A Cost-Efficient Resource Allocation for Fog Computing with Users and Providers Perspective

Joao Bachiega Jr
University of Brasilia, Brazil
joao.bachiega.jr@gmail.com

Breno Costa
University of Brasilia, Brazil
brenogcosta@gmail.com

Michel J. F. Rosa
University of Brasilia, Brazil
michelj@gmail.com

Leonardo R. Carvalho
University of Brasilia, Brazil
leouesb@gmail.com

Marcelo A. Marotta
University of Brasilia, Brazil
marcelo.marotta@unb.br

Aleteia Araujo
University of Brasilia, Brazil
The University of Melbourne, Australia
aleteia@unb.br

Rajkumar Buyya
The University of Melbourne, Australia
rbuyya@unimelb.edu.au

***Abstract*—Fog computing paradigm allows allocating computational resources and services at the edge of the network, closer to the end devices and users, complementing Cloud computing. Fog nodes comprise computational capabilities (such as CPU, memory, and disk) and behavioral characteristics (such as availability, scalability, and mobility). It should also be considered that while the end-user's goal is to obtain the best available resources, the service provider's concern is meeting user requests using minimal resources. Furthermore, it is crucial to consider the financial cost of each resource in both scenarios. Taking all of this into account, obtaining a cost-effective optimized resource allocation is viewed as a challenge. In this article, we propose an efficient algorithm for resource allocation, considering both the provider's and the end-user's perspectives, exploiting computational capabilities, behavioral characteristics, and the financial cost. The tests were carried out in both real and simulated environments and demonstrated that our proposal complies with the resource allocation needs for Fog computing and has a better performance compared to a similar solution.**

***Index Terms*—Fog computing, resource allocation, provider's perspective, end-user's perspective, cost-efficiency**

I. Introduction

Fog computing has emerged as a promising solution for Cloud computing bottlenecks with regard to high latency and response time [1]. To address these bottlenecks, Fog computing is designed as a highly geographically distributed and heterogeneous network composed of low computational capacity devices with different setups closer to end-users. The devices can remotely process the workloads initially meant for the Cloud with smaller response time and latency. This network has been used in dynamic contexts, with devices sharing and competing for computational resources, such as the Internet of Things (IoT). This competition creates uncertainty for resource management in Fog computing environments, leading to possible overloaded or underused nodes, causing unnecessary power consumption, service unavailability, high response time, and decreased reliability [2]. In such a scenario, resource allocation to determine suitable resources that satisfy a required workload remains a challenge [3].

A relevant concept in a Fog computing environment is the Fog node, defined as any computational device with system and hardware resources added to high communication capability [4]. A fog node has computational capabilities, such as CPU, storage, memory, and other hardware attributes that can be measured. In addition, reflecting the dynamicity and unpredictability of Fog computing, the Fog node is also made up of some behavioral characteristics, such as availability, interoperability, and mobility [5].

The end-user and service provider perspectives must be considered when implementing Fog computing management. On the one hand, end-users always want to obtain the best resources available, those with greater computational power and higher values for behavioral characteristics. On the other hand, service and infrastructure providers are interested in delivering minimal resources to avoid unnecessary costs or situations where resources are unavailable to other users [6]. Although it is desirable to consider the behavioral characteristics, their degradation is not an obstacle to the execution of the application. In this sense, most Fog computing resource allocation solutions focus exclusively on hardware attributes without considering behavioral characteristics [7] [8]. Despite this trend, some publications consider only the user's perspective [9] [10]. To the best of our knowledge, considering both perspectives, i.e., balancing the needs of end-users and providers with the computational capabilities and behavioral characteristics of the Fog node, is a matter of investigation [6].

This work presents an algorithm for resource allocation in Fog computing, considering hardware attributes and behavioral characteristics. Thus, the main contributions of this work are:

- A Fog node cost-efficient decision-making algorithm that considers both hardware attributes and behavioral characteristics of available nodes, considering the perspectives of end-users and providers. This model allows parameters to be weighted and balanced to create a ranking of available Fog nodes that comply with the user's request;
- A prototype of a system that efficiently uses the proposed algorithm, being scalable in terms of the number of nodes in the environment and adherent to the Fog aspects;
- An analysis of the proposed algorithm efficiency performed in a real test environment, validating the system

model to solve the problem of resource allocation in a Fog computing environment;
- An extensive experimental analysis of the prototype in a simulated scenario, validating the proposed algorithm performance in a high number of devices scenario;
- A comparison of the proposed algorithm with another resource allocation algorithm from the literature.

The rest of the paper is organized as follows. The system model is presented in Section II. Section III defines the resource allocation problem. An algorithm for efficient resource allocation is proposed in Section IV. In Section V, a motivating scenario is used to explain the usage of the algorithm. The performance evaluation of the proposal is presented in Section VI. Section VII presents some related work. Finally, Section VIII gives the conclusion and opportunities for future work.

