

TRIAD: Algoritmo de Otimização da Baleia para Decisão de Alocação de Recursos 5G-IoT em Computação de Borda

TRIAD: Whale Optimization Algorithm for 5G-IoT Resource Allocation Decision in Edge Computing

Douglas Dias Lieira, Matheus Sanches Quessada
Departamento de Ciências da Computação e Estatísticas
Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil
douglas.lieira,matheus.quesada@unesp.br

André Luis Cristiani
Departamento de Computação
Universidade Federal de São Carlos - UFSCar
São Carlos, SP, Brasil
andre.cristiani@estudante.ufscar.br

Roger Immich
Instituto Metrópole Digital
Universidade Federal do Rio Grande do Norte (UFRN)
Natal, RN, Brasil
roger@imd.ufrn.br

Rodolfo Ipolito Meneguette
Departamento de Sistemas de Computação - ICMC
Universidade de São Paulo - USP
São Carlos, SP, Brasil
meneguette@icmc.usp.br

Preprint - Resumo—O crescimento massivo na quantidade de dispositivos 5G-IoT circulando no mundo nos últimos anos aumentou a demanda por recursos computacionais. Dessa forma, se torna necessário a busca pelo desenvolvimento de novas soluções ou melhorias nas já existentes. A computação de borda é uma das soluções que têm sido utilizadas para melhorar o atendimento destes tipos de dispositivos. Neste trabalho é proposto um mecanismo que utiliza como base o algoritmo de otimização da baleia para decisão de alocação de recursos 5G-IoT em computação de borda (TRIAD). O TRIAD foi comparado com as técnicas Greedy e Reliable, disponíveis na literatura. Os resultados mostram que o algoritmo proposto teve ótima eficiência no atendimento dos dispositivos, além de negar menos atendimentos e bloquear menos dispositivos durante a busca. O TRIAD, em algumas situações da simulação, atendeu aproximadamente 265% a mais de serviços, negar 56% a menos de atendimentos e bloquear 65% a menos de serviços.

Palavras-Chave—computação de borda, internet das coisas, meta-heurístico, alocação de recursos, algoritmo de otimização da baleia

Abstract—The massive growth in the number of 5G-IoT devices circulating in the world has increased the demand for computing resources in recent years. That way, it is necessary to search for the development of new solutions or improvements to existing ones. Edge computing is one of the solutions that have been used to improve the care of these types of devices. In this work, we proposed a mechanism that uses the whale optimization algorithm for 5G-IoT resource allocation decision in edge computing (TRIAD). The TRIAD was compared with the Greedy and Reliable techniques, available in the literature. The results show that the proposed algorithm had excellent efficiency in the service of the devices, in addition to denying fewer requests and blocking fewer devices during the search. The TRIAD, in some situations of the simulation, served approximately 265% more services, denied 56% less requests and blocked 65% less

services.

Keywords—edge computing, internet of things, meta-heuristic, resource allocation, whale optimization algorithm

I. INTRODUÇÃO

Ao longo dos últimos anos, diversas tecnologias emergentes mudaram o mundo, como, por exemplo, Internet das Coisas (IoT) e a quinta geração de rede móvel (5G). Com estas tecnologias apareceram novas pesquisas relacionadas aos desafios da computação em nuvem para atender as demandas dos dispositivos IoT [1]. Estima-se que cerca de 72% das pessoas de todo o mundo utilizarão dispositivos móveis para acessar a internet em 2025 [2], e que cerca de 2,8 bilhões de pessoas e 1,3 bilhões de dispositivos IoT utilizem conexão 5G para isso [3].

Aplicações e dispositivos de IoT modernos são mais rigorosos quanto às suas conexões e exigem maiores critérios de desempenho, como uma conectividade massiva, baixa latência, alta taxa de transferência, maior segurança e boa taxa de cobertura de comunicação sem fio [4]. Para atender essas demandas, nos últimos anos, diversas tecnologias 5G vem sendo desenvolvidas buscando prover melhores infraestruturas para comunicação entre dispositivos móveis [5], [6]. Com isso, surgiu o termo 5G habilitado para IoT (5G-IoT), sendo uma tecnologia responsável por conectar dispositivos IoT por meio da rede 5G, atendendo essa atual demanda do mercado por serviços sem fio.

