

# Um Mecanismo de Alocação de Recursos em Nuvens Veiculares baseado em Teoria dos Jogos

Henrique Andrews Prado Marques<sup>1</sup>, Rodolfo I. Meneguette<sup>1</sup>

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (USP)  
São Carlos, SP – Brasil.

henrique.andrews.marques@usp.br, meneguette@icmc.usp.br

**Abstract.** *Intelligent transport systems (ITS) seeks to solve or at least mitigate some of the problems faced by the road sector through the provision of computer services. To assist in the provision of these services, the creation of vehicular clouds emerged as a way to bring computational resources and services closer to vehicular applications. However, most of the times the computational resources present in the vehicles that make up the vehicle clouds are underused. Therefore, in order to make the best use of these resources, this work proposes a resource allocation policy based on Game Theory to maximize their use and then compare this approach with others known in the literature and verify its performance in relation to other evaluation metrics beyond the use of resources.*

**Resumo.** *Os sistemas de transporte inteligentes (ITS) buscam resolver ou pelo menos mitigar alguns dos problemas enfrentados pelo setor rodoviário através do fornecimento de serviços computacionais. Para auxiliar na provisão desses serviços, a criação de nuvens veiculares surgiu como um meio de aproximar os recursos computacionais e serviços das aplicações veiculares. No entanto, na maior parte das vezes os recursos computacionais presentes nos veículos que compõem as nuvens veiculares são subutilizados. Sendo assim, para melhor aproveitar estes recursos este trabalho propõe uma política de alocação de recursos baseada em Teoria dos Jogos para maximizar a utilização dos mesmos e posteriormente comparar esta abordagem com outras conhecidas na literatura e averiguar seu desempenho com relação a outras métricas de avaliação além da utilização de recursos.*

## 1. Introdução

As nuvem veiculares (VC - vehicular cloud) visam uma cooperação eficiente na comunicação, alocação de tarefas e compartilhamento de recursos nas VANETs, e elas têm um impacto notável em diversas aplicações de ITS por utilizar rapidamente estes recursos, tais como processamento, armazenamento e comunicação para tomadas de decisão sem necessitar da computação em nuvem tradicional para tal [Brik et al. 2019].

Ao agregar o excesso de recursos que estão na borda da rede, cria-se um conjunto de serviços que são disponibilizados para os veículos, podendo ser utilizados em ambientes estáticos e dinâmicos [Hagenauer et al. 2019, Correa et al. 2014]. Sendo assim, os veículos se tornam componentes chave no contexto de aplicações de ITS, já que passam a atuar, além da gestão de tráfego, como fonte para coleta e processamento de dados em tempo real. Portanto, mecanismos que façam uso dos recursos das VCs de maneira eficiente são imprescindíveis, de forma que estes serviços sejam usados da melhor maneira

possível, fornecendo poder computacional mais próximo das aplicações veiculares e com qualidade para quem os consome.

Os principais desafios que envolvem a proposição de abordagens para uma eficiente alocação de recursos e tarefas em VCs giram em torno da natureza dinâmica das VANETs [Meneguette et al. 2019a]. A alocação e processamento de tarefas com requisitos intolerantes ao atraso, como detecção de acidentes, ou tarefas que exigem alto poder computacional, como processamento de imagens de trânsito e outras aplicações de multimídia de modo geral, acaba sendo bastante desafiador por conta da alta mobilidade dos veículos [Meneguette et al. 2019b].

Assim, para melhor aproveitar estes recursos este trabalho propõe uma política de alocação de recursos baseada em Teoria dos Jogos para maximizar a utilização dos mesmos e posteriormente comparar esta abordagem com outras conhecidas na literatura e averiguar seu desempenho com relação a outras métricas de avaliação além da utilização de recursos. O resto do artigo está organizado da seguinte forma: na Seção 2 são apresentados trabalhos relacionados a alocação de recursos e tarefas. Na Seção 3 são discutidos sobre a formulação do problema, o cenário da aplicação e o algoritmo desenvolvido. Na Seção 3.4 são apresentados e discutidos os resultados obtidos através das avaliações e por fim na Seção 5 são apresentadas as conclusões finais e diretrizes para trabalhos futuros.

