Alocação de Tarefas em Nuvens Veiculares Utilizando Jogos de Mercado

Aguimar R. Júnior[1], Joahannes B. D. da Costa[2], Geraldo P. Rocha Filho[3], Leandro A. Villas[2], Daniel L. Guidoni[4], Rodolfo I. Meneguette[1]

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (USP), São Carlos – SP – Brasil

[2]Instituto de Computação (UNICAMP), Campinas – São Paulo – Brasil

[3]Departamento de Ciência da Computação, Universidade de Brasília (UnB)
– Brasília – DF – Brasil,

[4]Departamento de Ciência da Computação, Universidade Federal de Ouro Preto, Brasil

aguimarjr@usp.br, geraldof@unb.br, guidoni@ufop.edu.br,

{joahannes.costa, leandro}@ic.unicamp.br, meneguette@icmc.usp.br

Abstract. *Globally, IoT devices will represent nearly half of the total connected devices by 2023, with the connected car applications sector being the fastest-growing category. As more computational resources integrate into vehicles, a new and diverse set of applications becomes possible, necessitating efficient resource allocation and management. So, in this paper, we propose a task allocation heuristic mechanism for vehicular cloud networks named ORANGE. It uses game theory, modeling the system as a market game aiming to maximize computational resource utilization under different request rates. We have analyzed and compared ORANGE against four algorithms by performing extensive simulations in 63 scenarios. The results showed that ORANGE presents statistically significant gains of up to 10.19% better than the second-placed algorithm regarding resource utilization, confirming the effectiveness of the proposed solution.*

Resumo. *Os dispositivos IoT representarão quase que metade de todos os dispositivos conectados mundialmente até 2023, sendo o setor de desenvolvimento de aplicações para carros conectados o de mais rápido crescimento. Com a incorporação cada vez maior de recursos computacionais aos veículos, o que viabiliza a criação de uma nova gama de aplicações, torna-se necessário alocar e gerenciar esses recursos de forma eficiente. Com isso em mente, este trabalho apresenta o ORANGE (alOcação de taRefAs em Nuvens utilizando joGos de mErcado), uma solução heurística para a alocação de tarefas em nuvens veiculares utilizando teoria dos jogos. A solução modela o sistema como um jogo de mercado que visa maximizar a utilização de recursos computacionais e considera diferentes taxas de requisição. O ORANGE foi avaliado com o uso extensivo de simulações e comparado a quatro outros algoritmos em 63 cenários. Considerando uma melhor utilização dos recursos, os resultados mostram que o ORANGE teve ganhos estatisticamente significativos de até 10.19% em relação ao segundo algoritmo melhor colocado. Confirmando assim, a eficácia da solução proposta.*

1. Introdução

Com o aumento do número de dispositivos Internet das Coisas (IoT) conectados, uma nova variedade de aplicações torna-se possível devido à incorporação de mais recur-

sos computacionais aos veículos, tais como capacidade de memória, processamento, armazenamento e largura de banda [CISCO 2020, Pereira et al. 2020]. No entanto, um ambiente com maior número de aplicações e recursos disponíveis traz também um novo desafio: a alocação desses recursos de forma eficiente. Então como atender de forma eficaz e eficiente às necessidades de diversas aplicações por recursos e ao mesmo tempo diminuir a quantidade de recursos ociosos em um ambiente de veículos conectados [Wei et al. 2021]?

Para responder a essa pergunta, [Abuelela and Olariu 2010, Eltoweissy et al. 2010] propõem o uso de computação em nuvem veicular (VC). As VCs estendem o conceito de computação em nuvem para uma rede veicular, permitindo que veículos com recursos computacionais, por exemplo, capacidade de armazenamento, processamento e largura de banda, compartilhem seus recursos ociosos com outros veículos que necessitem de recursos extras para concluir suas tarefas [Elahi et al. 2022]. São 3 as principais categorias de tarefas providas em redes veiculares [Liu et al. 2021, Meneguette et al. 2019a]: as relacionadas à segurança nas vias diminuindo o risco de acidentes; as relacionadas ao tráfego, citam-se aqui aplicações que auxiliem na redução do tempo de viagem e aliviem congestionamentos e, por fim, as que agregam valor ao usuário, por exemplo, facilitando o acesso à Internet.

Também, é apresentado o conceito de controlador de nuvens veiculares (VCT). O VCT é uma entidade na borda da rede responsável por criar e manter as VCs, assim como por gerenciar os recursos entre os veículos [Paul et al. 2017]. Esses veículos estão equipados com uma unidade de bordo (OBU), GPS, o mapa viário da cidade e o posicionamento das unidades de acostamento (RSU). Assim é possível que os veículos notifiquem suas posições atuais, recursos ociosos e solicitações por recursos às RSUs, que encaminham ao VCT. Além disso, os veículos quando se movem para uma área coberta por outra RSU, realizam o processo de *handover* e fazem as notificações necessárias. Essas comunicações entre veículos e infraestrutura (V2I) e entre um veículo e outro (V2V) tornaram-se possíveis com o licenciamento e a reserva de um espectro de frequências de rádio específicos para comunicação em redes veiculares e pela criação de padrões como o WAVE (*Wireless Access in Vehicular Environment*) que definiram um conjunto de protocolos, mais especificamente o IEEE 802.11p e a família IEEE 1609.x.

[Olariu et al. 2013] apresentam aplicações de VCs em cenários estáticos e dinâmicos. Citam por exemplo, estacionamentos de grandes aeroportos, onde os veículos ficam parados enquanto seus donos viajam. Os recursos computacionais desses veículos podem ser agregados em VCs cujo uso seria locado ao próprio aeroporto. Discute-se também cenários como estacionamentos de shoppings e o uso de VCs para o planejamento de grandes eventos esportivos, shows e festivais. Quanto a cenários dinâmicos, os autores citam a otimização do uso de semáforos e a utilização de faixas exclusivas de grandes vias de circulação para veículos coletivos (já existentes em grandes cidades brasileiras como São Paulo e Brasília). É citada também a troca de mensagens nas vias para maior segurança dos usuários a fim de evitar acidentes e mitigar congestionamentos. Entretanto, os autores destacam preocupações quanto à segurança, à privacidade e aos desafios arquiteturais que as VCs impõem. Quanto à segurança e à privacidade, citam a autenticação e a identificação de veículos em ambientes de alta mobilidade, o estabelecimento de relações de confiança entre os nós da rede e a heterogeneidade desses nós, assim como preocupações relacionadas à preservação da localização e da privacidade dos veículos e de seus donos. Parte desses desafios sanados, segundo os autores, com o uso de técnicas de virtualização.

