	33
Repetições	33
Tempo de simulação (segundos)	10000, 11600, 13200, 14800, 16400
Comportamento de mobilidade veicular	TAPASCologne
Área do mapa de simulação	405 km ²
Linguagem de programação utilizada	Python
Intervalo de confiança	95%
Status	Atendidos, bloqueados e negados

4.3 Resultados

As Figuras 12, 13, 14 mostram os gráficos com os resultados das simulações realizadas, onde dividiremos o VEITRA em VEITRA-AHP (baseado no método AHP) e VEITRA-MAHP (baseado no aprimoramento do AHP, o MAHP). As comparações foram realizadas com outros dois métodos, o método meta-heurístico NAUTILUS (QUES-

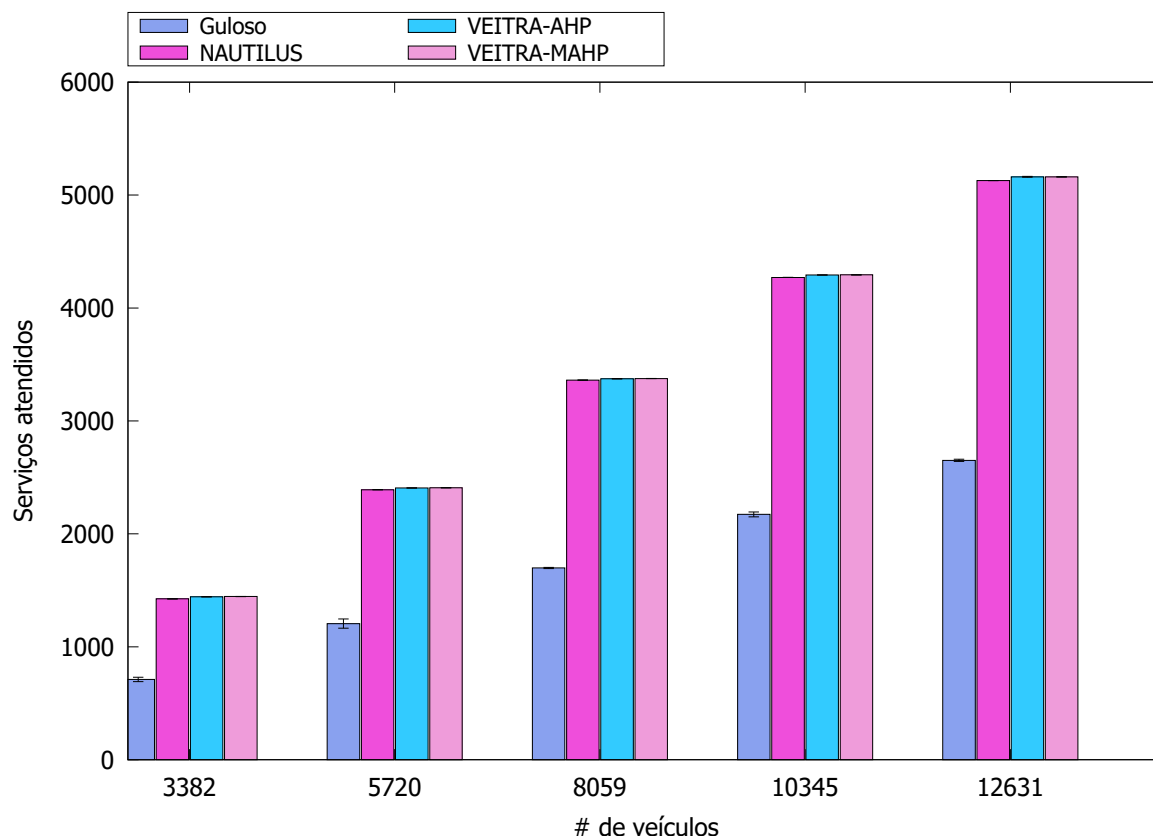


Figura 12 – Análise comparativa de desempenho dos serviços atendidos. Fonte: Própria

SADA et al., 2021) e o método Guloso (YANG et al., 2017).