II. System Model

This section presents the system model and further definitions for the resource allocation method discussed in this work. In a Fog environment, there is a set of Fog nodes, where each $f_i \in \mathcal{F} = \{f_1, f_2, ..., f_F\}$. Computational capabilities ($\mathcal{C}$) represent minimal computational requirements for task execution, such as CPU, memory, disk, etc. They are represented by the vector $\mathcal{C} = \{c_1, c_2, ..., c_C\}$, where C is the set cardinal. In contrast, behavioral characteristics ($\mathcal{B}$) are characteristics the user wants but is not required to operate. Not meeting the behavioral characteristics does not prevent the application from being executed, but can significantly reduce the user's experience. Security, mobility, scalability, and reliability are examples of behavioral characteristics. They are represented by $\mathcal{B} = \{b_1, b_2, ..., b_B\}$, where B is the cardinal of the set.

Taking into account both types of attributes, each Fog node is determined as a tuple $f_i = \{\mathcal{C}_i \cup \mathcal{B}_i\}$; i.e., $f_i = \{c_{i1}, ..., c_{iC}, b_{i1}, ..., b_{iB}\} \mid i \in \{1, 2, ..., F\}$. The total number of attributes (capabilities and behavioral characteristics) of each Fog node is represented by $N = C + B$. Considering that each column represents an attribute, a Fog node matrix can be denoted by $\{a_{mn} \in A \mid 1 \leq m < F; 1 \leq n < N\}$. Therefore, Matrix A can be represented as:

$$A = \begin{vmatrix} c_{11} & c_{12} & ... & c_{1C} & b_{11} & b_{12} & ... & b_{1B} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{F1} & c_{F2} & ... & c_{FC} & b_{F1} & b_{F2} & ... & b_{FB} \end{vmatrix}$$

Also, the Matrix A can be contracted to:

$$A = \begin{vmatrix} \mathcal{C}_1 & \cup & \mathcal{B}_1 \\ \mathcal{C}_2 & \cup & \mathcal{B}_2 \\ \vdots & \vdots & \vdots \\ \mathcal{C}_F & \cup & \mathcal{B}_F \end{vmatrix}$$

And, finally:

$$A = \begin{vmatrix} f_1 \\ f_2 \\ \vdots \\ f_F \end{vmatrix}$$

Since each attribute in ($\mathcal{C}$) and in ($\mathcal{B}$) can have different units and scales, it is necessary to normalize the Matrix A to allow comparisons and operations between its elements. First, using an adapted vector normalization technique [11], the factor (Υ) value for each attribute is calculated by the Equation 1:

$$\Upsilon_n = \sqrt{\left(\sum_{m=1}^{F} (a_{mn})^2 \right)} \tag{1}$$

Furthermore, each *Fog node* must have an assigned financial cost ($\mathcal{Q}$). For this, another matrix is constructed:

$$Cost = \begin{vmatrix} q_{f_1} \\ q_{f_2} \\ \vdots \\ q_{f_F} \end{vmatrix}$$

On the requester side, values ($\mathcal{V}$) are used to indicate the required amount of resources for each attribute (e.g., the value "8" is required for vCPUs), and weights ($\mathcal{W}$) are used to represent the importance that the requester has defined for each attribute in percentile form, such as "15%" for Memory and "30%" for Storage. A Mean Opinion Score (MOS) [12] from 1 to 5 indicates the value of the behavioral characteristics, in which 1 represents the lowest importance and 5 is the highest importance. When requesting a resource, the user informs the desired values ($\mathcal{V}$) and the weights ($\mathcal{W}$) of the computational capabilities and behavioral characteristics, as well as the maximum cost he or she is willing to pay for resource allocation ($price$). The quantity of necessary Fog nodes is also informed, represented by r. Consequently, the user request is defined as $\mathcal{R} = \mathcal{V} \cup \mathcal{W} \cup price \cup r$ where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$, and $\mathcal{W} = \{w_1, w_2, ..., w_N\}$, $price$ brings the maximum acceptable cost of the resource and r indicates the number of *Fog nodes* needed. Similar to what happens in Cloud computing, for this proposal, it was considered that once the resource is allocated, it remains available to the requester until the requester releases it. Finally, as a data property, it is essential to ensure that the sum of all weights informed equals 100%, so $\sum \mathcal{W} = 1$.

Taking into account the values ($\mathcal{V}$) and the weights ($\mathcal{W}$) informed by the requester, and using the normalization factor Υ presented in Equation 1, the normalization of each attribute in Matrix A is executed, generating a value P, as calculated in Equation 2.