## 2. Trabalhos Relacionados

Existem trabalhos na literatura que abordam o problema de alocação de tarefas e recursos em VANETs/VCs e alguns deles são referenciados nesta seção. Como por exemplo o trabalho de [Pereira et al. 2019] é proposta uma política de alocação de recursos em VCs em ambiente rodoviário. A política proposta tem por objetivo maximizar a disponibilidade de recursos na VC. Veículos conectados devem cooperar para a criação de um reservatório de recursos, para dessa forma se ter conhecimento dos recursos disponíveis, os quais serão fornecidos pelos veículos e o conjunto de nevoeiros onde mais recursos serão gerados e serviços alocados.

Em [da Costa et al. 2020] propõem um mecanismo para alocação de tarefas em VCs baseado em otimização combinatória e o comparam com outras 3 abordagens. Os resultados obtidos mostram tanto um maior uso dos recursos disponíveis quanto um maior número de tarefas alocadas e maior ganho em comparação com os outros métodos. É simulado um cenário urbano em tempo real onde as tarefas são geradas ao longo da execução e são alocadas nas VCs que são recomputadas a cada minuto devido à mobilidade dos veículos. Para geração das VCs e agregação dos recursos, é utilizada uma técnica de clusterização bastante conhecida na literatura, o DBSCAN *Density Based Spatial Clustering of Application with Noise* [Ester et al. 1996].

Em [Lieira et al. 2020] propõem-se um algoritmo que adota a técnica meta-heurística conhecida como *Grey Wolf Optimization* [Mirjalili et al. 2014] para a escolha da melhor *Edge* quando for alocar os recursos do veículo. Neste trabalho, é considerado que os veículos possuem recursos de processamento, armazenamento, tempo e memória. O algoritmo faz uso destes recursos para computar a adequação de cada *Edge* e decidir em qual delas alocar, se possível. Essa abordagem é comparada com outras duas políticas e obteve um número menor de serviços recusados e apresentou um baixo número de bloqueios durante a busca por uma *Edge*.

### 3. Um Mecanismo de Alocação de Recursos em Nuvens Veiculares baseado em Teoria dos Jogos

Nessa seção descreveremos em detalhes todo o processo de desenvolvimento da política de alocação de recurso, que inclui descrição do modelo de sistema, definição do problema.

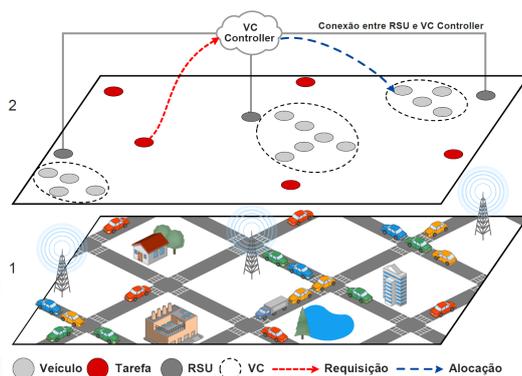
#### 3.1. Modelo de Sistema

A figura 1 representa o modelo do sistema em duas visões distintas. A visão número 1, que está embaixo, apresenta o cenário físico com os veículos e as RSUs, e a visão 2 representa o que está por trás, no caso como funciona a interação entre os componentes do sistema. Neste cenário, os veículos em deslocamento formam VCs por meio de clusterizações. Quando um veículo não pertence à nenhuma VC, mas precisa realizar o processamento de alguma tarefa e seus recursos computacionais não são suficientes para tal, solicita-se ao controlador de nuvem veicular (*VC Controller*) os recursos necessários para o atendimento da tarefa.

Considera-se um cenário composto por  $x$  veículos, onde cada veículo  $v_i$  possui uma identificação única ( $i \in [1, x]$ ) e é equipado com uma OBU *On-Board Unit* que permite tanto a comunicação entre veículos (V2V) quanto entre veículos e RSU (V2I). Por meio das RSUs é possível a comunicação com o *VC Controller*, cujo papel é definir, dado um conjunto de VCs disponíveis, qual delas é capaz de atender uma determinada tarefa que requer um certo poder computacional. As RSUs coletam informações dos veículos em tempo real, e quando os mesmos requisitam recursos para alocarem suas tarefas, o controlador de VC tem o conhecimento de quais nuvens possuem os recursos necessários para realizar a alocação das tarefas.

