

MARVEL: Um Algoritmo Meta-heurístico para Alocação de Recursos em Redes Veiculares ad-hoc

Euclides N. Gottsfritz^{1,2}, Matheus S. Quessada¹, Douglas D. Lieira^{1,2},
Geraldo P. Rocha Filho³, Rodolfo I. Meneguette⁴

¹Departamento de Ciências da Computação
Universidade Estadual Paulista (UNESP) São José do Rio Preto – SP – Brasil

²Departamento de Informática
Instituto Federal de Educação, Ciência e Tecnologia (IFSP) – Catanduva, SP – Brasil

³Departamento de Ciências Exatas e Tecnológica
Universidade Estadual do Sudoeste da Bahia (UESB) – Vitória da Conquista, BA – Brasil

⁴Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (USP) – São Carlos, SP – Brasil

{euclides.gottsfritz, matheus.quessada}@unesp.br,

ddlleira@ifsp.edu.br, geraldo.rocha@uesb.edu.br, meneguette@icmc.usp.br

Abstract. *Decision support technologies based on Intelligent Transport Systems (ITS) have been modernized over time by remodeling known techniques. In vehicular networks, clustering is used to increase service capacity, since each vehicle is a repository of storage, processing, and memory resources when idle. As these resources are limited and the topology is highly dynamic, there is a need to use efficient algorithms that support decision-making during the allocation of these resources. In this work, we propose MARVEL, a meta-heuristic algorithm for resource allocation in ad-hoc Vehicular Networks that implements Harris Hawk Optimization (HHO) with the objective of optimizing the search process for idle resources. The results indicate that the MARVEL method is more effective in resource allocation, especially in services that require a large amount of resources, with a higher success rate and a greater number of resources allocated in the same period of time.*

Resumo. *Tecnologias de suporte a decisão baseados em Sistemas de Transporte Inteligente (ITS) vem se modernizando ao longo do tempo através da remodelagem de técnicas já conhecidas. Em redes veiculares a clusterização é utilizada para aumentar a capacidade de atendimento de serviços, já que cada veículo é um repositório de recursos de armazenamento, processamento e memória quando ociosos. Como esses recursos são limitados e a topologia é altamente dinâmica existe a necessidade do uso de algoritmos eficientes que apoiem a tomada de decisão durante a alocação desses recursos. Neste trabalho propomos o MARVEL, um algoritmo meta-heurístico para alocação de recursos em Redes Veiculares ad-hoc que implementa a Otimização do Falcão Harris (HHO) com o objetivo de otimizar o processo de busca por recursos ocioso. Os resultados indicam que o método MARVEL é mais eficaz na alocação de recursos, especialmente em serviços que requerem grande quantidade de recursos, com maior taxa de sucesso e maior número de recursos alocados no mesmo período de tempo.*

1. Introdução

Como facilitador das necessidades diárias populacionais, os veículos automotores têm destaque na rotina de muitas pessoas. Seja de motocicleta, carro, caminhão ou ônibus,

grande parte da população os utiliza para se locomoverem, para iniciar ou encerrar uma rotina de trabalho e até mesmo para o lazer de fim de semana. Com essa perspectiva, foi observado no relatório mensal de novembro de 2022 da Federação Nacional da Distribuição de Veículos Automotores (FENABRAVE) que houve um crescimento de aproximadamente 4,47% no acumulado anual de emplacamentos de veículos novos em 2022 com relação ao mesmo período de 2021, com tendência de crescimento dessa margem para o mês de Dezembro. Em números absolutos, o acumulado anual de janeiro a novembro de 2022 houveram cerca de 141 mil emplacamentos a mais que o mesmo período de 2021 [FENABRAVE 2022].

Esse crescente aumento de veículos, aliado com a falta de estrutura para suportar tal necessidade de tráfego, traz problemas socioeconômicos com os maiores engarrafamentos que geram maiores gastos de combustíveis e conseqüentemente maiores emissões de gás carbônico, além do estresse vivenciado a cada dia que ocasionarão futuros problemas de saúde [Peixoto et al. 2023]. Uma proposta para auxiliar o tratamento desses problema é com o uso de um Sistema de Transporte Inteligente (ITS - *Intelligent Transportation Systems*) em que os veículos estão equipados com sensores, interface de comunicação e poder computacional para trocarem informações e compartilhar recursos entre si [Meneguette and Boukerche 2020]. Assim é possível explorar o ambiente onde transitam para que consigam lidar com os desafios que os centros urbanos proporcionam, e conseqüentemente reduzir o impacto socioeconômico e ambiental [Meneguette et al. 2021].

Para que a comunicação possa ocorrer, os veículos devem ser organizados em uma Rede Veicular *ad-hoc* (VANET - *Veicular Ad-Hoc Network*) que dará suporte para a comunicação entre os veículos [Meneguette and Boukerche 2017]. Neste modelo o processamento e armazenamento dessa informação é feito entre os próprios veículos, que detém pouco poder computacional se analisados separadamente, mas quando agregados conseguem suprir a demanda uns dos outros por esses recursos. Para essa situação utiliza-se a técnica de *Cluster* Veicular (*Veicular Cluster*), que organiza a comunicação entre os veículos próximos e elege um membro como Cabeça de *Cluster* (CH - *Cluster Head*) para centralizar as tomadas de decisões e organizar o fluxo de informação, o que auxilia a sobrecarga de mensagens na rede e também minimiza o problema de capacidade computacional e alta de latência de comunicação [Husnain et al. 2022].

Devido aos recursos finitos e não escaláveis dos veículos, é necessário um bom gerenciamento para otimizar o uso desses recursos. Uma abordagem comum é a adaptação de otimizadores de problemas contínuos e não lineares para determinar quais veículos devem atender a cada serviço [Liu et al. 2021]. Entre esses otimizadores, destacam-se os algoritmos meta-heurísticos, que se baseiam em fenômenos naturais para encontrar soluções múltiplas. Esses algoritmos usam variáveis estocásticas e produzem soluções aproximadas, o que significa que não é necessário um grande poder computacional por um longo período para encontrar uma solução viável [Yang and Karamanoglu 2020].