Devido à capacidade e recursos limitados, esses aplicativos e dispositivos armazenam, compartilham e buscam recursos para atender suas demandas em serviços de computação em nuvem, que possuem uma grande infraestrutura para gerenciar

esses recursos de uma maneira dinâmica, segura, confiável e econômica [7], [8]. Com o aumento expressivo no número de dispositivos que utilizam estes serviços, surge a necessidade de atender toda a demanda de modo eficiente e com baixa latência, para isso, foi introduzido o conceito de computação de borda (EC, do inglês *Edge Computing*).

A EC foi desenvolvida para lidar com dispositivos que possuem uso intensivo de recursos computacionais, fornecendo uma camada adicional de infraestrutura que distribui parte das máquinas virtuais dos data centers para as extremidades (*edges*) da rede. Isto possibilita alocar, mesmo que parcialmente, o poder computacional nas bordas da rede, permitindo que os recursos sejam processados em pontos mais próximos dos dispositivos solicitantes [9].

Embora esta abordagem reduza significativamente a latência, ela deu origem a um novo problema, também complexo e desafiador, que é lidar com a alocação dos recursos que são disponibilizados pelos serviços nas bordas da rede [10], [11]. Cada *edge* deve possuir mecanismos capazes de prover gerenciamento e compartilhamento de recursos de maneira eficaz [12].

Na prática, problemas de otimização envolvem cálculos complexos e exigem grande esforço computacional para serem resolvidos. Métodos clássicos de resolução exata encontram boas soluções para este tipo de problemas, porém, podem ser inadequados em diversos cenários do mundo real e necessitam de modificações nos códigos para cada cenário que for aplicado [13]–[16]. Diante disso, as técnicas meta-heurísticas, que se baseiam no comportamento natural dos animais ou em elementos da natureza, surgiram com um novo propósito para problemas de otimização: resolver estes problemas de forma eficiente e com baixo custo computacional, encontrando soluções de qualidade com baixo tempo de execução [17].

Desta forma, este trabalho propõe uma solução para a melhoria no gerenciamento e alocação de recursos 5G-IoT em EC. Para isso, criamos um algoritmo baseado no algoritmo meta-heurístico Algoritmo de Otimização da Baleia (WOA) [18], que é baseado no comportamento da baleia jubarte para tomada de decisão de alocação dos recursos nas *edges*, denominado TRIAD (algoritmo de otimização da baleia para decisão de alocação de Recursos 5G-IoT em computação de borda). Os principais objetivos do TRIAD são o de melhorar o aproveitamento dos recursos dos serviços de EC, minimizar a busca pelo melhor local para alocação dos recursos computacionais e atender o maior número de dispositivos em um menor tempo possível. Em comparação com outros algoritmos disponíveis na literatura, os resultados mostraram que nossa abordagem é eficiente e cumpre o objetivo proposto, alocando um maior número de serviços e reduzindo a quantidade de serviços negados e bloqueados.

O restante do artigo está organizado conforme descrito a seguir. A Seção II apresenta os principais trabalhos relacionados com essa pesquisa. A Seção III apresenta o modelo do sistema e descreve detalhadamente o algoritmo proposto. A Seção IV apresenta o cenário, a metodologia adotada para avaliar o algoritmo e os resultados obtidos. Finalmente, a Seção V apresenta as conclusões e direcionamentos para trabalhos futuros.

II. TRABALHOS RELACIONADOS

Os trabalhos abordados nesta seção trabalham com computação móvel de borda [13], [14] e em um ambiente em IoT [15], [16].

Em [13] os autores abordaram um cenário onde vários celulares enviam múltiplas tarefas a um servidor de computação de borda móvel alocando recursos do servidor e canais sem fio. O principal foco era a economia de energia nos celulares, tendo como base a utilização do algoritmo guloso (*greedy*). Os autores propuseram o SMSEF, um algoritmo que seleciona a opção de alocação que irá economizar a maior quantidade de energia. Como forma de avaliação eles consideraram a economia de energia nos celulares em diversos nós e canais da rede. Em comparação ao TRIAD, os autores utilizam serviço de computação de borda, no entanto, consideram como prioridade a economia de energia, e não o atendimento de usuários.