$$P_{mn} = \left(\frac{a_{mn} - v_n}{\Upsilon n \quad w_n} \right) \tag{2}$$

The sum of the normalized values obtained with Equation 2 for $\mathcal{C}$ creates a variable called Ω, presented in Equation 3:

$$\Omega_m = \sum_{n=1}^{C} P_{mn} \tag{3}$$

The sum of the normalized values obtained with Equation 2 for $\mathcal{B}$ creates a variable called Ψ, obtained with Equation 4:

$$\Psi_m = \sum_{n=C+1}^{N} P_{mn} \tag{4}$$

Since the normalized values Υ of $\mathcal{B}$ can be negative, to estimate the distance to the user's informed value, the sum of the module of normalized values obtained with Equation 2 for $\mathcal{B}$ creates a variable called ζ, presented in Equation 5:

$$\zeta_m = \sum_{n=C+1}^{N} |P_{mn}| \tag{5}$$

At this point, two perspectives can be developed. The first, called *USR Solution*, considers the user's perspective, in which the best *Fog node* is sought, that is, the one that has the maximum values of each attribute and that has the lowest cost, ensuring that the cost is lower than the cost informed by the user. The provider's perspective is the second, called *PRV Solution*. That is, the selected *Fog node* is the one that, while meeting the user's request, has the minimum available values for its attributes and, in addition, has the highest cost under the limit informed by the user. The provider aims to achieve the highest possible profit, meeting all user requirements.

III. Problem Definition

Resource allocation is a step in resource management that seeks the best available computing resources necessary to run an application in the Fog computing environment, aiming to meet Quality of Service (QoS) [13]. In this work, the resource is a Fog node (f), and it is composed of computational capabilities ($\mathcal{C}$) and behavioral characteristics ($\mathcal{B}$), as well as financial cost ($\mathcal{Q}$). When a user makes a request, it is expected that at least one Fog node (f) meets the minimum computational capability requirements ($v_1..v_C$), in addition to being cheaper than the cost limit. It is acceptable for the (f) available to be as close as possible to meeting all the behavioral characteristics required ($v_{C+1}..v_N$) even if it does not fully meet them. Therefore, it was considered that meeting the requirements for computational capabilities ($v_1..v_C$) and respecting the informed cost limit ($price$) are essential to run an application. However, meeting the requirements for behavioral characteristics ($v_{C+1}..v_N$) is only desirable.

With this, the objective is to find, among all Fog nodes available in the Matrix A, those that meet all capabilities requirements, the cost limit, and, within this subset, the *Fog node* that comes closest to meeting the requirements of behavioral characteristics, considering the weights informed for each attribute.

Therefore, the restrictions for this problem are that all attributes of capabilities ($\mathcal{C}$) must meet the user requirements, as indicated by Equation 6. To model this selection, x_m is a binary variable that will obtain the value 1 when the Fog node m is selected and the value 0 otherwise.

$$\begin{gathered} x_m \leftarrow f_{mn} \geq \mathcal{V}_n \\ 1 \leq n \leq C;\ 1 \leq m \leq F \end{gathered} \tag{6}$$

When a capability attribute must be lower than the value requested by the user (for example, latency is considered better when it is at the lowest value), the inverse of Equation 6 is executed.

Furthermore, the financial cost ($\mathcal{Q}$) must also be equal to or lower than that requested by the user, as indicated in Equation 7:

$$\begin{gathered} q_{f_m} \leq price \\ 1 \leq m \leq F \end{gathered} \tag{7}$$

Another restriction is that the number of *Fog nodes* selected cannot be greater than the quantity requested, as indicated by Equation 8:

$$\begin{gathered} \sum_{m=1}^{F} x_m = r \\ x_m = \{0, m\} \mid \forall m \in F. \end{gathered} \tag{8}$$

A. Objective Function

As mentioned, one of the differences of this proposal is that it has two distinct solutions for user and provider perspectives, called *USR Solution* and *PRV Solution*, respectively. Considering that the user always intends to obtain the best available resource and that has the best cost-benefit ratio, the *USR Solution* is defined as indicated in Equation 9:

$$\begin{gathered} min \sum_{m=1}^{F} \left(\frac{(\Omega_m + \zeta_m) x_m}{q_m}\right) \\ \text{s.t. (6), (8)} \end{gathered} \tag{9}$$

On the provider side, the objective is to find the *Fog node* that delivers all requested values but considers a minimum set of resources, avoiding waste. Thus, the *PRV Solution* is defined as shown in Equation 10:

$$\begin{gathered} max \sum_{m=1}^{F} \left(\frac{(\Omega_m + \zeta_m) x_m}{q_m}\right) \\ \text{s.t. (6), (8)} \end{gathered} \tag{10}$$

In Equation 9, which represents the user's perspective, the resource to be selected will be the one with the lowest cost-benefit ratio, whose capabilities attributes meet all user requirements, and whose behavioral characteristics are the most appropriate possible, with maximum values for each, given the weights informed by the user. On the other hand, in Equation 10, which represents the provider's perspective, the resource to be selected will be the one with the highest cost-benefit ratio, whose capabilities attributes meet all the user requirements and also whose behavioral characteristics are as appropriate as possible, given the weights assigned by the user, with minimum values for each attribute. Likewise, the fairest choice is guaranteed: the resource with the shortest distance from the user's request among all available *Fog nodes*.

IV. Proposed Algorithm

When requesting a *Fog node* f in a *Fog computing* environment, the user informs the values ($\mathcal{V}$) he needs for computational capabilities ($\mathcal{C}$) and behavioral characteristics ($\mathcal{B}$), as well as the maximum value he is willing to pay for the resource ($price$). The desired number of Fog nodes (r) is also informed as input. At this point, weights ($\mathcal{W}$) must be assigned to each attribute. The function of weight precisely directs the choice toward the most relevant attribute from the user's perspective.