Com isso em mente, este artigo apresenta o MARVEL, um algoritmo Meta-heurístico para Alocação de Recursos em Redes VEiculares *ad-hoc*, com o objetivo de otimizar o compartilhamento de recursos computacionais veiculares ociosos entre eles. Quando um veículo não possui recurso próprio suficiente, tais como processamento, armazenamento e memória, para executar um serviço, é escolhido um outro veículo na rede para atendê-lo, sejam serviços de mensagens de texto, multimídias, monitoramento de tráfego, sugestões de rotas, entretenimento e streaming de áudio/vídeo em centros urbanos, como a cidade de Brasília por exemplo. O MARVEL é modelado com base na Otimização do Falcão Harris (HHO - *Harris Hawk Optimization*), um método de otimização inspirado no comportamento de caça dessa espécie de Falcão contra presas fugitivas, normalmente coelhos. Nesta abstração, a presa fugitiva é o serviço e os falcões são os membros da VANET, ou seja, o falcão que alcançar e abater a presa primeiro será o

veículo que atenderá aquela requisição. Para que isso ocorra o MARVEL calcula o *fitness* de cada falcão a cada iteração até encontrar o mais apto para atender o serviço. Com isso a pesquisa possui duas principais contribuições:

1. Desenvolvimento de um algoritmo meta-heurístico para alocação de recursos em redes veiculares *ad-hoc*;
2. Minimização dos custos e maximização dos recursos através da alocação eficiente dos mesmos.

O restante do artigo está organizado da seguinte maneira: Na Seção 2 são apresentados os trabalhos relacionados a alocação de recursos, tarefas e serviços em redes veiculares. Através da Seção 3 é apresentada a proposta de trabalho com abstração do algoritmo, explicação de seu funcionamento e a definição do problema. Na Seção 4 são apresentados e discutidos os parâmetros da simulação, avaliações e seus resultados. E por fim, através da Seção 5 é apresentada a conclusão e direções para trabalhos futuros.

2. Trabalhos relacionados

Nesta seção, serão apresentados os principais problemas abordados em estudos da literatura, tais como: i) otimização de canais; ii) alocação de recursos, que envolve a utilização de um conjunto de recursos para atender a uma fila de tarefas; e iii) alocação de tarefas, que corresponde a um conjunto de tarefas que precisam ser realizadas por veículos com recursos disponíveis. Embora esses estudos apresentem diferentes cenários e tecnologias para gerenciamento de recursos, tarefas e serviços, todos eles compartilham o objetivo de otimizar esses processos dentro de VANETs.

Os autores em [Shalini and Jenicka 2021], utilizaram um algoritmo baseado no método Greedy para otimizar a alocação de recursos em um cenário de VANETs. O método proposto prevê a probabilidade dos melhores canais disponíveis para alocação de recurso, baseado no esquema de matriz de peso guloso estimado desenvolvido. Os autores analisaram o desempenho do algoritmo proposto em termos de taxa de entrega de pacotes, taxa de transferência, atraso, confiabilidade e taxa de perda de mensagens. Ao ser comparado com técnicas da literatura, apresentou melhor eficácia. Entretanto, os autores não analisam a quantidade de serviços atendidos e negados. Em [Li et al. 2019], os autores também desenvolveram um algoritmo baseado no método Greedy, que atende a demanda de recursos limitados ao espectro de canais em rede cognitiva veicular (CVN - *Cognitive Vehicular Network*). Para minimizar o problema de otimização de canais, o esquema proposto foi comparado com um algoritmo tradicional da Teoria dos Jogos na otimização de *throughput*, com o objetivo de maximizar a vazão do CVN. Os resultados dos experimentos mostraram que o método proposto teve uma grande vantagem durante as simulações. Mas, os autores não visam maximizar a utilização de recursos das tarefas computacionais.

Em [Pereira et al. 2019], os autores se basearam no Método Matemático de Decisão de Atributos Múltiplos para desenvolver o NANCY, um mecanismo que utiliza o método de Processo de Hierarquia Analítico (AHP - *Analytic Hierarchy Process*) para computar um fator de influência para cada parâmetro utilizado. O método foi aplicado num cenário de computação veicular em ambiente rodoviário. Foram analisadas as métricas de serviços atendidos, serviços bloqueados e serviços negados e o NANCY apresentou melhor desempenho quando comparado com técnicas tradicionais da literatura. No entanto, os autores não utilizaram um simulador de redes para analisar os resultados. No trabalho de [Liera et al. 2022b], foi utilizado um algoritmo meta-heurístico para otimização do processo de alocação de recursos computacionais em VANETs. Os autores desenvolveram o algoritmo MARIA baseado na Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) e aplicaram em um cenário com serviços de computação de borda para realizar as simulações. O mecanismo foi comparado com

algoritmos tradicionais da literatura e avaliados em relação a quantidade de serviços aceitos para alocação e serviços recusados. Os resultados demonstram que o MARIA foi mais eficiente para maximizar os atendimentos. Entretanto, a simulação não considerou as interferências que uma rede pode sofrer durante a execução do algoritmo.

Um algoritmo meta-heurístico também foi utilizado por [Ladeira et al. 2022a] para lidar com as limitações de recursos computacionais de serviços de computação de borda em VANETs. O MARTIAL foi desenvolvido com base no Algoritmo de Otimização da Baleia (WOA - *Whale Optimization Algorithm*) para otimizar o processo de alocação de tarefas. As métricas de tarefas alocadas, tarefas recusadas e tarefas bloqueadas foram consideradas para analisar o desempenho do mecanismo proposto e os resultados demonstraram que o MARTIAL conseguiu maximizar as tarefas atendidas e minimizar os bloqueios e recusas. Mas os autores não aplicaram o algoritmo considerando fatores como atraso e interferência na transmissão de pacotes. O WOA também foi utilizado por [Sing et al. 2022] para otimizar o processo de alocação de tarefas de aplicações IoT em serviços de computação em nuvem e névoa (*Fogs*). Dois módulos foram desenvolvidos pelos autores. O primeiro para classificação de tarefas, utilizando Fuzzy. E o segundo para descarregamento e alocação das tarefas, em que o processo é baseado na otimização do algoritmo meta-heurístico WOA. O algoritmo desenvolvido denominado WORA, foi comparado com outros métodos e apresentou melhor desempenho na economia de energia e na conclusão bem-sucedida das tarefas. Entretanto, o trabalho não considera as limitações de serviços de borda.