Um mecanismo de alocação de recursos para rede 5G em computação de borda móvel foi desenvolvido em [14]. Os autores propuseram o RELIABLE, um algoritmo de tomada de decisão baseado no método *Analytic Hierarchy Process* (AHP), onde é calculado o fator de importância dos parâmetros com um peso e assim é definido qual o *Mobile Edge Computing* (MEC) escolhido. Os autores avaliaram a quantidade de recursos alocados, serviços negados e serviços bloqueados. O algoritmo avalia as mesmas métricas que o TRIAD, mas a estrutura do algoritmo não permite se adaptar, sem alterações, a cenários diferentes.

Um algoritmo heurístico baseado no WOA foi proposto em [15]. Foi tratado o problema de alocação de recursos e agendamento em IoT (IRAS) reduzindo o custo total de comunicação entre recursos e gateways. O algoritmo proposto foi comparado com dois outros algoritmos, o algoritmo genético (GA) e o economia de pesquisa para alocação de recursos de IoT (SEIRA) e se mostrou eficaz na redução do custo e tempo de comunicação entre os recursos. Os autores também utilizam o WOA como base, mas consideram o custo e tempo de comunicação, enquanto o TRIAD é focado no atendimento de usuários.

Em [16] os autores propuseram uma abordagem de alocação de recursos baseada em grafo para gerenciamento de desastres em ambientes IoT. Alguns dos problemas considerados como desastres pelos autores são o restabelecimento da rede para comunicação, fornecimento de suprimentos médicos, resgate e recuperação. O cenário é dividido em tarefas e os recursos necessários para suprir essas tarefas. O algoritmo proposto foi comparado com o *greedy* e foram avaliadas questões como justiça e tempo de execução. Em comparação ao TRIAD, o algoritmo utiliza grafos para alocação de recursos IoT, mas não tem o desafio de computação urbana e computação de borda e a utilização.

A diferença do TRIAD em relação aos trabalhos relacionados é que ele trata o problema de alocação de recursos IoT em computação de borda, dentro um cenário de computação urbana. O algoritmo supera o desafio de muitos dispositivos requerendo recursos escassos de ECs em um curto espaço de tempo, sem precisar de modificações caso mude a quantidade de ECs e/ou recursos.

em serviços de computação de borda. Ele começa com a inicialização da população com os recursos das ECs (linha 1) e o cálculo do *fitness* (linha 2), que calcula a aptidão de cada indivíduo. Assim, é verificado qual o melhor *fitness* neste momento (linha 3). Inicia-se então uma estrutura de repetição com a quantidade máxima de iterações definida para o algoritmo (linha 4). Neste algoritmo, a variável *num_iteracao* foi definida para 30 iterações. Na linha 5 a estrutura de repetição é utilizada para atualizar as posições das ECs e realizar a aproximação em cada iteração. Os parâmetros responsáveis pelas movimentações são atualizados na linha 6.

A variável *p* representa a probabilidade de utilização dos mecanismos de encolhimento ou atualização da espiral. Se *p* for menor que 0.5, é realizado o mecanismo de encurtamento, sendo que, quando o coeficiente *A* é menor que 1, é realizada a atualização seguindo a melhor posição atual da busca aleatória (linha 9). Se *A* for maior ou igual a 1, a atualização é feita pela escolha aleatória de uma EC (linhas 11 e 12). Se *p* for maior ou igual a 0.5, é executado o mecanismo de atualização por espiral (linha 15). A cada iteração, são atualizadas as posições, *fitness* e coeficientes (linhas 18, 19 e 20). Após terminar o número máximo de iterações definido no código, é retornada a EC com a melhor aptidão (linha 22).

Algoritmo 1 TRIAD

portuguese

```

1: inicializaapopulaçãodeECs
2: calculaofitness
3: identificaamelhorECatual(ED)
4: enquanto t < num_iteracao faça
5:   for cadaedge faça
6:     atualizaa, A, C, l, p
7:     se p < 0.5 então
8:       se A < 1 então
9:         atualizaposiçãocomEq.1
10:      senão se A ≥ 1 então
11:        selecionaumaECaleatória
12:        atualizaposiçãocomEq.6
13:      fim se
14:    senão se p ≥ 0.5 então
15:      atualizaposiçãocomEq.5
16:    fim se
17:  fim for
18: calculaofitness
19: atualizaEDsetiveralgummelhor
20: t = t + 1
21: fim enquanto
22: retorna ED

```

Para cálculo do *fitness*, foi utilizada a função *benckmark* multimodal de Ackley (Equação 8). Optou-se por esta função, pois está no grupo das funções que apresentaram melhores resultados no trabalho de [18].

$$\begin{aligned}
Ackley(p) = & -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) \\
& - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) \\
& + 20 + e
\end{aligned} \quad (8)$$

IV. AVALIAÇÃO DO MECANISMO PROPOSTO

Nesta seção são apresentadas os cenários, as configurações da simulação, as técnicas de comparação, as métricas utilizadas e uma análise acerca dos resultados obtidos.