Quanto aos desafios arquiteturais, ou seja, como as VCs se formam e trocam mensagens, os desafios surgem com o aumento do número de nós e a alta mobilidade dos

veículos [Coutinho and Boukerche 2019] em redes de topologias instáveis e com recursos limitados. Além disso, satisfazer uma variedade de requisições por serviços com diferentes necessidades (QoS) é outro desafio permanente que deve ser enfrentado. Como exemplos, a garantia de *delay* reduzido para aplicações de tempo real e de largura de banda mínima e acesso aos canais de frequência. Uma situação prática é revisitada por [Meneguette et al. 2019b] ao explorarem a detecção de acidentes com requisitos intolerantes ao atraso. Com diferentes QoS a serem atendidos, os objetivos podem ser contraditórios e conflitantes, para citar alguns: *throughput* e *delay*, utilização dos recursos, energia consumida e segurança [Cordeschi et al. 2015]. Impondo assim, desafios para a tomada de decisão de cada um dos dispositivos inteligentes, sejam eles veículos ou unidades de acostamento (RSU), obrigando-os a fazerem escolhas (*trade-offs*).

Na implicação de fazer escolhas, a literatura apresenta modelos para alocação de recursos em redes veiculares que utilizam diversas técnicas, entre elas: algoritmos gulosos [Yang et al. 2017], meta-heurısticas [Lieira et al. 2020], otimização combinatória [da Costa et al. 2020] e aprendizagem por reforço [Peng and Shen 2020]. Cada trabalho apresenta o contexto, a técnica utilizada e as premissas restritivas ao abordar o problema. No entanto, não consideram a natureza do objetivo de cada um dos veículos, que pode ser colaborativa, ou seja, com intenção de cooperarem entre si para maximizar um objetivo comum; ou competitiva, onde cada participante tenta maximizar seu próprio objetivo de forma egoísta. Com foco na modelagem e análise de situações interativas entre tomadores de decisão, a teoria dos jogos [Maschler et al. 2020] mostra seu potencial em ambientes altamente dinâmicos e instáveis - características que vão ao encontro das características das redes veiculares. A teoria dos jogos dispõe de modelos matemáticos para otimização de problemas complexos em que múltiplos agentes com objetivos contraditórios competem por recursos limitados ou cooperam para maximizar seus interesses em comum [Sun et al. 2021].

Com isso em mente, este trabalho modela o problema de alocação de tarefas em nuvens veiculares utilizando teoria dos jogos. É implementado um algoritmo heurístico denominado ORANGE, que organiza as VCs como um jogo de mercado único, criando assim um ambiente cooperativo que visa maximizar a utilização dos recursos disponíveis com diferentes taxas de requisição de serviço na rede. Os resultados mostram que o ORANGE apresentou ganhos significativos ao fazer uso dos recursos disponíveis das VCs quando comparado a 4 outras soluções presentes na literatura.

O restante deste artigo está organizado como se segue. A Seção 2 apresenta os principais trabalhos relacionados a esta pesquisa. A Seção 3 apresenta a modelagem do sistema e o detalhamento da solução implementada. A Seção 4 define a metodologia utilizada para avaliação da solução, os parâmetros de simulação e a discussão dos resultados obtidos. Por fim, a Seção 5 apresenta as conclusões e os direcionamentos para trabalhos futuros.

2. Trabalhos Relacionados

Na literatura, o problema de alocação de recursos em redes veiculares é discutido por diversos autores. [Yu et al. 2015] analisam o problema de gerenciamento e compartilhamento de largura de banda para viabilizar o uso de aplicações móveis em ambientes de nuvens veiculares. Foi proposto um jogo de coalizão que permite a cooperação entre os provedores de serviço a fim de compartilharem recursos ociosos. Cenários com diferentes taxas de requisição de serviço foram avaliados e os resultados indicam uma melhor utilização dos recursos computacionais. Com um aumento de até 75% na qualidade dos serviços prestados aos usuários aos utilizarem as aplicações, se comparados a cenários sem cooperação.

[Pereira et al. 2019] formulam o problema de alocação de recursos utilizando um método matemático multicritério em um ambiente rodoviário com o objetivo de maximizar a disponibilidade de recursos na rede veicular. O mecanismo implementado utiliza o paradigma de computação em névoa, o qual permite que os recursos sejam agregados e estejam próximos aos veículos, aumentando assim a quantidade de serviços por eles atendidos. No entanto, o estudo considera apenas o cenário de rodovias cuja mobilidade é mais previsível do que um cenário urbano, além de não variarem a taxa de requisição na rede veicular.

[Hattab et al. 2019] modelam o problema de alocação de tarefas como um problema de otimização combinatória cujo objetivo é minimizar o tempo de conclusão das tarefas nas VCs disponíveis. Porém, os autores presumem a formação de uma única VC, não há variação na taxa de requisição e levam em consideração apenas cenários estáticos. Já [da Costa et al. 2020] apresentam um mecanismo que considera o mesmo método para alocar tarefas computacionais em VCs. O objetivo desse trabalho foi selecionar um conjunto ótimo de tarefas a serem alocadas em tempo real em cada uma das VCs visando maximizar a utilização de recursos. Entretanto, cenário com diferentes taxas de requisição não foram considerados.

[Marques and Meneguette 2021] propõem uma política de alocação de recursos baseada em teoria dos jogos para maximizar a utilização de recursos em VCs analisando o equilíbrio de Nash do jogo proposto. Os autores formulam o problema como um jogo não-cooperativo onde os jogadores, nesse caso veículos, possuem estratégias de oferecer ou não um recurso e consumir ou não um serviço. A ideia principal do modelo é que tanto consumidores quanto provedores de recurso tenham algum tipo de ganho. No entanto, um equilíbrio de Nash não é necessariamente um ótimo de Pareto [Sun et al. 2021], o que a depender do perfil de estratégias escolhido pode levar a soluções não eficientes. Além disso, o trabalho também não considera variações nas taxas de requisição na rede.