O *VC Controller* executa uma determinada política de alocação e seleciona alguma das VCs disponíveis para atender a requisição por recursos e alocar a tarefa, de forma mais rápida possível e fazendo uso do máximo de recursos computacionais ociosos da VC.

Figura 1. Ilustração representativa da Arquitetura do Sistema



Fonte: [da Costa et al. 2020]

#### 3.2. Definição do Problema

Considerando um conjunto de tarefas  $T$  com  $\{t = 1, \dots, n\}$ , em que cada tarefa  $t$  é representada por uma tupla  $(id_t, p_t, g_t)$ , onde  $id_t$  corresponde ao identificador único de cada tarefa,  $p_t$  o peso da tarefa (quantidade de recursos necessários para ser alocada) e  $g_t$  o ganho/recompensa por alocação bem-sucedida. Neste cenário, as VCs são os grupos

de veículos formados a partir de um processo de clusterização, este baseado em algum critério pré-estabelecido, e cada VC corresponde a soma dos recursos compartilhados por cada veículo que faz parte da respectiva nuvem.

Neste trabalho o foco da política de alocação baseada em Teoria dos Jogos será na maximização do uso dos recursos computacionais das nuvens veiculares dentro das limitações do modelo de jogo que será proposto. O tipo de jogo é não-cooperativo, o qual se concentra na previsão das ações e recompensas dos jogadores individuais e na análise do equilíbrio de Nash, e simultâneo, onde os jogadores agem sem saber qual a ação que os outros irão tomar. Assim, os jogadores busca maximizar seus lucros ou minimizar suas perdas e por conta da dinamicidade de uma rede veicular. Portanto, o problema tem como intuito:

$$\max \sum_{t=1}^n p_t \alpha_t \leq \Omega_j$$

### 3.3. Modelagem do Jogo

Definido os jogadores, pondera-se então a respeito das possíveis estratégias adotadas por cada um deles. Um veículo que demanda por um serviço pode escolher entre consumir ou não o serviço oferecido (permitir ou não a alocação de uma determinada tarefa) dado o custo para tal, enquanto que a VC também possui duas opções, oferecer ou não os recursos computacionais necessários para a execução da tarefa.

A ideia principal do modelo de jogo proposto é que ambas as partes envolvidas (veículos consumidores e VCs/veículos provedores) tenham algum tipo de ganho, seja ele monetário ao oferecer um serviço em que há um custo envolvido ou valor agregado através do atendimento do serviço solicitado. A ideia secundária é o jogo operar de acordo com a oferta de recursos computacionais e peso das tarefas, fazendo então o custo variar em detrimento desses fatores. Por exemplo, no caso de haver poucos recursos disponíveis o preço por recurso aumenta e caso haja uma grande disponibilidade de recursos o preço deles reduz.

Segue abaixo as equações de ganho para cada combinação de estratégias dos jogadores:

- **Consumir & Oferecer:**

$$X_{11} = g_t - p_t \cdot c(p_t, \Omega_j)$$

$$Y_{11} = p_t \cdot c(p_t, \Omega_j)$$

- **Consumir & Não oferecer:**

$$X_{12} = -g_t$$

$$Y_{12} = -2 \cdot p_t \cdot c(p_t, \Omega_j)$$

- **Não consumir & Oferecer:**

$$X_{21} = -2 \cdot g_t$$

$$Y_{21} = -p_t \cdot c(p_t, \Omega_j)$$

- **Não consumir & Não oferecer:**

$$X_{22} = 0$$

$$Y_{22} = 0$$

- ★  $c(p_t, \Omega_j)$  é um custo por recurso em função da quantidade de recursos computacionais disponíveis  $\Omega_j$  na VC com maior oferta e  $p_t$  o peso da tarefa em questão.
- ★  $X_{ij}$  e  $Y_{ij}$  representam respectivamente os ganhos dos consumidores e provedores.