A Tabela 1 relaciona as principais características dos trabalhos como os métodos utilizados, a estrutura em que o problema se insere e o problema propriamente dito.

Tabela 1. Comparação das publicações abordadas nos Trabalhos Relacionados

Trabalho	Método	Estrutura	Problema
[Li et al. 2019]	Greedy	CVN	Otimização de canais
[Pereira et al. 2019]	AHP	FOG	Alocação de recursos
[Ladeira et al. 2022a]	WOA	EC	Alocação de tarefas
[Ladeira et al. 2022b]	PSO	EC	Alocação de recursos
[Shalini and Jenicka 2021]	Greedy	V2X	Alocação de recursos
[Sing et al. 2022]	WOA	Cloud/FOG	Alocação de tarefas
MARVEL	HHO	V2V	Alocação de recursos

Os trabalhos de [Li et al. 2019, Pereira et al. 2019, Shalini and Jenicka 2021, Sing et al. 2022] não consideram as limitações dos serviços de computação de borda. Enquanto os trabalhos de [Ladeira et al. 2022a, Ladeira et al. 2022b] não utilizam um simulador de redes para considerar as variáveis de interferências de comunicação durante o processo de tomada de decisão dos algoritmos. De forma geral, os trabalhos relacionados foram aplicados em estruturas de VANETS, porém o MARVEL é o único que trabalha exclusivamente com comunicação V2V, a fim de diminuir a complexidade da rede.

Deste modo, neste trabalho é proposto o MARVEL, que visa otimizar o processo de alocação de recursos computacionais em VANETs, com base no algoritmo meta-heurístico HHO. Um ambiente de simulação de comunicação entre os veículos é implementado para considerar todos os problemas de interferência de comunicação que podem ocorrer e, assim, se aproximar de um cenário real.

3. Algoritmo meta-heurístico para Alocação de Recursos em VANETs

Nesta seção é apresentado o MARVEL, um algoritmo meta-heurístico para alocação de recursos em VANETs. A estrutura desta seção é composta pela descrição do algoritmo, declaração do problema, parâmetros utilizados nas simulações, bem como os algoritmos avaliados e as métricas utilizadas para sua avaliação.

3.1. Algoritmo

O MARVEL foi elaborado para gerenciar a alocação de recursos e tem a capacidade de escolher o membro ideal de um *Cluster* para atender a um determinado serviço. Esse algoritmo tem como base a Otimização do Falcão Harris (HHO) que foi proposto matematicamente por [Heidari et al. 2019]. O algoritmo HHO foi inspirado no comportamento cooperativo de caça às presas fugitivas, geralmente coelhos, dos falcões Harris. Os autores modelaram matematicamente o comportamento dessas aves para que ocorra a busca das melhores soluções em diversos problemas de otimização. Normalmente aves de rapina caçam sozinhas, porém isso não se aplica a realidade dos falcões Harris que se caracterizam por viverem em grupos e cooperarem entre si para a caça e compartilhamento de alimentos [Heidari et al. 2019].

A escolha do HHO se dá devido a capacidade de lidar com a volatilidade desse tipo de estrutura, além de se mostrar adaptativo a vários cenários [Hossain et al. 2021, Husnain et al. 2022, Marwah et al. 2022]. Equipado com duas estratégias diferentes de exploração e outras quatro estratégias de aproveitamento, o algoritmo foi adaptado para que cada membro do *Cluster* seja considerado um falcão, e que o serviço a ser atendido seja a presa.

Para o correto funcionamento do MARVEL, é necessário respeitar algumas premissas:

1. Cada veículo do *Cluster* é considerado um falcão, ou seja, um membro da população;
2. O membro da população com menor valor de *fitness* terá seus recursos alocados para o serviço solicitado;
3. A presa sempre tentará fugir dos falcões;
4. A ação de cada falcão dependerá do valor de 3 variáveis: Uma variável estocástica (q), capacidade de fuga energia (r) e energia ($|E|$);
5. A primeira posição de cada falcão é gerada aleatoriamente, porém suas atualizações ocorrem com base em algumas informações dos falcões (posição média da população) e da presa (posição, energia e potência de salto) além de variáveis estocásticas.

A variável $|E|$ sinaliza a energia que a presa possui, ela é utilizada como referência na transição entre as diferentes fases do algoritmo. Ela será decrescida a cada iteração e seus valores vão de 2 (totalmente fortalecida) até 0 (totalmente exaurida). De acordo com [Heidari et al. 2019], o valor inicial de $|E|$ como 2 possibilita um maior deslocamento. Com o seu decremento a cada iteração, com base no tempo inicial e atual, o falcão e a presa se movimentam menos. Até que suas posições não sofram mais alterações, ou seja, quando $|E|$ igual 0.

Pode-se observar através da Figura 1 que $|E|$ controlará duas transições. A primeira ocorre quando $|E| < 1$, logo o algoritmo irá parar a exploração do ambiente e iniciará a fase de aproveitamento (cerco à presa), através da técnica nomeada de Cerco Leve. A segunda transição ocorrerá $|E| < 0.5$, quando o algoritmo intensificará o cerco através da técnica nomeada de Cerco Pesado.

A variável q se relaciona somente ao processo de exploração, ela definirá se a exploração vai ocorrer de forma aleatória, quando $q \geq 0.5$, ou com base na posição média dos falcões, quando $q < 0.5$. Existe 50% de chance de cada uma dessas opções acontecerem visto que a cada iteração, durante o processo de exploração, um novo valor é gerado dentro e sempre dentro do intervalo de 0 e 1.