Assim, com base nos trabalhos relacionados, observou-se que somente [Yu et al. 2015] utilizaram diferentes taxas de requisição ao analisar o problema de alocação de recursos. Entretanto, consideraram recursos disponíveis em data centers e não somente aqueles ociosos em veículos. Além disso, consideraram a existência de agrupamentos prévios desses recursos em unidades lógicas denominadas provedores de serviços. Assim, propõe-se o ORANGE, uma solução heurística que modela o problema de alocação de recursos como um jogo de mercado baseado em teoria dos jogos. Diferentemente dos trabalhos citados, a solução não pressupõe a existência de agregação prévia de recursos e considera cenários dinâmicos com diversas taxas de requisição.

3. Alocação de Tarefas em Nuvens Veiculares Utilizando Jogos de Mercado

Esta seção descreve a modelagem do sistema, assim como define o problema de alocação de tarefas em nuvens veiculares utilizando um jogo de mercado e apresenta em detalhes a solução proposta.

3.1. Cenário

A Figura 1 apresenta um cenário exemplificativo onde existem RSUs depositadas que cobrem, em termos de comunicação, toda a malha viária. Os veículos em circulação formam nuvens veiculares por meio de clusterizações a cada 60 segundos devido à alta mobilidade dos veículos, agrupando em seguida os recursos computacionais disponíveis em cada VC. O cenário é composto por x veículos, sendo que cada veículo v_i onde $i \in [1, x]$ é equipado com uma unidade de bordo (OBU) que possibilita a comunicação tanto entre os veículos quanto entre os veículos e as RSUs. As RSUs são responsáveis por coletar informações e requisições dos veículos em tempo real e repassá-las ao VCT. Existindo a necessidade

de um veículo executar uma tarefa que necessite de mais recursos do que dispõe, por exemplo, não há capacidade suficiente de processamento para concluir a tarefa, esse pode solicitar recursos extras. O VCT recebe essas solicitações e executa o ORANGE para determinar qual a melhor forma de alocar as tarefas em cada uma das VCs com o intuito de atender o máximo possível de tarefas enquanto maximiza a utilização dos recursos ociosos na rede veicular.

Figura 1: Cenário do sistema.

3.2. Definição de Mercado e Alocação

[Maschler et al. 2020] define mercado como um conjunto composto de produtores $N = \{1, 2, \ldots, n\}$ que negociam l *commodities*, sendo o conjunto delas dado por $L = \{1, 2, ..., l\}$. Essas *commodities* podem ser combinadas pelos produtores para produzirem bens de diferentes tipos, por exemplo, equipamentos eletrônicos ou produtos alimentícios. Assim, sejam os números reais não-negativos $R_+ := [0, \infty)$, um vetor de *commodities* é representado por $x = (x_j)_{j=1}^{L} \in R_+^L$ e será chamado de pacote de *commodities*. O pacote do produtor i é dado por x_i e a quantidade de cada *commodity* j representada por $x_{i,j}$. Cada produtor tem a sua disposição uma tecnologia de produção representada por uma função de produção $u_i : R_+^L \to R$: se $x_i \in R_+^L$ é o pacote de *commodities* do produtor i que pode então produzir uma soma de valor igual a $u_i(x_i)$. As funções de produção podem ser diferentes entre produtores, portanto a quantidade de valor que uma unidade de *commodity* j gera, pode variar de um produtor para outro. Assume-se que todo produtor i tem uma dotação inicial de *commodities* $a_i \in R_+^L$ e que os produtores podem negociar entre eles.

Se um mercado S é formado então seus membros negociam *commodities* entre si com o objetivo de maximizar o valor que produzirão. O total das dotações iniciais de *commodities* disponíveis é dado por $aS := \sum_{i \in S} a_j \in R_+^L$. Assim, aloca-se para cada um dos integrantes um pacote de *commodities* $x_i \in R_+^L$, sujeito à restrição imposta pela Equação 1. Com a realocação das *commodities* no mercado, os membros produzem um valor total igual a $\sum_{i \in S} u_i(x_i)$.

$$x(S) = \sum_{i \in S} x_i = \sum_{i \in S} a_i = a(S) \tag{1}$$

Portanto, formalmente, um mercado é descrito por um vetor $(N, L, (a_i, u_i)_{i \in N})$ tal que:

- $N = \{1, 2, ..., n\}$ é o conjunto de produtores;
- $L = \{1, 2, ..., l\}$ é o conjunto de *commodities*;
- $\forall i \in N$, $a_i \in R_+^L$ é o vetor de dotações iniciais do produtor i;
- $\forall i \in N$, $u_i : R_+^L \to R$ é a função de produção do produtor i.

Assim, ao assumir que $a_i \in R_+^L$ para todo $i \in N$, implica na existência de uma quantidade finita de cada *commodity* no mercado. Por fim, considera-se que uma alocação para a coalizão S é a coleção de pacotes de *commodities* $x_{i \in S}$, onde $x_i \in R_+^L$ para todo produtor $i \in N$ satisfazendo $x(S) = a(S)$.

3.3. Conversão de Mercado em um Jogo de Mercado

Na teoria dos jogos, jogos de coalizão modelam situações em que os jogadores cooperam entre si para atender seus objetivos [Maschler et al. 2020]. Dentre as diversas classes de jogos de coalizão, optou-se neste trabalho pelos jogos de mercado para a modelagem do problema de alocação. Os jogos de mercado foram escolhidos por possuírem por definição pelo menos uma solução ótima que resolve o problema modelado [Bondareva 1963, Shapley 1965].