Para encontrar o Equilíbrio de Nash, analisa-se cada combinação de estratégia:

– **Combinação de estratégia (Consumir & Oferecer)**

Neste primeiro caso, a recompensa de quem consome o serviço é dada por  $X_{11}$ , onde  $g_t$ , já definido anteriormente, é o valor agregado pela execução da tarefa  $t$  e  $p_t$  seu peso em quantidade de recursos necessários para o processamento. A recompensa da nuvem veicular  $vc_j$  que fornece os recursos computacionais para o atendimento da tarefa  $t$  se dá por  $Y_{11}$ , em que  $c(p_t, \Omega_j)$  é uma função que varia de acordo com o peso  $p_t$  da tarefa  $t$  e a quantidade de recursos disponíveis  $\Omega_j$  da VC. Dependendo do resultado de  $X_{11}$ , nenhum dos jogadores consegue aumentar seu ganho ao trocar de estratégia, já que os outros ganhos são neutros ou negativos, então essa combinação é um possível equilíbrio de Nash.

– **Combinação de estratégia (Consumir & Não oferecer)**

Neste segundo caso, onde há a escolha de consumir mas não de oferecer, o ganho do consumidor ( $X_{12}$ ) é negativo pelo fato de não ter sua tarefa atendida. O provedor será penalizado com um ganho negativo dobrado ( $Y_{12}$ ), pois antes da formação do jogo é considerado que há recursos suficientes para atendimento da tarefa  $t$  de peso  $p_t$ . Essa combinação não representa um possível equilíbrio de Nash já que tanto o jogador  $X$  poderia aumentar sua recompensa ao escolher não consumir quanto o jogador  $Y$  ao decidir oferecer.

– **Combinação de estratégia (Não consumir & Oferecer)**

Neste terceiro caso, em que as ações são de não consumir e oferecer, o consumidor tem uma recompensa negativa dobrada ( $X_{21}$ ) como forma de penalização e o provedor tem uma recompensa negativa ( $Y_{21}$ ) por não ter seus recursos computacionais utilizados. Da mesma maneira que a combinação anterior, esta não é um equilíbrio de Nash pelos mesmos motivos de que ambos jogadores poderiam aumentar seus ganhos ao mudarem suas ações.

– **Combinação de estratégia (Não consumir & Não oferecer)**

Neste último caso, como não há interesse de nenhuma das partes em consumir ou oferecer o serviço, não há necessidade de penalização e o ganho de ambos é neutro ( $X_{22} = Y_{22} = 0$ ). Similarmente ao primeiro caso, nesta combinação nenhum dos jogadores consegue elevar sua recompensa ao escolher a outra estratégia, configurando então como um equilíbrio de Nash.

### 3.4. Metodologia para Simulação

A simulação foi realizada utilizando o SUMO *Simulation of Urban Mobility*<sup>1</sup> versão 1.4.0, um simulador de mobilidade urbana de código aberto. Os algoritmos apresentados foram implementados na linguagem *Python*<sup>2</sup> e a conexão com o SUMO feita através do *TraCI Traffic Control Interface*<sup>3</sup>. De forma que o cenário de tráfego urbano seja o mais realista possível, foi utilizado o *trace* de mobilidade *Luxembourg SUMO Traffic (LuST)* [Codeca et al. 2017].

<sup>1</sup><https://sumo.dlr.de/docs/>

<sup>2</sup><https://www.python.org/>

<sup>3</sup><https://sumo.dlr.de/docs/TraCI.html>

Este cenário contém 24 horas de mobilidade urbana com até aproximadamente 5000 veículos circulando nos horários de pico. A faixa de horário escolhida foi das 11:00 às 12:00, quando há uma baixa densidade veicular e conseqüentemente uma maior escassez de recursos computacionais compartilhados nas VCs, para então avaliar como as políticas de alocação operam quando a situação é mais desafiadora.