Já a variável r indica a capacidade de fuga da presa, o que definirá se os falcões realizarão movimentos adicionais a cada iteração. Ela também é uma variável estocástica que tem seus valores entre 0 e 1, onde sempre que $r \geq 0.5$ será um indicativo de que a presa é incapaz de escapar dos falcões e $r < 0.5$ indica que essa chance existe e nesse

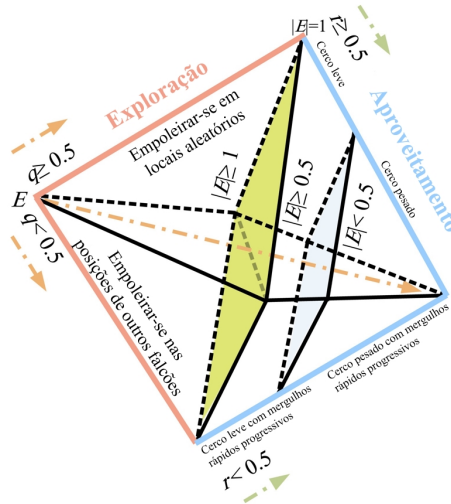


Figura 1. Descrição das fases de exploração e aproveitamento do MARVEL, adaptado de [Heidari et al. 2019].

caso os falcões utilizarão a técnica de mergulhos rápidos e progressivos para conseguir acompanhar a presa que é representado matematicamente no algoritmo com o uso do conceito de Voo de Levy (LF - *Levy Flight*) [Barthelemy et al. 2008] em sua estrutura.

Durante a rotina do algoritmo é realizado o cálculo do *fitness*, que acontece para os membros da população a cada iteração. Esse valor é utilizado durante as iterações e exprime a proximidade daquele membro da população para a solução ideal (ótimo global) que no caso da função de *benchmark* escolhida, função de Rosenbrok (Eq. 1), é 0 [Hussain et al. 2017]. Ou seja, ao final da execução o membro da população com menor valor de *fitness* será o membro mais apto para processar o serviço devida a sua proximidade com a solução ideal [Wong and Ming 2019].

$$Rosenbrock(x) = \sum_{i=1}^{d-1} \left[100 (x_{i+1} - x_t^2)^2 + (x_i - 1)^2 \right] \quad (1)$$

O algoritmo 1 é o pseudo-código do MARVEL, elaborado para facilitar seu entendimento. É possível observar os requisitos para cada acontecimento de maneira simplificada e inteligível.

Como parâmetros de entrada, o MARVEL recebe as informações sobre os recursos ociosos dos veículos e os requisitos mínimos para execução do serviço, enquanto a variável de saída será o *id* do veículo escolhido para atender a essa requisição.

Logo na linha 1 a população é iniciada com o cálculo da distância Euclidiana dos falcões/veículos com os requisitos do Serviço. Na linha 2 é iniciado o primeiro laço de repetição onde as iterações ocorrerão até que *t* seja igual ao número de membros do *Cluster*. Da linha 3 a 24, existe outro laço de repetição inserido dentro do primeiro, ou seja, todas as ações acontecerão para cada membro da população *i* no *range(0, Membros)*.

O valor de *fitness* do atual Membro é calculado na linha 4. Com base nesse valor, $X_{servico}$ é encontrado como o Membro com o menor dentre eles (linha 5). A linha 6 gera o valor da energia inicial e força do salto, enquanto o cálculo do valor da energia *E* acontece na linha 7.

Com *E* calculado, a primeira checagem para iniciar a movimentação do Membro acontece. Se $|E| \geq 1$, uma de duas diferentes ações ocorrerão (fase de exploração, linha

Algoritmo 1 MARVEL

Entrada: *Membros, serviço*

Saída: *melhorMembro*

```
1: Inicializa a população  $X_i$  com o cálculo da Distância Euclidiana dos recursos dos Membros do Cluster com os requisitos do Serviço
2: para  $t$  in range(0, Membros) faça
3:   para  $i$  in range(0, Membros) faça
4:     Calcula o valor de fitness do membro
5:     Define a posição do serviço  $X_{servico}$  através do menor fitness
6:     Atualiza a energia  $E_0$  e a força do salto  $J$ 
7:     Atualiza a energia  $E$  com base na energia inicial  $E_0$  e iteração  $t$ 
8:     se ( $|E| \geq 1$ ) então
9:       se ( $q \geq 0.5$ ) então
10:        Atualiza o vetor de localização aleatoriamente
11:      senão se ( $q < 0.5$ ) então
12:        Atualiza o vetor de localização com base na posição média da população
13:      fim se
14:    fim se
15:    se ( $|E| < 1$ ) então
16:      se ( $r \geq 0.5$  &  $|E| \geq 0.5$ ) então
17:        Atualiza o vetor de localização com base na localização do serviço  $X_{servico}$ , energia  $E$  e força do salto  $J$ 
18:      senão se ( $r \geq 0.5$  &  $|E| < 0.5$ ) então
19:        Atualiza o vetor de localização com base na localização do serviço  $X_{servico}$  e energia  $E$ 
20:      senão se ( $r < 0.5$  &  $|E| \geq 0.5$ ) então
21:        Atualiza o vetor de localização com base na localização do serviço  $X_{servico}$ , energia  $E$ , força do salto  $J$  e valor de Levy Flight
22:      senão se ( $r < 0.5$  &  $|E| < 0.5$ ) então
23:        Atualiza o vetor de localização com base na localização do serviço  $X_{servico}$ , energia  $E$  e valor de Levy Flight
24:      fim se
25:    fim se
26:  fim para
27: fim para
28: retorna melhorMembro
```

8 à 14). A primeira acontece caso $q \geq 0.5$ (linha 9), então o Membro selecionado da população terá sua localização atualizada aleatoriamente (linha 10). Caso $q < 0.5$ (linha 11), sua posição será calculada com base na posição média da população de falcões (linha 12).

Caso o valor de E seja inferior a 1, o algoritmo transitará para a fase de aproveitamento (linha 15 à 25), então uma de quatro diferentes ações poderão ocorrer e em todas elas a posição do serviço/presa $X_{servico}$ e valor de sua energia E serão utilizados para calcular a próxima posição do Membro selecionado da população.