A Fig. 2 apresenta o cenário utilizado para a simulação do TRIAD. Neste cenário são apresentadas 4 *Roadside Units* (RSUs), que são unidades de acostamento estáticas, interligadas através de uma conexão cabeada. Cada RSU possui um equipamento de computação de borda instalado. As ECs compartilham seus recursos com os dispositivos IoT, que estão na área de alcance das RSUs, por meio de rede sem fio, utilizando comunicação 5G. Os dispositivos IoT enviam a solicitação de recurso às ECs, que executam o TRIAD para saber em qual EC alocar os recursos.

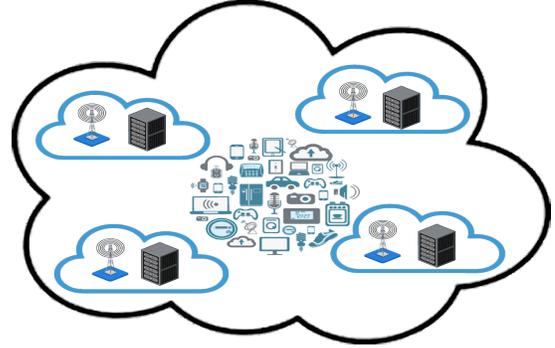


Fig. 2. Modelo de sistema com 4 RSUs do TRIAD.

Os *user equipment* (UEs) representam dispositivos IoT, tais como *smartphones*, *tablets*, *notebooks*, relógios inteligentes, equipamentos veiculares, sensores, entre outros. Cada dispositivo DU_i possui uma identificação ($i \in [1, n]$) e um conjunto de recursos computacionais que correspondem a largura de banda, memória, processamento e tempo, denotados como $DU_i = \{b, m, p, t\}$. As 4 ECs são equipamentos que gerenciam e compartilham os recursos computacionais de acordo com a disponibilidade. Ao receber uma requisição, as *Edges* executam o TRIAD para decidir a EC que irá armazenar os recursos do dispositivo solicitante. No cenário apresentado, o tráfego de requisições para entrada na rede são feitas de forma heterogênea, utilizando a distribuição de Pearson 3, que é uma distribuição gama avançada.

Para a simulação foi considerado o cenário acima, ou seja, um ambiente que seja atendido por 4 RSUs e cada uma delas possui um equipamento de computação de borda. Cada EC compartilha, com os dispositivos dos usuários, 100% de seus recursos computacionais através de uma rede 5G. Os dispositivos são gerados considerando serviços de consumo baixo, como mensagem de texto e multimídia. Portanto, os recursos são gerados com valores aleatório de 1% a 10%. Quando um dispositivo entra na rede, é decrementado da capacidade total da EC. Quando sai, os recursos ficam disponíveis novamente.

Cada etapa da simulação foi executada por 1200, 2400, 3600, 4800 e 6000 segundos, gerando as entradas de 207, 412, 641, 861 e 1072 dispositivos, respectivamente. As entradas e saídas de um dispositivo na rede é feita de forma heterogênea

utilizando a distribuição de Pearson 3. As simulações foram executadas 33 vezes e aplicado um intervalo de confiança de 95%.

Para efeitos de comparação, utilizou-se as técnicas Greedy e Reliable. A técnica Greedy (gulosa) seleciona a primeira EC que é capaz de atender os recursos do dispositivo que está entrando na rede e aloca naquela mesma [13]. A técnica Reliable, baseia-se no Processo de Hierarquia Analítico (AHP, do inglês, *Analytic Hierarchy Process*), que utiliza um fator de influência para cada parâmetro com seus graus relativos de importância para tomada de decisão [14]. Os algoritmos TRIAD, Greedy e Reliable foram desenvolvidos utilizando a linguagem de programação Python (versão 3.6.9) e adequados para o cenário da simulação.