In our proposal, the user's perspective aims to select the best available resource with the most substantial computational power and lowest cost. From the provider's perspective, the resource to be delivered must be as close as possible to the requested values, with minimum values for each attribute among the available resources and a higher cost.

Therefore, a solution based on the Multiple-Criteria Decision-Making (MCDM) method, Algorithm 1, is proposed. It chooses the best *Fog node* among all available that meet the user requirements, considering the end-user and service provider's perspectives.

Algorithm 1: USR Solution and PRV Solution for *Fog node* selection

Data: $\mathcal{C}$, $\mathcal{B}$, Υ, Matrix A, Matrix $Cost$, $\mathcal{V}$, $\mathcal{W}$, $price$, r
Result: f_u, f_p

1 $X \leftarrow TRUE$
2 **for** $i \leftarrow 1$ **to** F **do**
3   **for** $j \leftarrow 1$ **to** $(C + B)$ **do**
4     $P[i][j] \leftarrow (A[i][j] - \mathcal{V}[j])/(\Upsilon[j] * \mathcal{W}[j])$;
5   **if** $A[i][j] < V[j]$ AND $j \leq C$ **then**
6       $X[i] \leftarrow FALSE$
7   **end**
8   **end**
9   **if** $Cost[i] > price$ **then**
10     $X[i] \leftarrow FALSE$
11   **end**
12 **end**
13 calculate Ω;
14 calculate ζ;
15 **for** $i \leftarrow 1$ **to** F **do**
16   **if** $X[i]$ AND $f_u \leq r$ **then**
17     $f_u \leftarrow min((\Omega[i] + \zeta[i])/Cost[i])$;
18   **end**
19   **if** $X[i]$ AND $f_p \leq r$ **then**
20     $f_p \leftarrow max((\Omega[i] + \zeta[i])/Cost[i])$;
21   **end**
22 **end**

The algorithm receives $\mathcal{C}$, $\mathcal{B}$, Υ, Matrices A and $Cost$, $\mathcal{V}$, $\mathcal{W}$, $price$, and r as inputs. As the normalization factor vector depends solely on attribute values from Matrix A, it is pre-calculated (and updated when needed) before any request. Initially, the algorithm sets the X vector to TRUE, meaning that all the Fog nodes could attend the request, but this binary variable will be set to FALSE in the case of any attribute not meeting the user's request (lines 5-7) or if Fog node's cost is higher than requested (lines 9-11). The normalized attribute values P are calculated (line 4), as defined in Equation 2. After that, the values of Ω and ζ are also calculated (lines 13-14), allowing the assessment of the cost-benefit ratio. So, the next step is to determine the best *Fog nodes* for the *USR Solution* (f_u, in line 17) and for the *PRV Solution* (f_p, in line 20) within the limit of Fog nodes requested by the end-user (r).

Finally, considering that one of the challenges of Fog computing, when compared with other more consolidated computing paradigms such as Cloud computing, is the high geographic distribution that requires a resource allocation solution capable of handling many requests in a short time interval [13], our proposed system architecture was built based on a messaging system, such as Apache Kafka [14].

V. Motivating Scenario

Fog computing is employed to supply use cases in which Cloud computing is insufficient, such as healthcare, smart buildings, vehicular networks, and data stream processing [15]. Here, a scenario involving a Fog environment supporting a smart home application is presented for demonstration purposes. In this instance, the system must select a Fog node to perform a task related to processing safety circuit images. The capabilities attributes are CPU, memory, storage, and latency. Availability, scalability, reliability, and mobility are used as behavioral characteristics. They were randomly chosen to achieve performance tandem. We encourage the reader to look at other attribute references, such as those presented in [5], to further test this proposal.

Considering that pricing in Fog computing is still an issue under discussion in academia [16], two scenarios must be considered. In the first scenario, the cost of the Fog node is proportional to the amount of resources (CPU and Memory) available. That is, the greater the computational power, the higher the cost. Another possibility would be a single price for all Fog nodes, regardless of their configuration.

Table I details the needs of a potential requester. Therefore, they are considered input data in the presented proposal. In this case, the weights are distributed equally among the attributes. Table II presents hypothetical values with proportional values for all *Fog nodes* available in this smart home example. It is considered the Matrix A for analyzing the proposal.

When running the algorithm with the values presented in Table II, for *USR Solution*, we found FN12 with a calculated cost-benefit value of 0.68. When we analyze this Fog node, we see that it meets all the requirements requested and has a value well below the others, favoring decision-making based on cost-benefit. Likewise, when choosing the *PRV Solution*, FN18 is pondered, with a cost-benefit value of 3.28 and a cost of 0.43.

If all Fog nodes have the same cost, the efficiency of the proposed algorithm becomes even more evident. For this second scenario, it was considered that all Fog nodes have a single value of 1.00. All other values in Table II were maintained. After executing the algorithm, the Fog node FN08 was chosen for the USR Solution, which has the best attribute values. For PRV Solution, FN14 was selected as the Fog node

TABLE I
INPUT VALUES.