Se $r \geq 0.5$ e $|E| \geq 0.5$ (linha 16), o cálculo da nova posição considerará a força de salto J (linha 17). Porém se $r \geq 0.5$ e $|E| < 0.5$ (linha 18), então J será ignorado (linha 19).

O mesmo acontece quando $r < 0.5$, porém é adicionado ao cálculo o valor de *Levy Flight*. Logo, se $r < 0.5$ e $|E| \geq 0.5$ (linha 20), a atualização de posição se dará com as informações de força do salto J e o valor da equação de *Levy Flight* (linha 21). E caso $r < 0.5$ e $|E| < 0.5$ (linha 22), J será ignorado e a equação considera apenas o valor de *Levy Flight* (linha 23).

O segundo laço de repetição “**para**”, que é iniciado na linha 3, finalizará após o último membro da população ter sua nova localização calculada (linha 26). Após esse encerramento é a vez o primeiro laço de repetição “**para**” ser encerrado na linha 27 e o algoritmo retornar o *id* do veículo escolhido para ter seus recursos alocados no atendimento do serviço (linha 28).

3.2. Definição do Problema

Para o escalonamento de serviços, considera-se um conjunto de serviços S onde $\{s = 1, \dots, n\}$. Assim como consideramos um conjunto de veículos F , onde $\{f = 1, \dots, m\}$. Cada serviço e cada veículo possui três requisitos: Processamento, Armazenamento e Memória. Enquanto cada serviço é a representação de uma requisição recebida pelo *Cluster Head*, cada veículo possui uma quantidade finita dos mesmo recursos que são necessários para a execução de um serviço.

Cada serviço e veículo é identificado unicamente através do seu *id*, além de possuir suas outras três características: *i*) *proc* que representa a quantidade/capacidade de processamento; *ii*) *arm* que indica a capacidade/necessidade do volume de armazenamento e; *iii*) *mem* que representa a quantidade de memória necessária/ociosa.

Para a realização da simulação, algumas premissas tangíveis a problemática de execução de simulação foram delimitadas:

- Cada veículo adicionado a simulação tem seu próprio padrão de mobilidade gerado sinteticamente;
- Os veículos se locomovem segundo esses padrões e respeitam as leis de trânsito impostas pela topologia urbana adotada;
- Todos os veículos são capazes de trocar mensagens diretamente entre si (comunicação veículo-a-veículo - V2V);
- Através da emissão de mensagens *beacon*, os veículos localizam seus pares próximos e se organizam em um *Cluster*, logo a eleição do *Cluster Head* acontece;
- As solicitações de serviços chegam para o CH que monta uma fila de serviços a serem executados pelos membros do *Cluster*, a frequência de entrada desses serviços é uma distribuição de Poisson com $\lambda = 1$;
- O CH executará o MARVEL já que ele detém o conhecimento dos serviços que necessitam de atendimento e das informações sobre os membros do *Cluster* que está gerenciando;

Através da Figura 2, é possível observar uma abstração do cenário descrito, onde os veículos próximos um dos outros se agregam em *Clusters* através da troca de mensagens (*beacons*) enquanto trafegam pela via. Na Figura cada veículo tem seu id_f onde o de menor valor é eleito CH, ou seja, no Cluster 1 por exemplo o CH é o veículo com $id_f = 7$ enquanto os veículos 8 e 13 são membros.

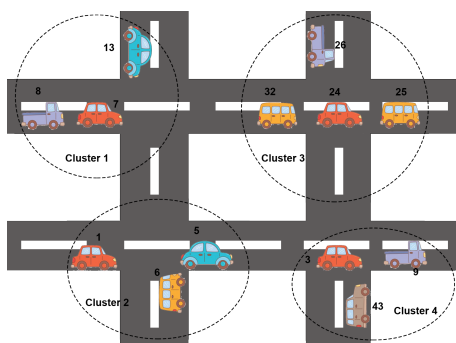


Figura 2. Abstração do cenário de clusterização.

4. Parâmetros, Avaliações e Resultados

Esta seção define os parâmetros utilizados nas simulações, avalia o desempenho do MARVEL em comparação com outros algoritmos da literatura, a fim de otimizar a alocação de recursos e o atendimento de serviços em redes veiculares, bem como apresenta os resultados obtidos nas comparações.

4.1. Parâmetros das simulações

A realização da simulação se deu através do *framework* de simulação veicular (VEINS - *Vehicular in Network Simulation*) [Sommer et al. 2011] em sua versão 5.1 que é o responsável pelo gerenciamento dos protocolos e comunicação de rede. Ele é suportado pelo *framework* de Simulação de Eventos Discretos OMNeT++ [Varga 2010] na sua versão 5.6.2. Também é utilizado o Simulador de Mobilidade Urbana (SUMO - *Simulation of Urban MObility*) [Lopez et al. 2018] na sua versão 0.32.0 para carregar o mapa da simulação e implementar a capacidade de locomoção dos veículos, sinalizações, regras de trânsito, etc. O padrão de mobilidade gerado sinteticamente é realizado através do *script random-Trips.py* disponibilizado pelo próprio SUMO, que considera o mapa utilizado (Brasília) e gera rotas com tempos, origens e destinos aleatórios para cada veículo no cenário.

O mapa escolhido foi da região central do planalto da cidade de Brasília - DF com uma área aproximada de $17,63 \text{ km}^2$. A Figura 3 representa a região selecionada para a simulação.

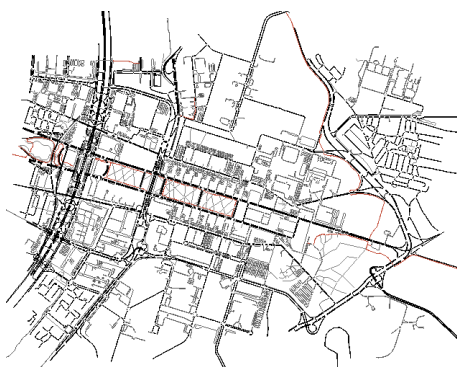


Figura 3. Cenário utilizado na simulação (Mapa da região central do planalto em Brasília - DF).