As três métricas utilizadas para avaliar as técnicas foram a quantidade de serviços atendidos, onde foi considerado o número de dispositivos que tiveram seus recursos alocados em alguma EC; quantidade de serviços negados, que aponta quantos dispositivos não conseguiram alocar recursos em nenhuma EC; e quantidade de serviços bloqueados, que contabiliza quantas vezes as ECs não tiveram disponibilidade para alocar um dispositivo. Além disso, também é apresentado um gráfico de convergência da técnica TRIAD, em que mostra a curva necessária até chegar no melhor *fitness* durante as simulações.

Na Fig. 3 pode-se observar a quantidade de serviços atendidos em cada técnica. O TRIAD conseguiu atender mais dispositivos em todas as configurações, tanto a com menor número quanto a com maior número de dispositivos. Em relação ao Greedy, ele evoluiu de 29.6% a mais de recursos alocados na comparação com menos dispositivos, e chegou a quase 3 vezes mais alocações na comparação com mais dispositivos. Em relação ao Reliable, o TRIAD manteve entre 1.5% e 12.5% de mais recursos alocados, considerando da configuração com mais dispositivos para a menor. A vantagem de alocação do TRIAD se deve ao fato dele ter uma estrutura de repetição para buscar sempre a EC com o melhor *fitness* em relação a sua quantidade de recursos.

Os serviços negados são apresentados na Fig. 4. O TRIAD negou em média 54 serviços na primeira simulação, enquanto o Reliable e o Greedy negaram 62 e 81, respectivamente. A diferença entre o Greedy aumentou consideravelmente, sendo que o TRIAD negou em média 267 e 472 dispositivos na terceira e na quinta comparação, enquanto o Greedy negou 443 e 842 dispositivos, respectivamente. Em relação ao Reliable, o TRIAD manteve-se sempre abaixo ou igual no número de serviços negados. A vantagem obtida pelo TRIAD é por conseguir se adaptar ao cenário e buscar sempre uma ótima solução para que a requisição de recursos do dispositivo não seja negada.

Na Fig. 5 é possível analisar os serviços bloqueados. Nesta métrica, o TRIAD apresentou novamente boa vantagem em relação ao Greedy durante as 5 comparações, bloqueando em média apenas 289, 767, 1344, 1873 e 2363 vezes as tentativas de alocação de recursos, enquanto o Greedy negou 496, 1188, 2049, 2896 e 3699 vezes, respectivamente. O Reliable bloqueou em média 383, 833, 1373, 1905 e 2348 vezes as tentativas, bloqueando menos que o TRIAD apenas na última configuração. Isso mostra que o TRIAD encontra a melhor solução em menos tempo que o Greedy e Reliable,

pois sempre analisa os recursos das ECs antes de realizar a tentativa de alocação.

Por fim, na Fig. 6 é apresentada a média da curva de convergência do algoritmo TRIAD. A curva mostra evolução da média dos *fitness* de cada iteração durante as simulações. Pode-se observar que iniciou-se com a média sendo um pouco maior que 10 na primeira iteração. Na décima primeira iteração o *fitness* já foi abaixo de 1. E a partir da décima sétima iteração o *fitness* foi de 0.22, bem próximo do valor da última iteração, que foi de 0.15. Com isso, pode-se observar que quase na metade das iterações o TRIAD chegou próximo do *fitness* ideal.

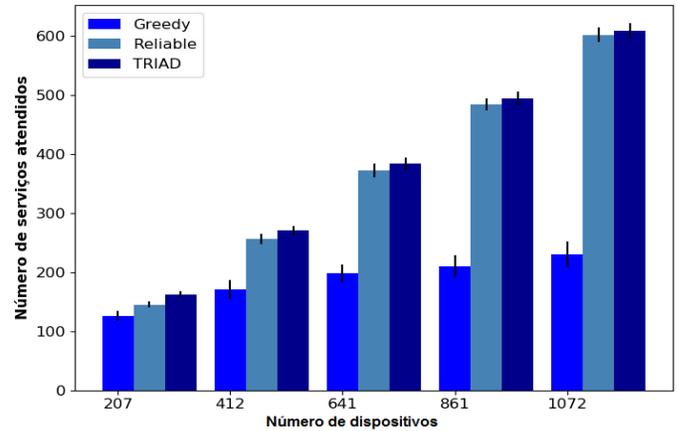


Fig. 3. Serviços de dispositivos alocados.