	Capabilities attributes				Behavioral characteristics			
	CPU	**Memory**	**Storage**	**Latency**	**Availability**	**Scalability**	**Reliability**	**Mobility**
Values	4	16	2000	4000	5	2	1	3
Weights	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Price	1,50							

with the slightest difference between what was requested and what is being delivered in all attributes.

Another difference in using a methodology based on multiple-choice criteria is the possibility of changing the weights between variables, favoring the choice of Fog nodes with more resources in a given attribute. Unlike the previous example, where the weights were distributed equally, if the requester needs to distribute them differently among the attributes, the proposed algorithm can handle it.

To illustrate this, the same Matrix A will be used, the same requester attribute values will be used, and the requester attribute weights will be changed to 47% for CPU and 1% for memory, storage, and latency. The behavioral characteristics were maintained at 12.5% each. The scenario of all Fog nodes having the exact price will be kept to demonstrate the algorithm's efficiency. In this case, the *Fog node* selected in *USR Solution* is FN08, the *Fog node* with the maximum value for CPU in Matrix A. In *PRV Solution*, the *Fog node* selected is FN07, with five CPUs delivered, just one CPU more than requested. FN07 is the *Fog node* in Matrix A with the lowest value for the CPU attribute. The differences between the choices with the weight variation are shown in Figure 1.

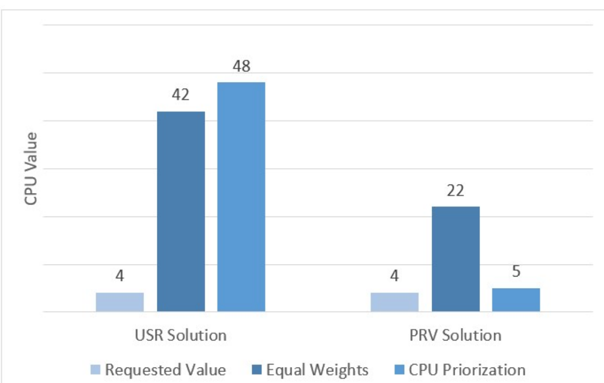

Fig. 1. CPU values with different weights.

VI. Performance Evaluation

We evaluated this proposal's performance for resource allocation in Fog computing in real and simulated environments.

A. Real Test Environment

A real test environment was built to support the evaluation of the proposal. This environment consisted of different IoT devices, such as Smart TVs, smartphones, notebooks, WiFi mesh, virtual assistants, etc. In addition, Fog nodes comprised four Raspberry Pi units and four virtual machines. The server was a Quad-core 16GB RAM. In total, the real test environment consisted of 27 devices. The IoT devices, the Fog nodes, and the server were in the same network range based on a wireless connection. Table III presents the computational capabilities and the behavioral characteristics of each Fog node in the environment. The costs of each Fog node were calculated in proportion to their computational capabilities.

To perform the tests, the nodes listed in Table III were registered in a Resource Catalog, a file external to the algorithm. To allow the allocation to be carried out with the algorithm's output, a script was created on the server that read the IPs indicated as selected by the application and made a connection via the SSH command to the indicated Fog nodes.

The values presented in Table IV were used in the tests performed. When running the tests with the input parameters, considering that practically all parameters are the same, including the cost, the node chosen for the USR solution was FN08 since it meets all requirements and has the lowest cost. As for the PRV solution, the node chosen was FN05, that delivers the requirements very close to what was requested and also has a good cost, which is more favorable for the provider.

B. Simulated Environment

We believe that testing in a real environment is essential, but limitations in the number of resources can influence the analysis of the performance of the proposed algorithm. Thus, another test scenario was designed in a simulated environment using *iFogSim* [17], which was used to generate Fog nodes. The main objective of testing in the simulated environment is to estimate the algorithm's performance as the number of Fog nodes in the resource catalog varies. That is the time required to find the one that should be allocated to the requester among all the Fog nodes in the catalog.

To allow a comparison of our proposal with others in the literature for the resource allocation problem, the same environment and variables were submitted for execution using another algorithm. TOPSIS [18] was selected since it has been used in many related works and is used mainly to solve MCDM problems. This comparison is presented in Figure 2.

When analyzing Figure 2 it is possible to observe that the proposed algorithm varies according to the number of Fog nodes available in the Resource Catalog. However, it proves to be suitable for Fog computing because it can always find a viable solution among the available options. This means that

TABLE II
MATRIX A.