O protocolo de rede escolhido é o IEEE 802.11p [Arena et al. 2020], desenvolvido especificamente para comunicação em redes veiculares, ele é carregado na simulação através do VEINS. A política de eleição de *Cluster Head* implementada foi o Algoritmo de Menor ID (LID - *Lower ID Algorithm*), que elege o membro com menor id_f como CH [Nguyen et al. 2015].

Foram definidas três diferentes classes de serviços a fim de compor a simulação e tornar o cenário mais fidedigno:

- **Baixo:** São serviços considerados de baixo custo computacional já que necessitam de pouca responsividade do sistema, como monitoramento de tráfego e sugestão de rotas;
- **Médio:** Esses serviços possuem um requisito computacional de nível intermediário pois existe mais urgência em sua execução, como serviços de troca de mensagens, ou necessidade de cruzar referências como um serviço de recomendações ou sugestão de lugares;
- **Alto:** São serviços com alto custo operacional já que necessitam de um contínuo fluxo de grandes informações e processamento constante, como por exemplo serviços de entretenimento de áudio/vídeo.

É possível observar através da Tabela 2 todas as características definidas para a execução das simulações.

Tabela 2. Parâmetros de Simulação

Parâmetro	Valor
Chegada do Serviço	Distribuição de Poisson
Baixo	proc (1 ~ 3) arm (1 ~ 3) mem (1 ~ 3)
Médio	proc (4 ~ 6) arm (4 ~ 6) mem (4 ~ 6)
Alto	proc (7 ~ 10) arm (7 ~ 10) mem (7 ~ 10)
Recursos do Veículo	aleat (1, 10)
Número de Veículos	≈ 1330
Velocidade Veicular Máxima	13,9 m/s
Protocolo de Comunicação	IEEE 802.11p
Distância Máxima de Comunicação	400m
Potência de Transmissão	20mW
Política de Eleição CH	LID
Tipo de cenário	Região Central do Planalto de Brasília - DF
Tamanho do Cenário	4,7Km x 3,75Km
Número de rodadas	33 vezes
Tempo de Simulação	500 segundos

Para comparação do MARVEL foram selecionados outros três algoritmos da literatura utilizados para otimização na tomada de decisão em alocação de recursos em redes veiculares, são eles: Algoritmo de Otimização da Baleia (WOA - *Whale Optimization Algorithm*) [Lieira et al. 2022a], Processo Hierárquico Analítico (AHP - *Analytic Hierarchy Process*) [Pereira et al. 2019] e o Algoritmo de Otimização Ganancioso (Greedy - *Greedy Optimization Algorithm*) [Shalini and Jenicka 2021].

Os resultados expostos na próxima seção são médias calculadas de 33 execuções para cada combinação de parâmetros, com intervalos de confiança de 95% para cada uma das classes de serviços. Para as avaliações, foram utilizadas as seguintes métricas:

- **Serviços atendidos:** são os serviços em que o veículo escolhido pelo MARVEL possui os recursos necessários para realizar o processamento. Para um serviço ser considerado atendido, o CH que enviou a solicitação de processamento tem que obter uma resposta positiva do veículo selecionado para a alocação desse recurso;
- **Serviços não atendidos:** é a junção dos serviços perdidos, aqueles que o CH não recebe uma resposta do veículo selecionado, com os serviços negados, aqueles os quais o veículo escolhido não é capaz de atender ao serviço devido a recursos insuficientes.
- **Recursos alocados:** é a soma das unidades de todos os recursos de processamento, armazenamento e memória dos serviços atendidos. Por exemplo, caso um serviço com requisitos 2 de processamento, 2 de memória e 2 de armazenamento for alocado, o valor de “Recursos alocados” será acrescido em 6 pontos, que é a soma dos três requisitos.

4.2. Resultados

Na figura 4 é possível observar a porcentagem de serviços atendidos para cada um dos algoritmos em cada uma das classes de serviços propostas. Para a classe “Baixa” o MARVEL teve 40,85% de alocação de serviços contra 40,23% do WOA, 34,82% do AHP e 32,24% do GREEDY. Uma pequena vantagem para o MARVEL que conseguiu alocar 1,5% mais serviços que o WOA. Já para a classe “Média”, o MARVEL contou com 7,01% de atendimentos aos serviços contra 6,37% do WOA, 5,53% do AHP e 5,14% do GREEDY. Nesse segundo resultado é possível notar uma diferença maior entre o MARVEL e o segundo colocado, cerca de 10,05%, o que se torna ainda maior com o uso da classe

“Alta” do perfil de simulação com o MARVEL em 1,15% de serviços atendidos enquanto o WOA e AHP empatados com 0,84% e GREEDY com apenas 0,69%. Isso mostra que o MARVEL performou cerca de 36,9% melhor que os demais.

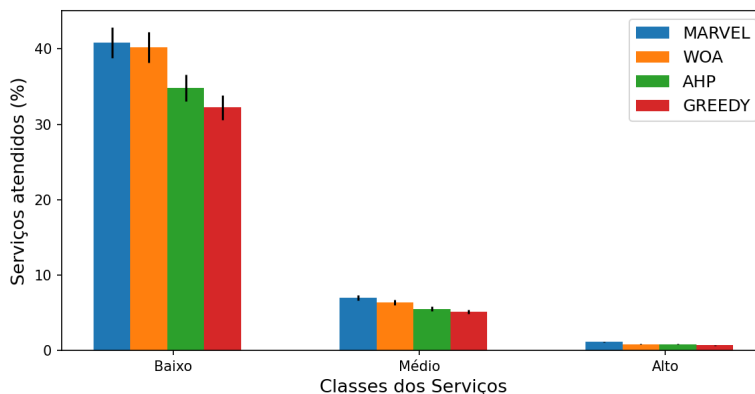


Figura 4. Gráfico de serviços atendidos.