Ainda segundo os mesmos autores, todo mercado pode ser associado a um jogo de coalizão, onde o conjunto de jogadores é o conjunto de produtores $N = \{1, 2, ..., n\}$ e o valor agregado de cada coalizão não vazia $S \subseteq N$ é dado pela Equação 2.

$$v(S) = \max \left\{ \sum_{i \in S} u_i(x_i) : x = (x_i)_{i \in S} \in X^S \right\} \tag{2}$$

Ou seja, o valor agregado de uma coalizão S é o máximo de valor que pode ser produzido por seus membros se negociarem *commodities* entre si. O jogo de coalizão $(N; v)$ é definido como sendo um jogo de mercado derivado do mercado $(N, L, (a_i, u_i)_{i \in N})$. Formalmente, um jogo de coalizão $(N; v)$ é um jogo de mercado se existe um número positivo l, um pacote inicial de *commodities* $a_i \in R_+^L$ para todo jogador $i \in N$ e uma função de produção $u_i : R_+^L \to R$ côncava e contínua tal que a Equação 2 seja satisfeita para toda coalizão $S \in P(N)$. Na subseção seguinte, é apresentada a modelagem do jogo proposto.

3.4. Modelagem do Jogo no Contexto de Alocação de Tarefas em Nuvens Veiculares

Apresentada a definição de mercado e como converter um mercado em um jogo de mercado, este trabalho faz analogias entre *commodities* e recursos computacionais; produtores e VCs ou VCT; coalizão e o conjunto formado pelas VCs+VCT e, por fim, realocação de *commodities* e alocação de tarefas. O modelo proposto é derivado do mercado $(N, L, (a_i, u_i)_{i \in N})$ tal que:

- $N = \{1, 2, ..., n\}$ é o conjunto formado pelas VCs e o VCT;
- $L = \{1, 2, ..., l\}$ é o conjunto de pesos das tarefas T, onde $l = \max\{peso(tarefa_i)_{i \in T})\}$;
- $\forall i \in N$, $a_i \in R_+^L$ é o vetor de *slots* disponíveis, sendo que $a_{i,j}$ representa a quantidade de tarefas de peso j alocadas se i é uma VC ou a serem alocadas se i representa o VCT. A Equação 3 apresenta os valores iniciais para os *slots*, onde $|T_j|$ representa a quantidade de tarefas com peso j a serem alocadas pelo VCT.
- $\forall i \in N$, $u_i : R_+^L \to R$ é a função de produção do participante i (VCs ou VCT) definida conforme a Equação 4, onde $|T_{i,j}|$ representa a quantidade de tarefas com peso j alocadas na VC i.

$$a_{i,j} = \left\{ \begin{array}{ll} 0, & se\, VC \\ |T_j|, & se\, VCT \end{array} \right\} \forall j \in [1, l] \tag{3}$$

$$u_i = \left\{ \begin{array}{ll} \sum_{j=1}^{l} j|T_{i,j}|, & se\, VC \\ 0, & se\, VCT \end{array} \right\} \forall j \in [1, l], i \in N \tag{4}$$

Em outras palavras, o valor agregado produzido por uma VC é igual à soma total do número de tarefas alocadas multiplicadas pelos seus respectivos pesos j. Assim, o VCT ao alocar tarefas nas VC participantes da coalizão pode gerar um valor agregado total correspondente a $\sum_{i \in N} u_i(x_i)$ conforme mostrado na subseção 3.2.

Consequentemente, para converter o mercado anterior em um jogo de mercado é necessário que se atenda à Equação 2 sujeita às restrições de recursos computacionais disponíveis Ω_i de cada uma das VCs $i \in N - 1$ da coalizão. Desta forma, o modelo é formalizado de acordo com a Equação 5. Na subseção seguinte são apresentados os detalhes de implementação da solução proposta.

$$\begin{array}{ll} Maximizar & \sum_{i \in S} u_i(x_i) \\ sujeito\ a & u_i(x_i) \leq \Omega_i \end{array} \tag{5}$$

3.5. ORANGE

Esta subseção detalha a implementação da solução proposta a fim de resolver o modelo de mercado modelado anteriormente. Conforme o Algoritmo 1, o conjunto de tarefas T e de nuvens veiculares V são informados ao ORANGE. Inicialmente, forma-se a coalizão (Linha 2) entre as VCs e o VCT, inicializam-se as dotações iniciais dos participantes do mercado (Linha 3) conforme Equação 3 e a dotação inicial do VCT é destacada (Linha 4), pois representa as demandas por recursos computacionais necessárias às tarefas a serem alocadas nas VCs.

Em seguida cria-se uma matriz (Linha 5) com a finalidade de manter todo o histórico de pacotes de recursos alocados em cada um dos participantes da coalizão, o valor agregado total gerado (Linha 6), a lista de tarefas atendidas é inicializada (Linha 7) e por fim ordena-se as VCs em ordem decrescente de recursos disponíveis (Linha 8). Para cada uma das VCs do mercado (Linha 9), se ainda existir necessidade do VCT por recursos a serem alocados (Linha 10), o ORANGE procura por recursos na VC a fim de atender total ou parcialmente as demandas do VCT. É o procedimento MERCADO que realiza a busca por recursos na nuvem (Linha 11) e será detalhado em breve. Por fim são atualizados o valor agregado total da coalizão (Linha 12), a lista de recursos a serem alocados (Linha 13), a lista de tarefas ainda não atendidas (Linha 14) e a lista de tarefas atendidas até o momento (Linha 15). Caso não existam mais demandas, não há necessidade de continuar percorrendo a lista de VCs (Linha 16-17). O algoritmo encerra e retorna o valor agregado gerado pela coalizão, a lista de tarefas atendidas, não atendidas e o histórico detalhado de recursos utilizados em cada uma das VCs (Linha 18).