Fog node (f)	Capabilities attributes ($\mathcal{C}$)				Behavioral characteristics ($\mathcal{B}$)				Cost (q)
	CPU	Memory	Storage	Latency	Availability	Scalability	Reliability	Mobility	
FN01	32	204	4438	941	4	3	2	3	0.24
FN02	32	331	1251	403	3	3	4	4	0.36
FN03	38	357	2730	114	1	5	1	3	0.29
FN04	36	454	2922	367	4	2	1	1	0.49
FN05	25	249	8270	3257	1	2	3	1	0.27
FN06	42	245	7648	930	5	1	2	4	0.28
FN07	5	233	3858	631	5	2	1	2	0.23
FN08	46	223	8184	903	4	1	1	1	0.27
FN09	10	376	4099	490	1	2	3	4	0.39
FN10	12	268	3737	624	1	3	3	1	0.28
FN11	14	434	7298	1775	4	3	1	2	0.44
FN12	33	58	8408	261	4	2	2	1	0.09
FN13	40	86	5071	3148	5	1	3	4	0.12
FN14	22	172	3406	2604	4	2	2	4	0.19
FN15	42	371	6057	1418	5	2	1	2	0.41
FN16	31	100	2567	689	4	5	2	3	0.13
FN17	48	119	7237	2619	5	2	4	1	0.16
FN18	32	403	5540	3856	4	4	3	3	0.43
FN19	35	225	7043	4725	5	3	3	4	0.25
FN20	20	447	1369	427	2	2	4	5	0.47

TABLE III
REAL TEST ENVIRONMENT FOG NODES MATRIX.

Fog node (f)	Capabilities attributes ($\mathcal{C}$)				Behavioral characteristics ($\mathcal{B}$)				Cost (q)
	CPU	Memory	Storage	Latency	Availability	Scalability	Reliability	Mobility	
FN01	4	4	2048	510	3	4	5	4	0.08
FN02	4	4	2048	563	3	4	5	4	0.08
FN03	4	4	2048	423	3	4	5	4	0.08
FN04	4	4	2048	601	3	4	5	4	0.08
FN05	2	8	1024	300	5	4	5	4	0.10
FN06	4	4	2048	230	5	4	5	4	0.08
FN07	4	8	4096	213	5	4	5	4	0.12
FN08	2	2	2048	360	5	4	5	4	0.04

TABLE IV
INPUT VALUES FOR REAL ENVIRONMENT.

	Capabilities attributes				Behavioral characteristics			
	CPU	Memory	Storage	Latency	Availability	Scalability	Reliability	Mobility
Values	2	2	1000	600	5	5	5	5
Weights	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Price	0.50							

our algorithm has guaranteed to find the solution in 100% of requests.

Although TOPSIS performance was better for scenarios with few Fog nodes (less than 3,000 Fog nodes), our algorithm proved to be better regarding time consistency concerning the growth in the number of Fog nodes in the environment. This is important because a Fog computing environment is expected to be populated with many devices, and therefore having adequate performance for a larger volume of Fog nodes is essential to ensure good execution of the resource allocation application. This shows that our algorithm is scalable, as it maintains a very low variation in the time to find the viable solution in Fog computing with up to 6,000 Fog nodes.

Another essential comparison with this other algorithm is the cost-benefit for the provider and the user. Specifically, TOPSIS does not use criteria to offer two different solutions, as the main objective of the algorithm is to find only the best solution, the one with the highest values. To illustrate this, let us consider that in this simulated environment, we will run an application for 1, 5, 10, and 60 minutes. To do so, let's assume that we have available the Fog nodes presented in Table II and the demand input are those given in Table I. In this case, the

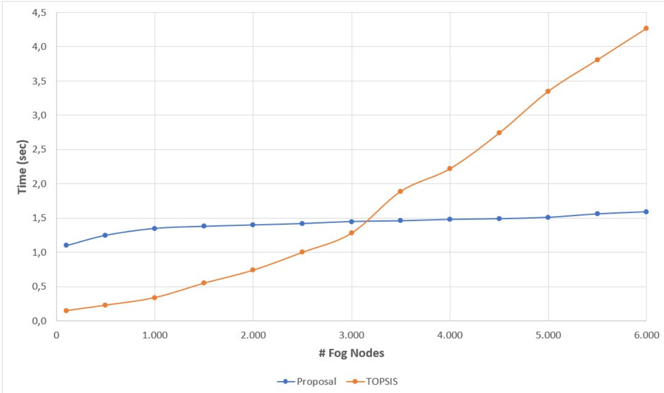

Fig. 2. Execution time with compared algorithms.

USR Solution selected the FN12, which has a cost of 0,09 per minute; to the PRV Solution, the FN18 was chosen with a cost of 0,43; otherwise, using TOPSIS, the selected fog node was FN03, which has a cost of 0,29 per minute.

The results of the algorithm executions are presented in Figure 3. Note that even though TOPSIS has a shorter execution time for scenarios with few Fog nodes, it does not select the most suitable Fog node for the user or the provider when considering the cost-benefit ratio. This becomes even more evident when the application execution time increases, showing that TOPSIS always remains in the middle line between the best cost-benefit for the user and the best cost-benefit for the provider. Thus, the efficiency of our algorithm in maximizing the results for the providers or, at the same time, delivering the best options for users, reducing resource usage costs, becomes evident.