O intervalo de clusterização definido foi de 60s por conta da dinamicidade da rede e alterações recorrentes nas VCs. As tarefas são geradas no momento em que o agrupamento dos veículos é feito e então as políticas as alocam nas VCs disponíveis. A ocorrência de tarefas no sistema, independentes umas das outras, é definida seguindo uma distribuição de Poisson com média de tarefas  $\lambda = 25$ . O raio de comunicação escolhido foi de 100 metros e mínimo de 2 veículos por nuvem veicular. Houve também variação da média de peso  $p_t$  das tarefas geradas nas simulações com  $\mu = \{1, 5, 10, 15, 20\}$ , o ganho de alocação  $g_t$  variou com  $2 \times \mu$  e a quantidade de recursos computacionais compartilháveis por veículo em  $\omega_i = \{1, 2, 3\}$ .

Abaixo seguem as métricas utilizadas para avaliação de desempenho dos algoritmos:

#### 1. Ganho de alocação

Representa valor total agregado pela alocação de tarefas.

$$ganho = \sum_{t=1}^n g_t \mid g_t \in S$$

#### 2. Utilização de recursos

Representa a porcentagem de recursos computacionais de VCs utilizados.

$$utilizacao = \frac{\sum_{t=1}^n p_t \mid p_t \in S}{\sum_{j=1}^m \Omega_j} \cdot 100$$

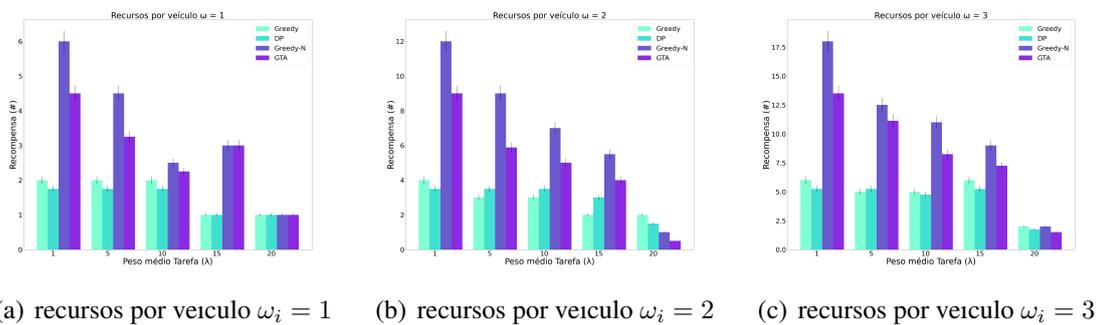
★  $S$  representa o conjunto de tarefas que tiveram uma alocação bem sucedida.

Para avaliar o desempenho da política de alocação deste trabalho, há comparação com outras, sendo elas uma abordagem envolvendo programação dinâmica a qual será referida nos resultados como DP *Dynamic Programming*, uma gulosa referida como *Greedy* em que o algoritmo considera apenas uma VC e por fim uma gulosa referida como *Greedy-N* em que o algoritmo considera todas as VCs disponíveis.