O Algoritmo 2 detalha o procedimento MERCADO. Como entrada, são fornecidas as demandas por recursos, o total de recursos disponíveis na VC e as tarefas a serem alocadas. Inicialmente, a solução procura pelo valor do maior peso entre as tarefas a serem alocadas (Linha 2), inicializa o valor agregado a ser gerado pela alocação de tarefas na VC (Linha 3) e a lista de tarefas atendidas (Linha 4). A seguir, iniciando pelo valor de maior peso (J $\leftarrow$ L) e decrementando esse valor a cada iteração (Linha 5), o algoritmo salva quantas serão as tarefas com o peso (J) atendidas (Linha 6) e quantas delas são possíveis de ser alocadas na VC (Linha 7-12). Caso seja possível alocar pelo menos uma

Algoritmo 1: Algoritmo do ORANGE

Entrada:
T - conjunto de tarefas
V - nuvens veiculares
Saída:
U - valor agregado da coalizão
S - tarefas alocadas
X - histórico de alocação de recursos nas VCs
T - tarefas não alocadas

1 **início**
2 C $\leftarrow V \cup \{VCT\}$
3 A $\leftarrow$ `DotacoesIniciais(`*C, T*`)`
4 Ac $\leftarrow A[N]$
5 X $\leftarrow \vec{0}$
6 U $\leftarrow 0$
7 S $\leftarrow \emptyset$
8 `Ordenar(`*V*`)`
9 **para** $i \leftarrow 0$ **até** $N-1$ **faça**
10 **se** $sum(Ac) \neq 0$ **então**
11 $X[i], Ui, Au, Ta \leftarrow$ `MERCADO(`*Ac, V[i].total_recursos, T*`)`
12 $U \leftarrow U + Ui$
13 $Ac \leftarrow Ac - \{Au\}$
14 $T \leftarrow T - \{Ta\}$
15 $S \leftarrow S \cup \{Ta\}$
16 **senão**
17 **break**
18 **retorna** *U, S, X, T*

tarefa (Linha 13), o valor agregado gerado pela VC é atualizado (Linha 14), assim como a demanda restante por recursos (Linha 15) e o histórico de alocações também atualizado com o quantitativo (n) de tarefas com peso J que foi possível alocar (Linha 16). Por fim, são recuperadas as n tarefas com peso J que geram maior ganho e a lista de tarefas atendidas é atualizada (Linha 17). O procedimento retorna o histórico de alocações, o valor agregado gerado pela VC e a lista de tarefas alocadas (Linha 18).

Algoritmo 2: Algoritmo do MERCADO

Entrada:
Ac - recursos necessários para alocação
Ω - total de recursos da VC
T - tarefas a serem alocadas
Saída:
X - recursos alocados da VC
Ui - valor agregado gerado
Ta - tarefas alocadas

1 **início**
2 $L \leftarrow$ `ArgMax(`*Ac*`)`
3 $Ui \leftarrow 0$
4 $Ta \leftarrow \emptyset$
5 **para** $j \leftarrow L$ **até** 0 **faça**
6 $n \leftarrow Ac[j]$
7 $Uij \leftarrow n * j$
8 **se** $Uij > \Omega$ **então**
9 **repita**
10 $n \leftarrow n - 1$
11 $Uij \leftarrow n * j$
12 **até** $n \neq 0$ *e* $Ui + Uij > \Omega$
13 **se** $n > 0$ **então**
14 $Ui \leftarrow Ui + Uij$
15 $Ac[j] \leftarrow Ac[j] - n$
16 $X[j] \leftarrow n$
17 $Ta \leftarrow Ta \cup$ `NTarefasPesoK(`*n,j*`)`
18 **retorna** *X, Ui, Ta*

4. Avaliação Metodológica

Esta seção apresenta a metodologia e a métrica utilizadas na avaliação do ORANGE, assim como a análise dos resultados obtidos nas diferentes simulações. Os experimentos foram realizados no Simulador de Mobilidade Urbana (Simulation of Urban MObility - SUMO) [Lopez et al. 2018] (versão 1.10.0) e os algoritmos implementados na linguagem Python e conectados ao SUMO por meio da interface TraCI. O *trace* de mobilidade utilizado nas simulações foi o tráfego de Luxemburgo [Codeca et al. 2017] que contém 24 horas de mobilidade veicular com picos de até cinco mil veículos. Optou-se neste trabalho por avaliar o ORANGE no intervalo de 11 horas e meio-dia onde a densidade veicular e portanto com menos recursos disponíveis. Considerou-se também que uma vez alocada, a tarefa tem todos os recursos necessários para sua execução e será concluída com sucesso. Os parâmetros utilizados pelo ORANGE são apresentados na Tabela 1 e foram assim definidos para proporcionar diferentes cenários de simulação.

Tabela 1: Parametrização da simulação

Parâmetro	Valor
Faixa de comunicação	100m
Ocorrência de tarefas	Distribuição de Poisson
Número de tarefas (λ)	20, 50, 100
Peso das tarefas	20, 25, 30, 40, 50, 75, 100
Ganho das tarefas	2 x {20, 25, 30, 40, 50, 75, 100}
Quantidade de recursos	1, 2, 3
Algoritmo de clusterização	DBSCAN $\rightarrow$ eps = 100, min_samples = 2
Número de veículos	1000 $\sim$1500
Tempo de simulação	1h de Lust Scenario (Intervalo: 11-12h)

O DBSCAN [Ester et al. 1996] é utilizado para agrupar os veículos em VCs e ocorre a cada 60 segundos devido à alta mobilidade veicular. Assim que o agrupamento acontece, o VCT executa o ORANGE que viabiliza o jogo de mercado. As taxas de requisição no sistema seguem distribuições de Poisson, uma vez que as ocorrências de solicitações são independentes umas das outras [Kadhim and Hosseini Seno 2019] e seus valores médios de ocorrências variados com o propósito de analisar o comportamento do algoritmo com diferentes volumes de tarefas. Variou-se também os pesos das tarefas, ou seja, as quantidades de recursos necessários para alocá-las e os ganhos por alocá-las a fim de trazer maior aleatoriedade ao sistema e permitir uma análise mais abrangente do comportamento do ORANGE. Por fim, variou-se a quantidade de recursos compartilhados por cada um dos veículos, pois é importante observar o impacto que o compartilhamento de diferentes unidades de recurso traz para o problema de alocação como um todo [da Costa et al. 2020].

Como métrica, avaliou-se o percentual de utilização dos recursos computacionais ociosos das VCs. Calculado conforme a Equação 6, mede o grau de sucesso obtido na alocação de recursos quando comparado ao modelo definido pela Equação 5.

$$taxa\ de\ utilizacao = \sum_{i \in S} \frac{u_i\left(x_i\right)}{\Omega_i} \tag{6}$$

O ORANGE foi comparado a 4 outros algoritmos para alocação de tarefas em VCs, sendo 2 deles baseados em programação dinâmica e os 2 outros baseados em algoritmos gulosos. O MORFEU [da Costa et al. 2020] e o GREEDY-N (baseado em [Nabi et al. 2017]) consideram todas as VCs disponíveis enquanto que o algoritmo DP e o GREEDY consideram apenas uma VC no momento da alocação. Todos os algoritmos iniciam classificando a lista de tarefas em ordem decrescente de peso e utilizam primeiramente a VC de maior capacidade para alocar o maior número de tarefas possível.