Esse fenômeno pode ser explicado pelo fato do MARVEL ser mais eficiente em alocar recursos para serviços de maior complexidade, que exigem uma quantidade maior de requisitos, que é o caso dos serviços da classe “Alto” onde os requisitos do serviço estão todos entre 7 e 10 pontos enquanto os veículos possuem recursos entre 1 e 10, como a Tabela 2 expõe. Mesmo que os pontos percentuais demonstrem pouca diferença se analisados separadamente, é importante frisar que a avaliação de cenários mais restritivos é essencial para entender o comportamento dos algoritmos em ambientes com serviços de altos requisitos.

A Figura 5 representa a porcentagem de serviços que tiveram problemas durante sua execução, seja porque os veículos escolhidos não tinham recursos suficientes para atender a solicitação, seja por uma falha na comunicação do membro selecionado. Para o perfil “Baixo” o MARVEL deixou de atender cerca de 1,06% menos que o WOA (59,15% do MARVEL contra 59,77% do WOA), ou seja, houve menos escolhas erradas de qual membro forneceria os recursos no atendimento do serviço.

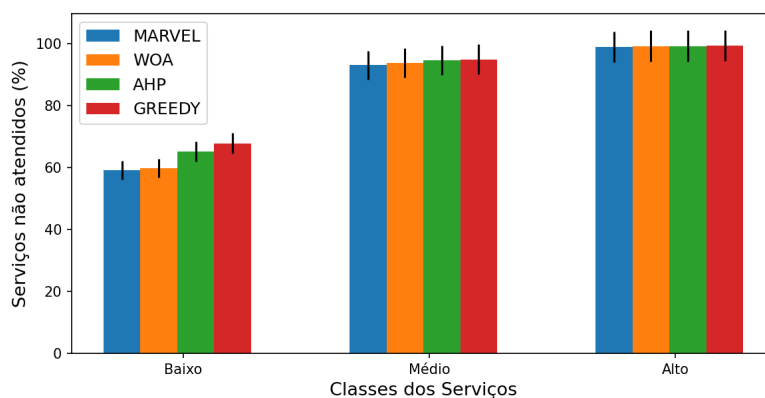


Figura 5. Gráfico de serviços não atendidos.

Já na classe “Média” o MARVEL teve um desempenho mais equilibrado com os seus concorrentes, com uma diferença de aproximadamente 0,68% menos perda que o WOA (92,99% do MARVEL contra 93,63% do WOA), 1,56% menos que o AHP

(94,47%) e 1,97% menos que o GREEDY (94,86%). Para o perfil “Alto” a pouca diferença é mantida já que o MARVEL obteve 98,85% de serviços não atendidos contra 99,16% do WOA e AHP, e 99,31% do GREEDY. Uma diferença de 0,31% menos perca para o MARVEL.

Através da Figura 6 é possível observar a quantidade absoluta de recursos alocados para cada algoritmo e em cada classe de serviços. Visualmente é possível notar uma grande proximidade entre o MARVEL e o WOA na execução da classe “Baixa”, enquanto o MARVEL alocou 7030 unidades de recurso o WOA conseguiu alocar 6991, ou seja, apenas 0,55% a mais, contra 4990 unidades de recursos do AHP e 5464 unidades do GREEDY.

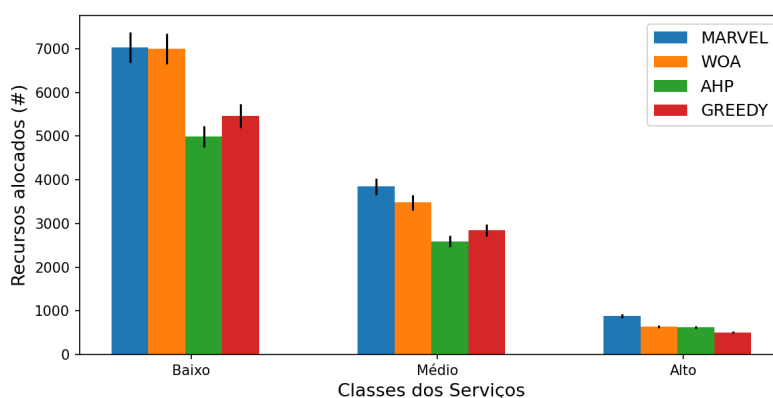


Figura 6. Gráfico de recursos alocados.

Porém, para o perfil “Médio” a diferença cresce, seguindo a tendência observada no gráfico de serviços atendidos. O MARVEL apresenta 3842 unidades de recursos alocadas contra 3484 do WOA, ou seja, cerca de 10,27% mais recursos alocados. Para os demais algoritmos a diferença é ainda maior, 48,11% contra o AHP (que tem 2594 unidades em recursos absolutos) e 34,94% contra o GREEDY (que tem 2847 em recursos absolutos). Já na classe “Alto”, o MARVEL (com 887 unidades de recurso alocadas) tem uma performance 39,24% melhor que o WOA (637 unidades) e 41,01% melhor que o AHP (629 unidades).

A análise dos recursos alocados confirma o indicativo do gráfico de serviços atendidos, o uso do valor de *fitness* como referência na escolha do membro que fornecerá os recursos para atender o serviço é um dos motivos do MARVEL ter melhores resultados que os demais, já que iterativamente a solução será melhorada.

O outro motivo é a capacidade de adaptação que o HHO proporciona ao MARVEL, com duas diferentes estratégias na fase de exploração e quatro diferentes estratégias na fase de aproveitamento para melhor abranger a volatilidade de uma rede veicular, onde se têm inúmeras requisições de serviços a serem atendidas pelos recursos dispersos. Dessa forma, buscou-se analisar desde um cenário menos complexo (classe baixa) a um cenário mais desafiador (classe alta), para conseguir representar a volatilidade das requisições que podem chegar até uma VANET.

5. Conclusão

A problemática da alocação de recursos em VANETs está cada vez mais em evidência dentro do cenário acadêmico. Existem vários desafios a serem resolvidos, como a dinamicidade da rede, baixa escalabilidade, gargalo de rede e alto tempo de resposta. Porém constantes avanços estão sendo feitos e pequenas contribuições ajudam a agregar valor ao todo.