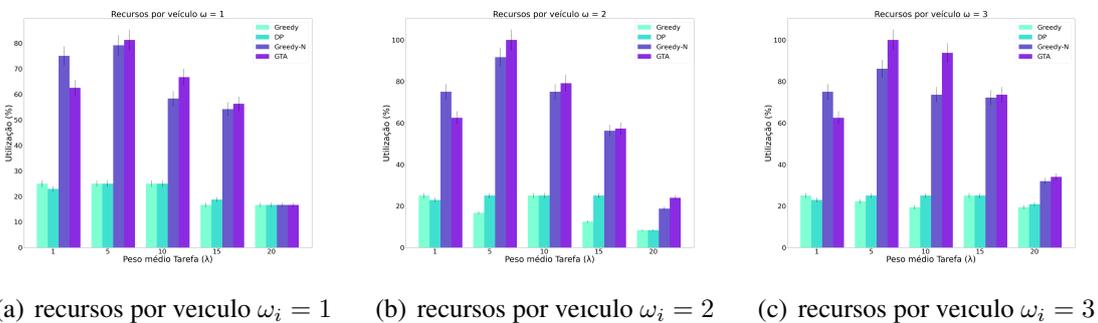
## 4. Resultados

As Figuras 2(a), 2(b) e 2(c) são apresentados os valores de recompensa que cada algoritmo obteve de acordo com as tarefas que foram alocadas por eles. Observa-se que de maneira geral, o GTA obteve ganhos maiores do que as abordagens *Greedy* e DP, as quais fazem uso apenas da VC com maior abundância de recursos computacionais, mas obteve ganhos menores quando comparado com a abordagem *Greedy-N*, que faz uso de todas as VCs disponíveis para realizar a alocação das tarefas. Também vale ressaltar que a medida que o peso médio das tarefas aumenta, o ganho das políticas vai se tornando próximo devido a dificuldade de atender as tarefas cada vez mais pesadas nas VCs com a mesma quantidade de recursos anterior.

As Figuras 3(a), 3(b) e 3(c) mostra-se a porcentagem de recursos utilizados por cada algoritmo em suas alocações. É evidente ao observar os resultados que o GTA cumpre seu principal objetivo que é maximizar a utilização dos recursos computacionais disponíveis, tendo resultado melhor que todos os outros algoritmos. Todavia, é visível que maior utilização dos recursos não implica diretamente em um maior atendimento ou maior ganho, já que nas outras duas métricas o GTA teve um desempenho inferior ao *Greedy-N*. Ao priorizar tarefas mais pesadas, o GTA acaba inevitavelmente atendendo um número menor de tarefas, e as mesmas por mais pesadas que sejam, não é garantido que possuam um valor agregado consideravelmente maior que outras tarefas que exijam uma menor quantidade de recursos das VCs.



**Figura 2. Recompensa das tarefas por peso medio de tarefa**



**Figura 3. Porcentagem de utilizacao dos recursos por peso medio de tarefa**

## 5. Conclusão

Neste artigo apresentou e detalhou diversos elementos importantes do mecanismo de alocação de recurso, começando pelo modelo de sistema e descrição da problemática em que o projeto está inserido, em seguida discorrendo a respeito das atividades realizadas para que se chegasse à solução proposta. Os ganhos nas estratégias adotadas e a racionalidade em suas decisões, se adequando melhor a uma realidade onde há um custo envolvido na utilização dos recursos e serviços fornecidos nos ITS e VANETs. Este trabalho também comprova que uma maior utilização dos recursos computacionais não vem necessariamente acompanhada de um maior ganho de alocação ou maior número de tarefas atendidas, fato este salientado na parte de análise dos resultados, significando que ainda há bastante espaço para melhorias e trabalho a ser desenvolvido nesta política. Como trabalhos futuro pretendemos melhorar a estratégia dos jogadores para melhor que o sistema tenha um ganho maior.

## Agradecimento

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) processo número #2020/07162-0 pelo apoio financeiro para o desenvolvimento desta pesquisa.

## Referências

- Brik, B., Khan, J. A., Ghamri-Doudane, Y., and Lagraa, N. (2019). Publish: A distributed service advertising scheme for vehicular cloud networks. In *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–6.
- 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.
- Correa, C., Ueyama, J., Meneguette, R. I., and Villas, L. A. (2014). Vanets: An exploratory evaluation in vehicular ad hoc network for urban environment. In *2014 IEEE 13th International Symposium on Network Computing and Applications*, pages 45–49.
- da Costa, J., Peixoto, M., Meneguette, R., Rosário, D., and Villas, L. (2020). Morfeu: Mecanismo baseado em otimização combinatória para alocação de tarefas em nuvens veiculares. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 505–518, Porto Alegre, RS, Brasil. SBC.
- 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*, page 226–231. AAAI Press.
- Hagenauer, F., Higuchi, T., Altintas, O., and Dressler, F. (2019). Efficient data handling in vehicular micro clouds. *Ad Hoc Networks*, 91:101871.
- 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.
- 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.
- Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61.
- Pereira, R. S., Lieira, D. D., da Silva, M. A., Pimenta, A. H., da Costa, J. B., Rosário, D., and Meneguette, R. I. (2019). A novel fog-based resource allocation policy for vehicular clouds in the highway environment. In *2019 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6.