4.1. Análise dos Resultados

O ORANGE foi avaliado em 63 cenários que são resultantes das combinações dos diferentes parâmetros estabelecidos na Tabela 1. Cada cenário foi simulado 33 vezes e se aplicou o teste de Friedman, teste não-paramétrico, em cada um deles para verificar a existência de diferença estatística entre os 5 algoritmos. Em seguida, para os casos onde a hipótese inicial tenha sido rejeitada, o teste post-hoc de Nemenyi [Demšar 2006] foi aplicado. Em todos os cenários onde a diferença entre os algoritmos foi estatisticamente significante, o ORANGE apresentou melhores desempenhos na métrica avaliada.

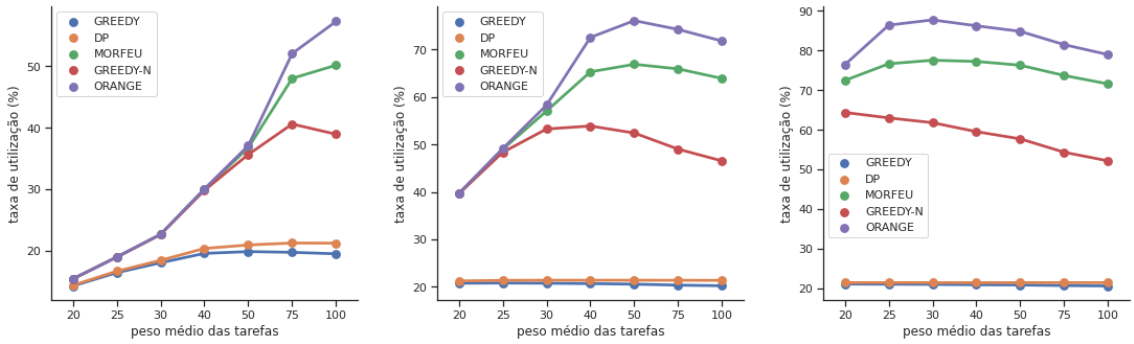
Foram calculados os coeficientes de correlação de Spearman entre o percentual de utilização dos recursos disponibilizados pelas VCs e os parâmetros de taxa de requisição (0.52), número de recursos por veículo (-0.28), peso das tarefas (0.33) e algoritmo utilizado (0.42). A correlação negativa entre o número de recursos por veículo e a taxa de utilização dos recursos se explica por cenários menos desafiadores com alta disponibilidade de recursos onde apenas parte desses é suficiente para atender às necessidades de alocação. Já a correlação positiva moderada entre a taxa de requisição e a variável dependente é analisada com mais detalhes com o auxílio das Figuras 2, 3 e 4.

Na Figura 2 são apresentados os resultados obtidos com diferentes taxas de requisição em cenários com 1 recurso compartilhado por veículo. Observa-se que em todos os cenários, independentemente da taxa utilizada, os algoritmos GREEDY e DP apresentam percentuais de utilização de recursos muito aquém dos demais. Esse comportamento decorre do uso de uma única VC, aquela com mais recursos disponíveis, para alocação de tarefas, não aproveitando assim os recursos de outras VCs que por ventura possam estar disponíveis no momento da alocação.

Na Figura 2a, em que a taxa de requisição é igual a 20, observa-se a partir de cenários com tarefas de peso igual a 50 que os algoritmos começam a se diferenciar e o ORANGE começa a se destacar. É nesse momento que a demanda por recursos disponíveis começa a gerar estresse na rede exigindo, dos algoritmos de alocação, tomadas de decisão nas estratégias de balanceamento utilizadas. Observa-se a partir de então que, com o aumento do peso das tarefas, acentua-se a diferença entre os algoritmos avaliados, com o ORANGE apresentando ganhos estaticamente significativos de 4.04% e 11.47% quando o peso é 75 e 7.13% e 18.36% quando o peso é 100 em relação aos algoritmos MORFEU e GREEDY-N respectivamente em ambos os casos.

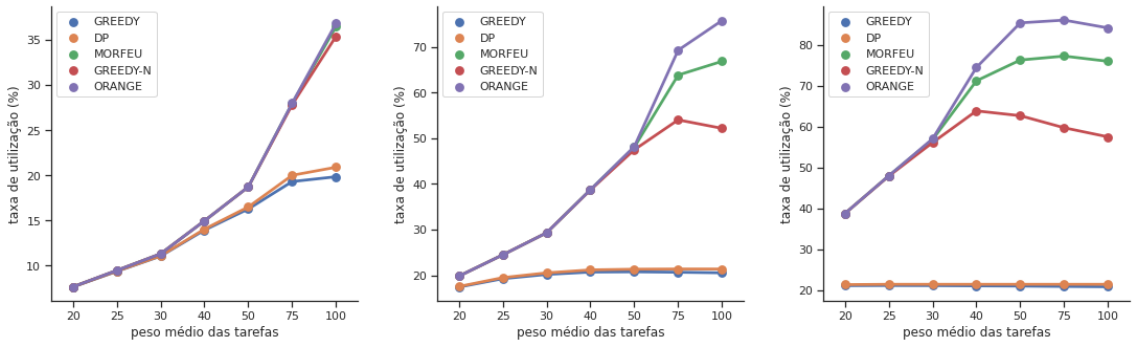
Na Figura 2b, em que a taxa de requisição aumenta para 50, pode-se observar o mesmo comportamento ao aumentar o peso das tarefas. No entanto, os algoritmos começam a divergir quando o peso é igual a 30, diferentemente do momento apresentado na Figura 2a. Essa divergência antecipada se explica pela maior taxa de requisição de tarefas no caso da Figura 2b e portanto maior volume de tarefas para alocação, causando o estresse da rede mais cedo com relação aos pesos das tarefas. Para esses cenários, a solução proposta apresentou ganhos significativos de 7.26% e 18.69% quando o peso é 40; 9.21% e 23.71% quando o peso é 50; 8.32% e 25.25% quando o peso é 75 e 7.89% e 25.29% quando o peso é 100 em relação aos algoritmos MORFEU e GREEDY-N respectivamente em todos os casos.