A Figura 12 exibe os resultados de serviços atendidos nas simulações. Considerando 3382 veículos, o mecanismo proposto baseado no método AHP, apresentou 1445 serviços atendidos, sendo 0,13% menor do seu aprimoramento baseado no método MAHP, 1,31% maior que o mecanismo NAUTILUS, e 50.86% maior que o método Guloso, quais apresentaram 1445, 1424 e 710 serviços atendidos, respectivamente. Considerando 5720 veículos, o VEITRA-AHP apresentou resultados de 2406 atendimentos aos serviços solicitados, enquanto sua evolução apresentou 2408 serviços atendidos, já o mecanismo meta-heurístico apresentou 2390 atendimentos aos serviços requisitados, por último o método Guloso apresentou 1205 serviços atendidos, sendo assim, o VEITRA-AHP alcançou 0.08% menos serviços atendidos quando comparado a sua evolução, 0.66% a mais de serviços atendidos quando comparado ao mecanismo NAUTILUS e 49.91% mais serviços atendidos em comparação ao método Guloso. Já no cenário com 8059 veículos, o VEITRA-AHP alcançou 3373 serviços atendidos, sendo 0.05% menor em relação ao VEITRA-MAHP, qual alcançou 3375 serviços atendidos, além disso, apresentou um resultado 0.35% maior quando comparado ao mecanismo NAUTILUS, quando comparado ao método Guloso, nosso mecanismo apresentou resultados 49.65% maiores, já que o método comparado atingiu 1698 serviços atendidos. Considerando 10345 veículos, tanto VEITRA-AHP, quanto VEITRA-MAHP alcançaram o mesmo número de serviços atendidos, qual foi 4293, apenas diferindo do método meta-heurístico que atendeu 4270 serviços, alcançando um resultado 0.53% maior e do método Guloso, qual atendeu 2173 serviços, sendo um número 49.38% menor em relação a nossa proposta. O último cenário com 12631 exibe mais uma vez uma equidade entre os métodos "irmãos", onde ambos atingiram 5161 serviços atendidos, sendo um número 0.63% maior quando comparado ao método NAUTILUS, que atingiu a marca de 5128 serviços atendidos e 48.63% maior que o método Guloso, qual atingiu 2651 serviços atendidos. Com isso, é possível observar que o estado de espera e a possibilidade de uma nova solicitação aumentam o número de serviços atendidos do mecanismo proposto. Isso acontece porque enquanto os outros algoritmos descartam os veículos, nossa política abre outra chance de alocação, mostrando que VEITRA tem uma escalabilidade maior do que os algoritmos comparados.

A Figura 13 exibe os serviços negados das simulações performadas. Considerando 3382 veículos, o mecanismo proposto baseado no método AHP, apresentou 215 serviços negados, sendo 9.30% maior do seu aprimoramento baseado no método MAHP, 3.72% menor que o mecanismo NAUTILUS, e 1240% menor que o método Guloso, quais apresentaram 195, 223 e 2881 serviços negados, respectivamente. Considerando 5720 veículos, o VEITRA-AHP apresentou resultados de 409 negações aos serviços solicitados, enquanto

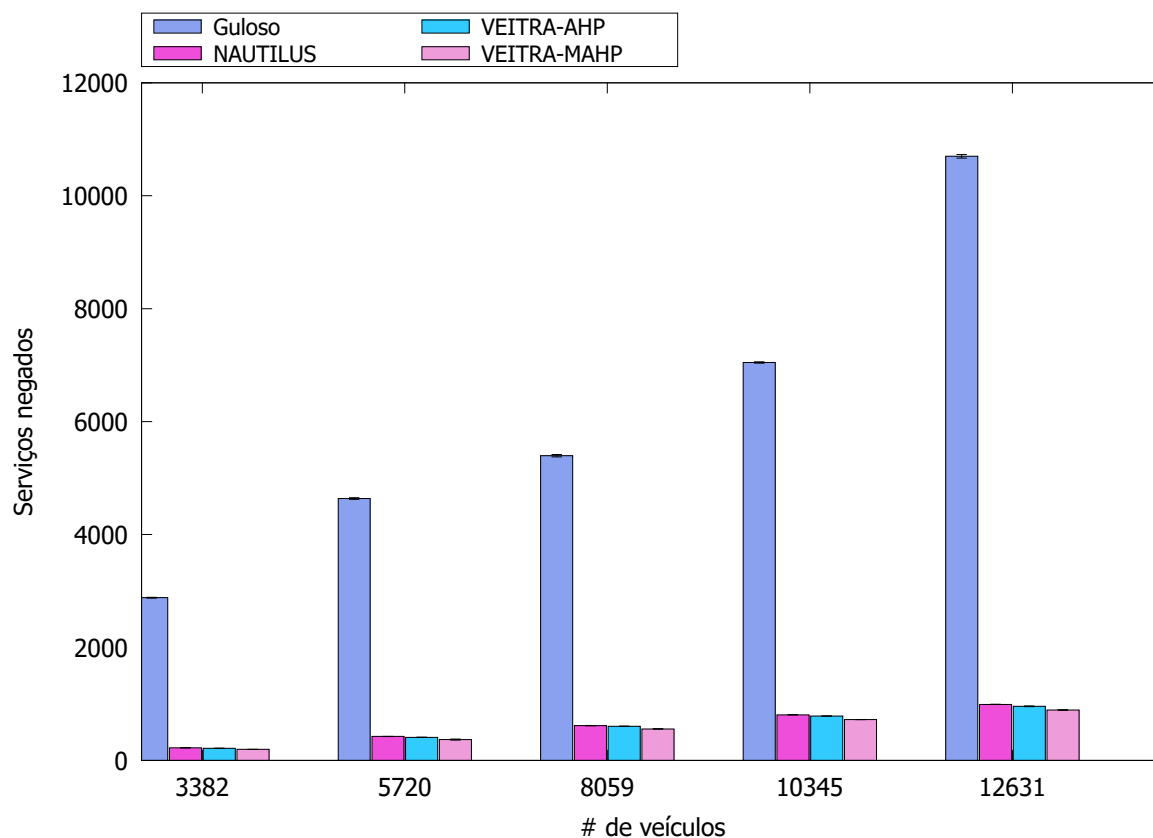


Figura 13 – Análise comparativa de desempenho de serviços negados. Fonte: Própria.