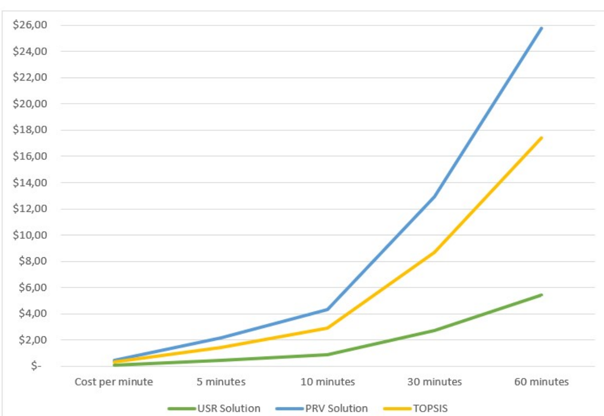

Fig. 3. Comparison of algorithm costs.

VII. Related Work

A summary of related work on resource allocation for Fog computing environments based on computational capabilities or behavioral characteristics is shown in Table V. As there is no consensus in the literature on the scope of Fog and Edge computing [19], both paradigms were considered.

From the summary in Table V, we observe that most works focus on finding the best Fog node based only on one attribute type (e.g., capabilities attributes and behavioral characteristics) and without guaranteeing that minimum computational requirements will be met. Furthermore, no paper addresses capabilities attributes and behavioral characteristics differently, i.e., the same method is used to treat both attributes. This can lead to mistakes and failure to meet users' requirements. Furthermore, no paper presents a solution that considers both user's and provider's perspectives in depth. This means that computational resources may be wasted by becoming underused or overused, which is not prudent in a Fog computing environment.

VIII. Conclusions and Future Work

We proposed an efficient algorithm for resource allocation in a Fog computing environment, considering hardware capacities and desirable features and behaviors. It covers the perspectives of both users and service providers. The user wants the best resource, and the provider seeks to find the fairest Fog node, which meets the user's needs but with minimum values. To reconcile these different goals, two solutions were presented, namely *USR Solution* and *PRV Solution*. Our experimental results show that the proposed approach is able to offer the best Fog node to the user and ensure the lowest resource consumption from provider perspective.

As part of future work, we will extend the proposed method to support the distribution of computational requirements between several Fog nodes and Cloud virtual machines or even migrate running services when current Fog nodes become unavailable. We aim to implement the resource allocation process to the Fog node selected in the scheduling step.

Acknowledgments

This study was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Brasil - Finance Code 001. Gratitude also goes to the BioCloud2 project, approved under CNPq/AWS Call No. 064/2022, CNPq process No. 421828/2022-6.

References

[1] R. Mahmud, R. Kotagiri, and R. Buyya, *Fog Computing: A Taxonomy, Survey and Future Directions*. Singapore: Springer Singapore, 2018, pp. 103–130.

[2] B. Costa, J. Bachiega Jr, L. R. Carvalho, M. Rosa, and A. Araujo, "Monitoring fog computing: A review, taxonomy and open challenges," *Computer Networks*, vol. 215, p. 109189, 2022.

[3] B. Costa, J. Bachiega Jr, L. R. de Carvalho, and A. P. Araujo, "Orchestration in fog computing: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 2, pp. 1–34, 2022.

[4] J. Bachiega, B. G. S. da Costa, and A. P. F. Araujo, "Computational perspective of the fog node," *22nd International Conference on Internet Computing and IoT*, 2021.

[5] S. Sengupta, J. Garcia, and X. Masip-Bruin, "Essentiality of resource and service-task characterization in the coordinated fog-to-cloud paradigm," in *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. IEEE, 2018, pp. 249–254.

[6] J. Bachiega Jr., B. Costa, L. Carvalho, M. Rosa, and A. Araujo, "Computational resource allocation in fog computing: A comprehensive survey," *ACM Computing Surveys*, vol. 55, no. 14, 2023.

TABLE V
RELATED WORKS SUMMARY.

Paper	Year	Method	Paradigm	Computational	Behavioral	Minimal Requirements	Avoid Waste	User´s Perspective	Provider´s Perspective	Treat attributes differently	Cost
[20]	2016	Linearized Decision Tree	Fog	✓		✓	✓		✓		
[21]	2017	Benchmarking Algorithm	Edge	✓		✓		✓			✓
[22]	2018	Regression Markov Chain	Edge	✓	✓	✓	✓	✓			
[23]	2019	FCAP	Fog	✓		✓		✓			
[24]	2019	WSGA / NSGA-II / MOEA/D	Fog	✓					✓		
[25]	2019	Genetic Algorithm	Fog	✓	✓			✓			✓
[26]	2019	PROMETHEE-II	Fog	✓				✓			
[27]	2019	ELECTRE	Fog	✓					✓		✓
[28]	2019		Edge	✓	✓	✓		✓			
[29]	2018	TOPSIS	Fog	✓	✓	✓		✓			✓
[30]	2019		Fog	✓	✓				✓		
[31]	2020		Edge	✓				✓			
[32]	2020		Fog	✓		✓	✓		✓		✓
[7]	2018	AHP	Fog	✓					✓		
[8]	2020		Fog	✓					✓		
[33]	2020		Edge	✓			✓	✓			✓
[10]	2020		Fog	✓				✓			
[9]	2021	KL	Fog		✓	✓		✓			
[34]	2022	Monarch-dragon Algorithm	Fog	✓				✓			
[35]	2024	FAHP and FTOPSIS	Fog	✓		✓	✓				
This work	2024	Adapted MCDM	Fog	✓	✓	✓	✓	✓	✓	✓	✓

[7] S. P. Eswaran, S. Sripurushottama, and M. Jain, "Multi criteria decision making (MCDM) based spectrum moderator for fog-assisted internet of things," *Procedia computer science*, vol. 134, pp. 399–406, 2018.