Na Figura 2c, em que a taxa de requisição aumenta para 100, os algoritmos divergem desde o início das simulações com tarefas de peso igual a 20. Nesses cenários o ORANGE apresentou ganhos significativos de 3.97% e 12.05% quando o peso é 20; 9.76% e 23.42% quando o peso é 25; 10.19% e 25.96% quando o peso é 30; 9% e 26.73% quando o peso é 40; 8.56% e 27.16% quando o peso é 50; 7.78% e 27.17% quando o peso é 75 e 7.4% e 26.84% quando o peso é 100 em relação aos algoritmos MORFEU e GREEDY-N respectivamente em todos os casos.



(a) Taxa de requisição $= 20$. **(b)** Taxa de requisição $= 50$. **(c)** Taxa de requisição $= 100$.

Figura 2: Taxa de utilização de recursos (%) em cenários onde os veículos compartilham 1 recurso cada.



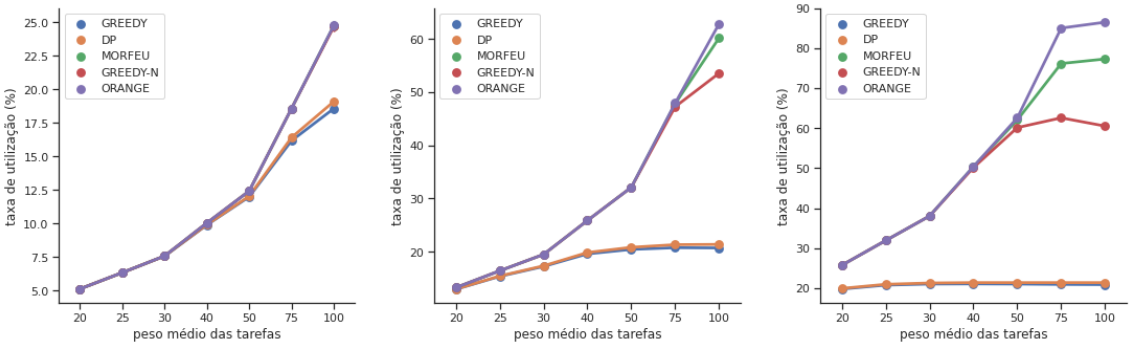
(a) Taxa de requisição $= 20$. **(b)** Taxa de requisição $= 50$. **(c)** Taxa de requisição $= 100$.

Figura 3: Taxa de utilização de recursos (%) em cenários onde os veículos compartilham 2 recursos cada.

Em ambientes com maior disponibilidade de recursos (Figura 3 e 4), os cenários se tornam menos desafiadores diminuindo a possibilidade de afirmar que um dos algoritmos difere estatisticamente dos demais. Foi somente com o aumento da taxa de requisição para 50 e 100 que surgiram cenários com ganhos significativos entre os algoritmos avaliados. Sendo assim, não foi possível determinar estatisticamente a superioridade de um dos algoritmos nos cenários das Figuras 3a e 4a.

Na Figura 3b, com 2 recursos compartilhados por veículo e taxa de requisição igual a 50, o ORANGE apresentou ganhos significativos de 5.43% e 15.24% quando o peso é 75 e 8.93% 23.6% quando o peso é 100 em relação aos algoritmos MORFEU e GREEDY-N respectivamente em todos os casos. Já na Figura 3c, com taxa de requisição igual a 100 e mesmo número de recursos compartilhados, o ORANGE apresentou ganhos de 3.28% e 10.64% quando o peso é 40; 9.1% e 22.75% quando o peso é 50; 8.81% e 26.37% quando o peso é 75 e 8.17% e 26.68% quando o peso é 100 em relação aos algoritmos MORFEU e GREEDY-N respectivamente em todos os casos.

Na Figura 4b, com 3 recursos compartilhados por veículo e taxa de requisição igual a 50, o ORANGE apresentou ganhos significativos de 2.65% e 9.28% com peso igual a 100 em relação aos algoritmos MORFEU e GREEDY-N. E, na Figura 4c, com taxa de requisição igual a 100 e mesmo número de recursos compartilhados, o ORANGE apresentou ganhos de 8.87% e 22.46% quando o peso é 75 e 9.2% e 25.99% quando o peso é 100 em relação aos algoritmos MORFEU e GREEDY-N respectivamente em ambos os casos.



(a) Taxa de requisição = 20. **(b)** Taxa de requisição = 50. **(c)** Taxa de requisição = 100.

Figura 4: Taxa de utilização de recursos (%) em cenários onde os veículos compartilham 3 recursos cada.

Portanto, a solução proposta neste artigo, denominada ORANGE, mostrou-se mais eficiente ao fazer uso dos recursos computacionais ociosos disponibilizados pelas VCs para alocação de tarefas. Em todos os cenários em que foi possível afirmar estatisticamente que um algoritmo é superior aos demais, o ORANGE apresenta os melhores resultados. Por fim, observou-se também que variar a taxa de requisição de tarefas na rede permitiu avaliar comparativamente os algoritmos em cenários onde apenas a variação do peso das tarefas não seria suficiente.

5. Conclusão

Até o final de 2023, o setor de desenvolvimento de aplicações para veículos conectados será o setor de IoT com mais rápido crescimento [CISCO 2020]. Para acompanhar esse crescimento é necessário que a alocação de recursos entre os dispositivos seja realizada de forma eficiente. Este trabalho propõe o ORANGE, uma solução heurística que utiliza teoria dos jogos e modela o problema de alocação de tarefas em nuvens veiculares visando maximizar a utilização dos recursos disponíveis considerando diferentes taxas de requisição. Os resultados numéricos, após o uso extensivo de simulações, mostram que o ORANGE foi superior em todos os cenários onde houve diferença estatisticamente significativa e que a variação da taxa de requisição é um importante parâmetro a ser levado em consideração na avaliação de algoritmos de alocação de recursos. Em trabalhos futuros, pretende-se considerar a utilização de *Shapley values* a fim de realizar a alocação de acordo com a contribuição marginal de cada VC para a coalizão. Além disso, tornar possível a formação de mais de uma coalizão desde que mais atrativas para os participantes e propor mecanismos de incentivo para a participação da VC em uma coalizão.