sua evolução apresentou 369 serviços negados, já o mecanismo meta-heurístico apresentou 425 negações aos serviços requisitados, por último o método Guloso apresentou 4638 serviços negados, sendo assim, o VEITRA-AHP alcançou 9.77% mais serviços negados quando comparado a sua evolução, 3.91% a menos de serviços negados quando comparado ao mecanismo NAUTILUS e 1033% menos serviços negados em comparação ao método Guloso. Considerando uma densidade de 8059 veículos, VEITRA-AHP negou 604 serviços, atingindo um número 7.78% maior que sua evolução e 1.82% menor que o método NAUTILUS, quais atingiram 557 e 615 negações, respectivamente. Além disso, VEITRA-AHP continua negando menos que o método Guloso, qual negou 5396 serviços, atingindo um número 793% menor. Considerando um cenário com 10345 veículos, mais uma vez o VEITRA-MAHP atinge a marca de menor quantidade de negações com 723 negados, sendo 7.78% menor que o VEITRA-AHP que atingiu a marca de 784 negações. Quando comparamos o VEITRA-AHP com os métodos Guloso e NAUTILUS, quais atingiram valores de 807 e 7048 negações, sendo 2.93% e 798% menor, respectivamente. No último cenário, com a densidade de 12631 veículos, o VEITRA-AHP negou 957 serviços, atingindo uma marca 6.79% maior que o VEITRA-MAHP, qual negou 892 serviços, já na comparação com o método meta-heurístico, o mecanismo aqui proposto alcançou um

número 3.55% menor no número de negações já que NAUTILUS negou 991 serviços. Por último, o método Guloso alcançou um número de negações 1014% maior que o de nosso mecanismo proposto, realizando a negação de 10669 serviços. Os resultados dos serviços negados mostraram que VEITRA-MAHP teve desempenho com menos negados do que os outros em todas as comparações. Isso ocorre quando o sistema abre uma nova possibilidade para um determinado veículo já negado de tentar novamente a alocação. O resultado é uma diminuição no número de serviços negados.

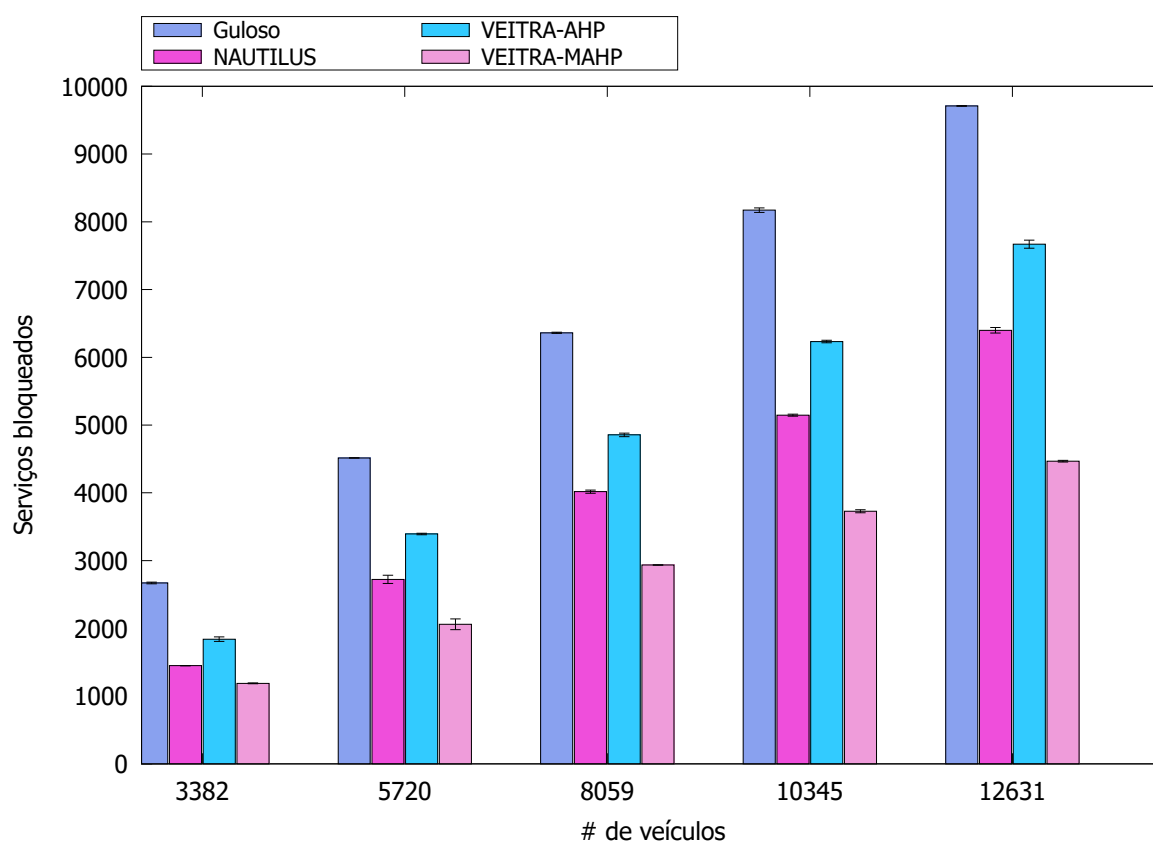


Figura 14 – Análise comparativa de desempenho de serviços bloqueados. Fonte: Própria.