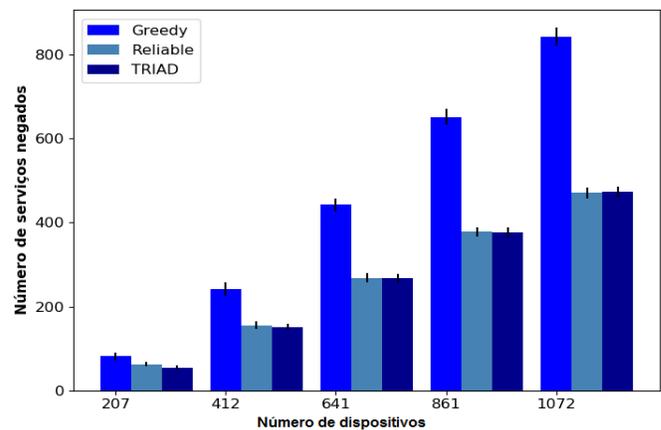


Fig. 4. Serviços de dispositivos negados.

Com os resultados apresentados, é possível ver que o TRIAD consegue otimizar a utilização dos recursos computacionais disponibilizados pelas ECs, conseguindo atender mais dispositivos de usuários e de maneira mais eficiente do que o Greedy e o Reliable, contribuindo assim para a experiência do usuário durante a utilização dos serviços.

V. CONCLUSÃO

Com a grande quantidade de dispositivos 5G-IoT que temos hoje em dia, é necessário buscar otimizar os serviços

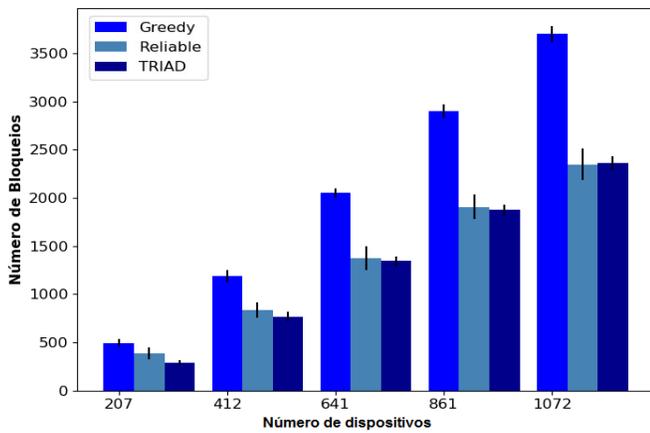


Fig. 5. Tentativas de alocação bloqueadas.

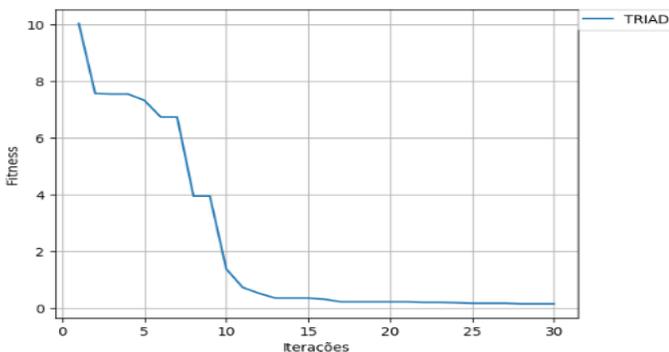


Fig. 6. Curva de convergência do TRIAD.

disponibilizados aos usuários, para melhorar a qualidade do serviço e a experiência do usuário. Neste trabalho, utilizou-se a técnica meta-heurística WOA, em um ambiente com ECs que gerenciam a alocação de recursos entre si, para decidir qual a melhor EC para alocar os recursos solicitados pelos dispositivos. O mecanismo TRIAD foi desenvolvido e comparado com as técnicas de alocação já utilizadas na literatura, como o Greedy e o Reliable. As métricas comparadas foram as quantidades de dispositivos atendidos, negados e bloqueados. Para a simulação, considerou-se desde um ambiente com um número menor de dispositivos, evoluindo para uma quantidade maior. Os resultados demonstram que o TRIAD consegue maximizar a quantidade de serviços atendidos de maneira eficiente, com até 265% em uma das comparações, bloqueou até 65% menos de requisições, além de minimizar em até 56% a quantidade de dispositivos que tem suas requisições negadas. Assim, o TRIAD demonstra ser mais eficiente do que as técnicas comparadas para alocação de dispositivos IoT em serviços de computação de borda, com recursos mais limitados.