É com esse pensamento que neste trabalho foi proposto o MARVEL, um algoritmo meta-heurístico inspirado na Otimização do Falcão Harris que auxilia o *Cluster Head* da VANET a alocar de maneira mais efetiva o recurso ocioso de outro veículo para atendimento de um serviço. Em comparação com outros algoritmos da literatura, como o WOA, AHP e GREEDY, o MARVEL teve uma melhor performance nos quesitos “serviços atendidos”, “serviços não atendidos” e “recursos alocados”, principalmente durante a alocação para serviços que demandam maior complexidade computacional e que exigem maior quantidade de recursos. Outro ponto observado é o melhor desempenho dos algoritmos que utilizaram a métrica do valor de *fitness* para balizar a escolha do veículo.

Por fim, como direções futuras, pretende-se implementar diferenças na estrutura da VANET, com diferentes formas de eleição do CH e uso de RSUs, a fim de comparar o desempenho geral da rede. Além da otimização do MARVEL para uma melhor performance em cenários de baixa complexidade, adição do tempo de execução e gasto de recursos computacionais em seu funcionamento como meio de avaliação de desempenho. Pretende-se também aumentar o escopo de comparações, ou seja, utilizar mais algoritmos meta-heurísticos durante as comparações além de implementar uma curva de convergência como meio de avaliação deles. Outro ponto a ser desenvolvido é a utilização de malhas viárias de maior complexidade com diferentes parâmetros de quantidade fluante de veículos, outros protocolos de comunicação e até mesmo com limites de velocidades diferentes.

Referências

- Arena, F., Pau, G., and Severino, A. (2020). A review on IEEE 802.11p for intelligent transportation systems. *Journal of Sensor and Actuator Networks*, 9(2).
- Barthelemy, P., Bertolotti, J., and Wiersma, D. S. (2008). A lévy flight for light. *Nature*, 453(7194):495–498.
- FENABRAVE (2022). Informativo emplacements: Resumo mensal dezembro de 2021. Acessado em: 14 de dez. de 2022.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97:849–872.
- Hossain, M. A., Noor, R. M., Yau, K.-L. A., Azzuhri, S. R., Z’Abar, M. R., Ahmedy, I., and Jabbarpour, M. R. (2021). Multi-objective harris hawks optimization algorithm based 2-hop routing algorithm for cr-vanet. *IEEE Access*, 9:58230–58242.
- Husnain, G., Anwar, S., Shahzad, F., Sikander, G., Tariq, R., Bakhtyar, M., and Lim, S. (2022). An intelligent harris hawks optimization based cluster optimization scheme for vanets. *Journal of Sensors*, 2022:6790082.
- Hussain, K., Salleh, M. N. M., Cheng, S., and Naseem, R. (2017). Common benchmark functions for metaheuristic evaluation: A review. In *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION*, volume 1, pages 218–223.
- Li, R., Zhu, P., and Jin, L. (2019). Channel allocation scheme based on greedy algorithm in cognitive vehicular networks. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 803–807.
- Lieira, D. D., Gottsfriz, E. N., Quessada, M. S., Cristiani, A. L., Filho, G. P. R., and Meneguette, R. I. (2022a). Meta-heuristic mechanism based on whale optimization algorithm for tasks allocation in edge computing. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- Lieira, D. D., Quessada, M. S., Cristiani, A. L., De Grande, R. E., and Meneguette, R. I. (2022b). Mechanism for optimizing resource allocation in vanets based on the pso bio-

- inspired algorithm. In *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 283–290.
- 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.
- Marwah, G. P. K., Jain, A., Malik, P. K., Singh, M., Tanwar, S., Safirescu, C. O., Mihaltan, T. C., Sharma, R., and Alkhayyat, A. (2022). An improved machine learning model with hybrid technique in vanet for robust communication. *Mathematics*, 10(21).
- Meneguette, R., De Grande, R., Ueyama, J., Filho, G. P. R., and Madeira, E. (2021). Vehicular edge computing: Architecture, resource management, security, and challenges. *ACM Comput. Surv.*, 55(1).
- Meneguette, R. I. and Boukerche, A. (2017). A cooperative and adaptive resource scheduling for vehicular cloud. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 398–403.
- Meneguette, R. I. and Boukerche, A. (2020). Vehicular clouds leveraging mobile urban computing through resource discovery. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2640–2647.
- Nguyen, V., Kim, O. T. T., Dang, D. N. M., Kim, S. S., and Hong, C. S. (2015). Application of the lowest-id algorithm in cluster-based tdma system for vanets. In *2015 International Conference on Information Networking (ICOIN)*, pages 25–30.
- Peixoto, M., Mota, E., Maia, A., Lobato, W., Salahuddin, M., Boutaba, R., and Villas, L. (2023). Fogjam: A fog service for detecting traffic congestion in a continuous data stream vanet. *Ad Hoc Networks*, 140:103046.
- 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.
- Shalini, T. G. and Jenicka, S. (2021). Weighted greedy approach for low latency resource allocation on v2x network. *Wireless Personal Communications*, 119:2303–2322.
- Sing, R., Bhoi, S., Panigrahi, N., Sahoo, K., Jhanjhi, N., and AlZain, M. (2022). A whale optimization algorithm based resource allocation scheme for cloud-fog based iot applications. In *Electronics 2022*, volume 11, page 3207.
- Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15.
- Varga, A. (2010). *OMNeT++*, pages 35–59. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Wong, W. and Ming, C. I. (2019). A review on metaheuristic algorithms: Recent trends, benchmarking and applications. In *2019 7th International Conference on Smart Computing Communications (ICSCC)*, pages 1–5.
- Yang, X.-S. and Karamanoglu, M. (2020). Chapter 1 - nature-inspired computation and swarm intelligence: a state-of-the-art overview. In Yang, X.-S., editor, *Nature-Inspired Computation and Swarm Intelligence*, pages 3–18. Academic Press.