A Figura 14 apresenta os serviços bloqueados das simulações performadas. Considerando o primeiro cenário com uma densidade de 3382 veículos, VEITRA-AHP alcançou 1840 serviços bloqueados, já VEITRA-MAHP atingiu 1188 bloqueios, enquanto NAUTILUS atingiu 1449 e Guloso 2671, sendo 21.96% e 45.16% menor quando comparados a NAUTILUS e Guloso, respectivamente, apenas bloqueando mais serviços quando comparado com sua evolução, onde a porcentagem foi 35% maior. No segundo cenário com 5720 veículos, VEITRA-AHP bloqueou 3392 serviços, 39.26% a mais que sua evolução, qual bloqueou 2060 serviços e 19.72% a mais que o método meta-heurístico NAUTILUS, qual bloqueou 2723 serviços, entretanto continuou bloqueando menos que o Guloso, qual atingiu 4515 serviços bloqueados, sendo a porcentagem de bloqueio 33.10%

menor. Considerando uma densidade de 8059 veículos, VEITRA-AHP bloqueou 4856 serviços, atingindo um número 39.55% maior que sua evolução e 17.25% maior que o método NAUTILUS, quais atingiram 2935 e 4018 bloqueios, respectivamente. Por outro lado, VEITRA-AHP continua bloqueando menos que o método Guloso, qual bloqueou 6361 serviços, atingindo um número 30.99% menor. Considerando um cenário com 10345 veículos, mais uma vez o VEITRA-MAHP atinge a marca de menor quantidade de bloqueios com 3728 serviços bloqueados, sendo 40.18% menor que o VEITRA-AHP que atingiu a marca de 6233 bloqueios. Quando comparamos o VEITRA-AHP com os métodos Guloso e NAUTILUS, quais atingiram valores de 8172 e 5146 bloqueios, sendo 31.10% maior e 17.43% menor, respectivamente. No último cenário, com a densidade de 12631 veículos, o VEITRA-AHP bloqueou 7669 serviços, atingindo uma marca 41.73% maior que o VEITRA-MAHP, qual bloqueou 4668 serviços, já na comparação com o método meta-heurístico, o mecanismo aqui proposto alcançou um número 16.56% maior no número de serviços bloqueados já que NAUTILUS bloqueou 6398 serviços. Por último, o método Guloso alcançou um número de bloqueados 26.61% maior que o de nossa proposta, bloqueando 9710 serviços. VEITRA e sua evolução apresentaram menos serviços bloqueados que os outros devido à sua escalabilidade.

4.4 Considerações parciais

Em resumo, nossa política alcançou resultados significativos em todas as métricas de comparação. VEITRA consegue atender mais serviços e, ao mesmo tempo, reduz o número de serviços negados e bloqueados quando comparado a todos os algoritmos da literatura avaliados. Portanto, o objetivo principal deste trabalho é otimizar a alocação de recursos para atender o máximo possível os veículos. A importância de otimizar os serviços atendidos reflete em diversos processos *Cloud* veicular, por exemplo, recomendar alguma mudança de rota para diminuir o tempo de viagem para nv número de veículos, também pode mostrar informações sobre um determinado posto de gasolina nas estradas. Portanto, otimizando o número de serviços atendidos infere-se que os serviços negados irão diminuir.

5 CONCLUSÃO

Neste capítulo serão apresentadas as considerações finais do trabalho, além das principais contribuições realizadas, também como as produções científicas que o trabalho desenvolvido possui impacto direto na criação, e finalmente ideias para trabalhos futuros.