Referências

Abuelela, M. and Olariu, S. (2010). Taking VANET to the clouds. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia - MoMM '10*, page 6, Paris, France. ACM Press.

Bondareva, O. N. (1963). Some applications of linear programming methods to the theory of cooperative games. *Problemy kibernetiki*, 10(119):139.

CISCO (2020). Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper. `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html`. Acessado em 4 de Novembro, 2021.

Codeca, L., Frank, R., Faye, S., and Engel, T. (2017). Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63.

Cordeschi, N., Amendola, D., Shojafar, M., and Baccarelli, E. (2015). Distributed and adaptive resource management in Cloud-assisted Cognitive Radio Vehicular Networks with hard reliability guarantees. *Vehicular Communications*, 2(1):1–12.

Coutinho, R. W. L. and Boukerche, A. (2019). Guidelines for the Design of Vehicular Cloud Infrastructures for Connected Autonomous Vehicles. *IEEE Wireless Communications*, 26(4):6–11.

da Costa, J. B. D., Peixoto, M. L. M., Meneguette, R. I., Rosário, D. L., and Villas, L. A. (2020). MORFEU: Mecanismo baseado em Otimização Combinatória para Alocação de Tarefas em Nuvens Veiculares. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 505–518. SBC.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30.

Elahi, M. M., Rahman, M. M., and Islam, M. M. (2022). An efficient authentication scheme for secured service provisioning in edge-enabled vehicular cloud networks towards sustainable smart cities. *Sustainable Cities and Society*, 76:103384.

Eltoweissy, M., Olariu, S., and Younis, M. (2010). Towards Autonomous Vehicular Clouds. In Zheng, J., Simplot-Ryl, D., and Leung, V. C. M., editors, *Ad Hoc Networks*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pages 1–16, Berlin, Heidelberg. Springer.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231, Portland, Oregon. AAAI Press.

Hattab, G., Ucar, S., Higuchi, T., Altintas, O., Dressler, F., and Cabric, D. (2019). Optimized Assignment of Computational Tasks in Vehicular Micro Clouds. In *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, EdgeSys '19, pages 1–6, New York, NY, USA. Association for Computing Machinery.

Kadhim, A. J. and Hosseini Seno, S. A. (2019). Maximizing the Utilization of Fog Computing in Internet of Vehicle Using SDN. *IEEE Communications Letters*, 23(1):140–143.

Lieira, D. D., Quessada, M. S., Cristiani, A. L., and Meneguette, R. I. (2020). Resource Allocation Technique for Edge Computing Using Grey Wolf Optimization Algorithm. In *2020 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6.

Liu, L., Chen, C., Pei, Q., Maharjan, S., and Zhang, Y. (2021). Vehicular Edge Computing and Networking: A Survey. *Mobile Networks and Applications*, 26(3):1145–1168.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE.

Marques, H. A. P. and Meneguette, R. I. (2021). Um Mecanismo de Alocação de Recursos em Nuvens Veiculares baseado em Teoria dos Jogos. In *Anais Estendidos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 241–248. SBC.

Maschler, M., Solan, E., and Zāmîr, Š. (2020). *Game Theory*.

Meneguette, R. I., Boukerche, A., and Pimenta, A. H. M. (2019a). Avarac: An availability-based resource allocation scheme for vehicular cloud. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3688–3699.

Meneguette, R. I., Rodrigues, D. O., da Costa, J. B. D., Rosario, D., and Villas, L. A. (2019b). A Virtual Machine Migration Policy Based on Multiple Attribute Decision in Vehicular Cloud Scenario. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6.

Nabi, M., Benkoczi, R., Abdelhamid, S., and Hassanein, H. S. (2017). Resource assignment in vehicular clouds. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. Ieee.

Olariu, S., Hristov, T., and Yan, G. (2013). The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds. In Basagni, S., Conti, M., Giordano, S., and Stojmenovic, I., editors, *Mobile Ad Hoc Networking*, pages 645–700. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Paul, A., Chilamkurti, N., Daniel, A., and Rho, S. (2017). Vehicular network (VN) model. pages 43–75.

Peng, H. and Shen, X. (2020). Deep Reinforcement Learning Based Resource Management for Multi-Access Edge Computing in Vehicular Networks. *IEEE Transactions on Network Science and Engineering*, 7(4):2416–2428.

Pereira, R. S., Lieira, D. D., Silva, M. A. C. d., Pimenta, A. H. M., da Costa, J. B. D., Rosário, D., Villas, L., and Meneguette, R. I. (2020). Reliable: Resource allocation mechanism for 5g network using mobile edge computing. *Sensors*, 20(19).

Pereira, R. S. U., Lieira, D. D., da Silva, M. A. C., Pimenta, A. H. M., da Cost, J. B. D., Rosario, D., Meneguette, R. I., and Gutierrez, C. A. (2019). A novel fog-based resource allocation policy for vehicular clouds in the highway environment. *2019 Ieee Latinamerican Conference On Communications (ieee Latincom)*, page 6.

Shapley, L. S. (1965). On balanced sets and cores. Technical report, RAND CORP SANTA MONICA CALIF.

Sun, Z., Liu, Y., Wang, J., Li, G., Anil, C., Li, K., Guo, X., Sun, G., Tian, D., and Cao, D. (2021). Applications of Game Theory in Vehicular Networks: A Survey. 23(4):2660–2710.

Wei, W., Yang, R., Gu, H., Zhao, W., Chen, C., and Wan, S. (2021). Multi-Objective Optimization for Resource Allocation in Vehicular Cloud Computing Networks. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–10. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Yang, W., Zhang, R., Chen, C., and Cheng, X. (2017). Secrecy-Based Resource Allocation for Vehicular Communication Networks with Outdated CSI. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pages 1–5.

Yu, R., Huang, X., Kang, J., Ding, J., Maharjan, S., Gjessing, S., and Zhang, Y. (2015). Cooperative Resource Management in Cloud-Enabled Vehicular Networks. *IEEE Transactions on Industrial Electronics*, 62(12):7938–7951.