Como trabalhos futuros pretende-se comparar o TRIAD com outras técnicas meta-heurísticas, além de utilizar outras funções *benchmark* para calcular o *fitness* e realizar as comparações com vários números de iterações para analisar a curva de convergência. Além disso, também projeta-se utilizar

um simulador de rede para os testes dos algoritmos.

ACKNOWLEDGMENT

Os autores agradecem ao CNPq pelo apoio financeiro (407248/2018-8 e 309822/2018-1), e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro (2020/07162-0) em sua pesquisa.

REFERÊNCIAS

- [1] D. D. Lieira, M. S. Quessada, A. L. Cristiani, and R. I. Meneguette, "Resource allocation technique for edge computing using grey wolf optimization algorithm," in *2020 IEEE Latin-American Conference on Communications (LATINCOM)*, 2020, pp. 1–6.
- [2] Bankmycell. (2021, jan) How many smartphones are in the world? [Online]. Available: <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>
- [3] Ericsson. (2021, Jan) Cellular networks for massive iot. [Online]. Available: <https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-networks-for-massive-iot-enabling-low-power-wide-area-applications>
- [4] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rana, "The internet of things, fog and cloud continuum: Integration and challenges," *Internet of Things*, vol. 3-4, pp. 134 – 155, 2018.
- [5] S. Li, L. D. Xu, and S. Zhao, "5g internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2452414X18300037>
- [6] E. S. Gama, R. Immich, and L. F. Bittencourt, "Towards a multi-tier fog/cloud architecture for video streaming," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 13–14.
- [7] D. C. Marinescu, *Cloud Computing: Theory and Practice*, second edition ed., J. Grover, P. Vinod, and C. Lal, Eds. Morgan Kaufman - Elsevier, 2018.
- [8] R. I. Meneguette and A. Boukerche, "Peer-to-peer protocol for allocated resources in vehicular cloud based on v2v communication," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.
- [9] R. I. Meneguette and A. Boukerche, "A cooperative and adaptive resource scheduling for vehicular cloud," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 2017, pp. 398–403.
- [10] X. Li, Y. Liu, H. Ji, H. Zhang, and V. C. M. Leung, "Optimizing resources allocation for fog computing-based internet of things networks," *IEEE Access*, vol. 7, pp. 64 907–64 922, 2019.
- [11] R. I. Meneguette, A. Boukerche, and A. H. M. Pimenta, "Avarac: An availability-based resource allocation scheme for vehicular cloud," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3688–3699, 2019.
- [12] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware resource allocation for edge computing," in *2017 IEEE International Conference on Edge Computing (EDGE)*, 2017, pp. 47–54.
- [13] F. Wei, S. Chen, and W. Zou, "A greedy algorithm for task offloading in mobile edge computing system," *China Communications*, vol. 15, no. 11, pp. 149–157, 2018.
- [14] R. S. Pereira, D. D. Lieira, M. A. C. d. Silva, A. H. M. Pimenta, J. B. D. da Costa, D. Rosário, L. Villas, and R. I. Meneguette, "Reliable: Resource allocation mechanism for 5g network using mobile edge computing," *Sensors*, vol. 20, no. 19, 2020.
- [15] A. Sangaiah, A. Hosseinabadi, M. Shareh, S. B. Rad, A. Zolfagharian, and N. Chilamkurti, "Iot resource allocation and optimization based on heuristic algorithm," in *Sensors 20*, no. 2, 2020, p. 539.
- [16] J. S. Kumar and M. A. Zaveri, "Graph-based resource allocation for disaster management in iot environment," ser. AWICT 2017. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: <https://doi.org/10.1145/3231830.3231842>
- [17] S. Nasmachnow, "An overview of metaheuristics: accurate and efficient methods for optimisation," *International Journal of Metaheuristics*, vol. 3, no. 4, pp. 320–347, 2014.
- [18] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0965997816300163>
- [19] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: Whale optimization algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1–24, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210650218309350>