5.1 Considerações finais

Propusemos VEITRA, um mecanismo de alocação de recursos baseada em paradigmas de *Fog* projetada com base em um método hierárquico que aborda os desafios em um VC para gerenciar seus recursos. O mecanismo permite construir um ambiente com um controlador que executa os algoritmos para comunicar essa decisão a *Fogs* e veículos. Os veículos têm recursos computacionais disponíveis para compartilhar dentro da rede, e *Fogs* podem processar informações e alocar os recursos para os veículos executarem os serviços. Formulamos VEITRA com base em um método hierárquico, no qual executa as decisões para otimizar a tomada de decisão de alocação de recursos, indicando a viabilidade de uma alocação de recursos, ao decorrer do desenvolvimento do trabalho, uma melhoria em seu núcleo de tomada de decisão foi levada em consideração, utilizando-se de uma evolução do método base, fazendo com que essa melhoria atingisse resultados melhores do que a proposta inicial. O principal objetivo deste projeto de mestrado é o desenvolvimento de um mecanismo de alocação de recursos que é maximizar o número de serviços atendidos. Uma das técnicas utilizadas durante a simulação foi de criarmos uma fila pendente no sistema para realizar a alocação durante a simulação para atingir esse objetivo. A avaliação mostra que VEITRA-AHP possui mais serviços atendidos, menos serviços bloqueados e negados do que o método Guloso, já em comparação com o método meta-heurístico NAUTILUS, VEITRA-AHP atende mais serviços em todas as comparações, entretanto, possui um número maior de negações e de bloqueios. Por último, quando comparado à evolução VEITRA-MAHP, é notável que seu sucessor desempenha números mais significativos, superando-o em todas as comparações. Portanto, é possível constatar a escalabilidade e alta adaptação do método proposto nesta dissertação de mestrado, sendo possível aliar outros métodos de tomadas de decisão.

5.2 Principais contribuições

Essa seção aborda as principais contribuições possíveis durante o desenvolvimento deste projeto de mestrado, sendo elas:

- Proposta de utilização um método analítico hierárquico para auxílio na tomada de decisões na alocação de recursos baseado em paradigmas de *Fog*, abusando de infraestruturas físicas nas vias que ajudam no processo de alocação.
- Desenvolvimento de um mecanismo e de sua evolução utilizando a linguagem Python, constatando a escalabilidade do método de decisão selecionado.
- Montagem e implementação de ambiente de simulação aplicável ao mecanismo proposto.
- Otimização dos recursos alocados para a execução dos serviços requisitados, consequentemente, a diminuição dos serviços negados.

5.3 Produções científicas

Essa seção aborda as produções científicas publicadas durante o desenvolvimento dessa dissertação de mestrado, sendo elas:

- Pereira et al. (2021a) **Pereira, Rickson**; Gomides, Thiago; Sanches Quessada, Matheus; Lieira, Douglas; Meneguette, Rodolfo; Guidoni, Daniel; Nakamura, Luis H. V.; De Grande, Robson. (2021). Fog-oriented Hierarchical Resource Allocation Policy in Vehicular Clouds.
- Pereira et al. (2021b) **Pereira, Rickson**; Boukerche, Azzedine; Colombo da Silva, Marco; Nakamura, Luis H. V.; Freitas, Heitor; Filho, Geraldo P.; Meneguette, Rodolfo. (2021). FORESAM—FOG Paradigm-Based Resource Allocation Mechanism for Vehicular Clouds. *Sensors*. 21. 5028. 10.3390/s21115028.
- Pereira et al. (2020) **Pereira, Rickson**; Lieira, Douglas; Colombo da Silva, Marco; Pimenta, Adinovam; da Costa, Joahannes B. D.; Rosário, Denis; Villas, Leandro; Meneguette, Rodolfo. (2020). RELIABLE: Resource Allocation Mechanism for 5G Network using Mobile Edge Computing. *Sensors*. 20. 5449. 10.3390/s20195449.
- Pereira et al. (2019) **Pereira, Rickson**; Lieira, Douglas; Colombo da Silva, Marco; Pimenta, Adinovam; da Costa, Joahannes B. D.; Rosário, Denis; Meneguette, Ro-

dolfo. (2019). A novel fog-based resource allocation policy for vehicular clouds in the highway environment. 10.1109/LATINCOM48065.2019.8937912.

5.4 Trabalhos futuros

Esta dissertação de mestrado foi desenvolvida com o objetivo de alcançar a otimização da utilização dos recursos em um ambiente de *Cloud* veicular utilizando paradigmas de *Fog* e prover a maior quantidade de serviços requisitados pelos veículos. Entretanto, muitos desafios causados pelas redes veiculares ainda estão abertos. Portanto, amplia-se a necessidade de estender esta dissertação de mestrado.

Um dos maiores desafios é a implementação de nosso mecanismo em um cenário considerando variáveis de rede e comunicação no simulador VEINS, para realização de simulações onde essas variáveis impactam diretamente no desempenho da alocação de recursos. Por exemplo, cada *Fog* conhece a quantidade de recursos das outras *Fogs* usando um *beacon*, criando uma lista e compartilhando esta informação entre elas também como aliando o cenário realista e aumentando sua escala para realizar o algoritmo e avaliar seu comportamento. Além disso, pretendemos investigar métodos de estimação, como a regressão linear, para construir e ajustar a matriz de decisão de forma dinâmica, também como, métodos de decisão para que a implementação conjunta possa ser realizada, como por exemplo, a utilização de redes neurais.

REFERÊNCIAS

- ARKIAN, H. R. et al. A cluster-based vehicular cloud architecture with learning-based resource management. *The Journal of Supercomputing*, v. 71, n. 4, p. 1401–1426, Apr 2015. ISSN 1573-0484.
- ARKIAN, H. R. et al. A cluster-based vehicular cloud architecture with learning-based resource management. *The Journal of Supercomputing*, v. 71, n. 4, p. 1401–1426, 2015. ISSN 1573-0484.
- BAYKAL-GÜRISOY, M. *Semi-Markov Decision Processes*. [S.l.]: Wiley encyclopedia of operations research and management science, 2011. ISBN 9780470400531.
- BHUVANESWARI, M.; PARAMASIVAN, B.; KANDASAMY, A. Stochastic dynamic programming model for optimal resource allocation in vehicular ad hoc networks. *Sādhanā*, Springer Science and Business Media LLC, v. 43, n. 4, abr. 2018. Disponível em: <https://doi.org/10.1007/s12046-018-0853-7>.
- BITAM, S.; MELLOUK, A.; ZEADALLY, S. Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks. *IEEE Wireless Communications*, v. 22, n. 1, p. 96–102, February 2015.
- DAI, P. et al. Vehicle assisted data update for temporal information service in vehicular networks. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. [S.l.: s.n.], 2015. p. 2545–2550. ISSN 2153-0009.
- DANQUAH, W. M.; ALTILAR, D. T. Vehicular cloud resource management, issues and challenges: A survey. *IEEE Access*, v. 8, p. 180587–180607, 2020.
- DENATRAN, D. N. de T. *Estatísticas - Frota de Veículos*. 2019. Disponível em: <https://infraestrutura.gov.br/component/content/article/115-portal-denatran/8552-estat> \unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{\OT1\i\global\mathchardef\accent@spacefactor\spacefactor}\let\begin\group\end\group\relax\let\ignorespaces\relax\accent19\OT1\i\egroup\spacefactor\accent@spacefactorsticas-frota-de-ve\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{\OT1\i\global\mathchardef\accent@spacefactor\spacefactor}\let\begin\group\end\group\relax\let\ignorespaces\relax\accent19\OT1\i\egroup\spacefactor\accent@spacefactorculos-denatran.html), Acessado em: 07-10-2019.
- DINH, H. T. et al. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, v. 13, n. 18, p. 1587–1611, 2013.
- EGUTI, C.; GOMES, J.; BELDERRAIN, M. N. Application of method multiplicative ahp in the machine tool choice-case study. In: . [S.l.: s.n.], 2007.
- GAI, K. et al. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*, Elsevier, v. 59, p. 46–54, 2016.

- GERLA, M. Vehicular cloud computing. In: *Proceedings of the 11th Annual Mediterranean Ad Hoc Networking Workshop*. [S.l.: s.n.], 2012. p. 152–155.
- GOMIDES, T. S. et al. An adaptive and distributed traffic management system using vehicular ad-hoc networks. *Computer Communications*, v. 159, p. 317 – 330, 2020. ISSN 0140-3664. Disponível em: [⟨https://doi.org/10.1016/j.comcom.2020.05.027⟩](https://doi.org/10.1016/j.comcom.2020.05.027).
- HAWKING, S. In: _____. *A Brief History of Time*. [S.l.]: Bantam Dell Publishing Group, 1998. ISBN 978-0-553-38016-3.
- HEPPNER, F.; GRENANDER, U. A stochastic nonlinear model for coordinate bird flocks. In: _____. [S.l.: s.n.], 1990.
- JAISWAL, N. K. The analytic hierarchy process. In: _____. *Military Operations Research: Quantitative Decision Making*. Boston, MA: Springer US, 1997. p. 209–232. ISBN 978-1-4615-6275-7. Disponível em: [⟨https://doi.org/10.1007/978-1-4615-6275-7_8⟩](https://doi.org/10.1007/978-1-4615-6275-7_8).
- KIM, W. Cloud computing: Today and tomorrow. *Journal of Object Technology*, v. 8, p. 65–72, 01 2009.
- LIEIRA, D. D. et al. TOVEC: Task optimization mechanism for vehicular clouds using meta-heuristic technique. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2021. Disponível em: [⟨https://doi.org/10.1109/iwcmc51323.2021.9498784⟩](https://doi.org/10.1109/iwcmc51323.2021.9498784).
- LIN, F. et al. Optimization-oriented resource allocation management for vehicular fog computing. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 6, p. 69294–69303, 2018. Disponível em: [⟨https://doi.org/10.1109/access.2018.2879988⟩](https://doi.org/10.1109/access.2018.2879988).
- LIU, J. et al. Secure intelligent traffic light control using fog computing. *Future Generation Computer Systems*, v. 78, p. 817 – 824, 2018. ISSN 0167-739X.
- LIU, L.; FAN, Q. Resource allocation optimization based on mixed integer linear programming in the multi-cloudlet environment. *IEEE Access*, v. 6, p. 24533–24542, 2018.
- LOOTSMA, F. A. Scale sensitivity in the multiplicative AHP and SMART. *Journal of Multi-Criteria Decision Analysis*, Wiley, v. 2, n. 2, p. 87–110, ago. 1993. Disponível em: [⟨https://doi.org/10.1002/mcda.4020020205⟩](https://doi.org/10.1002/mcda.4020020205).
- MELL, P. M.; GRANCE, T. *SP 800-145. The NIST Definition of Cloud Computing*. Gaithersburg, MD, United States, 2011.
- MENEGUETTE, R.; BOUKERCHE, A.; De Grande, R. SMART: an efficient resource search and management scheme for vehicular cloud-connected system. In: *2Proceedings of the IEEE Global Communications Conference: Mobile and Wireless Networks*. Washington, USA: [s.n.], 2016.
- MENEGUETTE, R. et al. A resource allocation scheme based on Semi-Markov decision process for dynamic vehicular clouds. In: *Proceedings of the IEEE ICC 2017 Mobile and Wireless Networking*. Paris, France: [s.n.], 2017.
- MENEGUETTE, R. et al. A solution for detection and control for congested roads using vehicular networks. *IEEE Latin America Transactions*, v. 14, n. 4, p. 1849–1855, 2016.

- MENEGUETTE, R. I.; BOUKERCHE, A. Peer-to-peer protocol for allocated resources in vehicular cloud based on v2v communication. In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. [S.l.: s.n.], 2017. p. 1–6. ISSN 1558-2612.
- MENEGUETTE, R. I.; BOUKERCHE, A.; GRANDE, R. de. Smart: An efficient resource search and management scheme for vehicular cloud-connected system. In: *2016 IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2016. p. 1–8.
- MENEGUETTE, R. I.; BOUKERCHE, A.; PIMENTA, A. H. M. Avarac: An availability-based resource allocation scheme for vehicular cloud. *IEEE Transactions on Intelligent Transportation Systems*, v. 20, n. 10, p. 3688–3699, Oct 2019.
- MENEGUETTE, R. I.; BOUKERCHE, A.; PIMENTA, A. H. M. Avarac: An availability-based resource allocation scheme for vehicular cloud. *IEEE Transactions on Intelligent Transportation Systems*, v. 20, n. 10, p. 3688–3699, 2019.
- MENEGUETTE, R. I.; GRANDE, R. D.; LOUREIRO, A. A. *Intelligent Transport System in Smart Cities*. [S.l.]: Springer, 2018. 182 p.
- MENEGUETTE, R. I.; GRANDE, R. E. D.; LOUREIRO, A. A. F. Vehicular cloud. In: _____. *Intelligent Transport System in Smart Cities: Aspects and Challenges of Vehicular Networks and Cloud*. Cham: Springer International Publishing, 2018. p. 113–145. ISBN 978-3-319-93332-0. Disponível em: https://doi.org/10.1007/978-3-319-93332-0_6.
- MENG, H. et al. A utility-based resource allocation scheme in cloud-assisted vehicular network architecture. In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. [S.l.: s.n.], 2015. p. 1833–1838. ISSN 2164-7038.
- MENG, Y. et al. A low-cost resource re-allocation scheme for increasing the number of guaranteed services in resource-limited vehicular networks. *Sensors*, MDPI AG, v. 18, n. 11, p. 3846, nov. 2018. Disponível em: <https://doi.org/10.3390/s18113846>.
- MIAO, Z. et al. On resource management in vehicular ad hoc networks: A fuzzy optimization scheme. In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. [S.l.: s.n.], 2016. p. 1–5.
- MITCHELL, K. H.; WASIL, E. A. Ahp in practice: Applications and observations from a management consulting perspective. In: _____. *The Analytic Hierarchy Process: Applications and Studies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989. p. 192–212. ISBN 978-3-642-50244-6. Disponível em: https://doi.org/10.1007/978-3-642-50244-6_13.
- MOSTAED, M.; ALDABAS, K.; OLAVERRI-MONREAL, C. V2v communication system to increase driver awareness of emergency vehicles. In: MORENO-DÍAZ, R.; PICHLER, F.; QUESADA-ARENCIBIA, A. (Ed.). *Computer Aided Systems Theory – EUROCAST 2017*. Cham: Springer International Publishing, 2018. p. 413–418. ISBN 978-3-319-74727-9.
- NING, Z.; HUANG, J.; WANG, X. Vehicular fog computing: Enabling real-time traffic management for smart cities. *IEEE Wireless Communications*, v. 26, n. 1, p. 87–93, February 2019.

- OLARIU, S.; HRISTOV, T.; YAN, G. The next paradigm shift: From vehicular networks to vehicular clouds. In: *Mobile Ad Hoc Networking*. [S.l.: s.n.], 2013.
- PEREIRA, R. et al. Fog-oriented hierarchical resource allocation policy in vehicular clouds. In: . [S.l.: s.n.], 2021.
- PEREIRA, R. et al. Fog-oriented hierarchical resource allocation policy in vehicular clouds. In: . [S.l.: s.n.], 2021.
- PEREIRA, R. et al. A novel fog-based resource allocation policy for vehicular clouds in the highway environment. In: . [S.l.: s.n.], 2019.
- PEREIRA, R. et al. Reliable: Resource allocation mechanism for 5g network using mobile edge computing. *Sensors*, v. 20, p. 5449, 09 2020.
- PUTERMAN, M. L. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. [S.l.]: John Wiley & Sons, 2014.
- PYUN, S. Y.; LEE, W.; CHO, D. H. Resource allocation for vehicle-to-infrastructure communication using directional transmission. *IEEE Transactions on Intelligent Transportation Systems*, v. 17, n. 4, p. 1183–1188, April 2016. ISSN 1524-9050.
- QUALCOMM. *Connecting vehicles to everything*. 2018. Disponível em: <https://www.qualcomm.com/invention/5g/cellular-v2x>.
- QUESSADA, M. S. et al. A bat bio-inspired mechanism for resource allocation in vehicular clouds. In: . [S.l.: s.n.], 2021.
- RAMANATHAN, R. Stochastic decision making using multiplicative ahp. *European Journal of Operational Research*, v. 97, n. 3, p. 543–549, 1997. ISSN 0377-2217. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221796002858>.
- REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: Association for Computing Machinery, 1987. (SIGGRAPH '87), p. 25–34. ISBN 0897912276. Disponível em: <https://doi.org/10.1145/37401.37406>.
- SAATY, T. L. *The analytic hierarchy process : planning, priority setting, resource allocation*. New York; London: McGraw-Hill International Book Co., 1980. ISBN 0070543712 9780070543713. Disponível em: http://www.worldcat.org/search?qt=worldcat_org_all&q=0070543712.
- SANAEI, Z. et al. Heterogeneity in mobile cloud computing: Taxonomy and open challenges. *IEEE Communications Surveys Tutorials*, v. 16, n. 1, p. 369–392, First 2014.
- SIBAİ, R. E. et al. Connectivity-aware service provision in vehicular cloud. In: *2015 International Conference on Cloud Technologies and Applications (CloudTech)*. [S.l.: s.n.], 2015. p. 1–5.
- STEPHAN, O. Taking vanet to the clouds. Emerald Group Publishing Limited, v. 7, n. 1, p. 7–21, Jan 2011. ISSN 1742-7371.

- TAHERKHANI, N.; PIERRE, S. Congestion control in vehicular ad hoc networks using meta-heuristic techniques. In: *Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*. New York, NY, USA: Association for Computing Machinery, 2012. (DIVANet '12), p. 47–54. ISBN 9781450316255. Disponível em: <https://doi.org/10.1145/2386958.2386966>.
- TAO, J. et al. Non-cooperative resource allocation scheme for data access in vanet cloud environment. In: *2015 Third International Conference on Advanced Cloud and Big Data*. [S.l.: s.n.], 2015. p. 190–196.
- WASICEK, A. et al. System simulation from operational data. In: *Proceedings of the 52Nd Annual Design Automation Conference*. New York, NY, USA: ACM, 2015. (DAC '15), p. 3:1–3:6. ISBN 978-1-4503-3520-1.
- WHAIIDUZZAMAN, M. et al. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, v. 40, p. 325 – 344, 2014. ISSN 1084-8045.
- WU, Q. et al. Smart fog based workflow for traffic control networks. *Future Generation Computer Systems*, v. 97, p. 825 – 835, 2019. ISSN 0167-739X.
- YANG, W. et al. Secrecy-based resource allocation for vehicular communication networks with outdated CSI. In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017. Disponível em: <https://doi.org/10.1109/vtcfall.2017.8288288>.
- YU, R. et al. Cooperative resource management in cloud-enabled vehicular networks. *IEEE Transactions on Industrial Electronics*, v. 62, n. 12, p. 7938–7951, Dec 2015.
- YU, R. et al. Cooperative resource management in cloud-enabled vehicular networks. *IEEE Transactions on Industrial Electronics*, v. 62, n. 12, p. 7938–7951, Dec 2015.
- YU, R. et al. Toward cloud-based vehicular networks with efficient resource management. *IEEE Network*, v. 27, n. 5, p. 48–55, September 2013. ISSN 0890-8044.
- ZHANG, T.; GRANDE, R. E. D.; BOUKERCHE, A. Vehicular cloud: Stochastic analysis of computing resources in a road segment. In: *Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. New York, NY, USA: Association for Computing Machinery, 2015. (PE-WASUN '15), p. 9–16. ISBN 9781450337595. Disponível em: <https://doi.org/10.1145/2810379.2810383>.
- ZHANG, X.; SHEN, Z.; YANG, D. A permutation-based model for analysis of resource allocation overheads in vehicular ad hoc networks. *IEEE Access*, v. 9, p. 12282–12290, 2021.
- ZHANG, Y. et al. Resource allocation in software defined fog vehicular networks. In: . [S.l.: s.n.], 2017. p. 71–76.
- ZHENG, K. et al. An smdp-based resource allocation in vehicular cloud computing systems. *IEEE Transactions on Industrial Electronics*, v. 62, n. 12, p. 7920–7928. ISSN 0278-0046.
- ZHENG, K. et al. An smdp-based resource allocation in vehicular cloud computing systems. *IEEE Transactions on Industrial Electronics*, v. 62, n. 12, p. 7920–7928, Dec 2015.