[8] S. Mishra, M. N. Sahoo, S. Bakshi, and J. J. Rodrigues, "Dynamic resource allocation in fog-cloud hybrid systems using multicriteria AHP techniques," *IEEE Internet of Things*, vol. 7, no. 9, pp. 8993–9000, 2020.

[9] V. L. Hallappanavar and M. N. Birje, "Prediction of quality of service of fog nodes for service recommendation in fog computing based on trustworthiness of users," *Journal of Reliable Intel. Env.*, pp. 1–18, 2021.

[10] S. Varshney, R. Sandhu, and P. Gupta, "Qoe-based multi-criteria decision making for resource provisioning in fog computing using AHP technique," *IJKSS*, vol. 11, no. 4, pp. 17–30, 2020.

[11] N. Vafaei, R. A. Ribeiro, and L. M. Camarinha-Matos, "Normalization techniques for multi-criteria decision making: analytical hierarchy process case study," in *doctoral conference on computing, electrical and industrial systems*. Springer, 2016, pp. 261–269.

[12] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

[13] A. Mukherjee, D. De, and R. Buyya, *Resource Management in Distributed Systems*. Springer, 2024.

[14] N. Garg, *Apache kafka*. Packt Publishing Birmingham, UK, 2013.

[15] R. Buyya and S. N. Srirama, *Fog and edge computing: principles and paradigms*. John Wiley & Sons, 2019.

[16] S. N. Srirama, "A decade of research in fog computing: relevance, challenges, and future directions," *Software: Practice and Experience*, vol. 54, no. 1, pp. 3–23, 2024.

[17] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[18] J. Papathanasiou, N. Ploskas, J. Papathanasiou, and N. Ploskas, *TOPSIS*. Springer, 2018.

[19] J. Bachiega Jr, B. G. Costa, L. R. de Carvalho, V. H. Oliveira, W. X. Santos, M. C. S. de Castro, and A. P. Araújo, "From the sky to the ground: Comparing fog computing with related distributed paradigms." in *CLOSER*, 2022, pp. 158–169.

[20] A. A. Alsaffar, H. P. Pham, C.-S. Hong, E.-N. Huh, and M. Aazam, "An architecture of iot service delegation and resource allocation based on collaboration between fog and cloud computing," *Mobile Information Systems*, vol. 2016, 2016.

[21] M. Noreikis, Y. Xiao, and A. Ylä-Jaäiski, "QoS-oriented capacity planning for edge computing," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[22] G. Li, J. Song, J. Wu, and J. Wang, "Method of resource estimation based on qos in edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[23] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, "Methods of resource scheduling based on optimized fuzzy clustering in fog computing," *Sensors*, vol. 19, no. 9, p. 2122, 2019.

[24] C. Guerrero, I. Lera, and C. Juiz, "Evaluation and efficiency comparison of evolutionary algorithms for service placement optimization in fog architectures," *FGCS*, vol. 97, pp. 131–144, 2019.

[25] S. B. Akintoye and A. Bagula, "Improving quality-of-service in cloud/fog computing through efficient resource allocation," *Sensors*, vol. 19, no. 6, p. 1267, 2019.

[26] M. K. Mishra, N. K. Ray, A. R. Swain, G. B. Mund, and B. S. P. Mishra, "An adaptive model for resource selection and allocation in fog computing environment," *Computers & Electrical Engineering*, vol. 77, pp. 217–229, 2019.

[27] I. Lera, C. Guerrero, and C. Juiz, "Analyzing the applicability of a multi-criteria decision method in fog computing placement problem," in *2019 Fourth FMEC*. IEEE, 2019, pp. 13–20.

[28] M. Lin, Z. Chen, H. Liao, and Z. Xu, "ELECTRE II method to deal with probabilistic linguistic term sets and its application to edge computing," *Nonlinear Dynamics*, vol. 96, no. 3, pp. 2125–2143, 2019.

[29] H. Bangui, S. Rakrak, S. Raghay, and B. Buhnova, "Moving towards smart cities: A selection of middleware for fog-to-cloud services," *Applied Sciences*, vol. 8, no. 11, p. 2220, 2018.

[30] H. Bashir, S. Lee, and K. H. Kim, "Resource allocation through logistic regression and multicriteria decision making method in iot fog computing," *Transactions on Emerging Telecommunications Technologies*, vol. 33, p. e3824, 2019.

[31] H. Huang, K. Peng, and X. Xu, "Collaborative computation offloading for smart cities in mobile edge computing," in *2020 IEEE 13th International Conference on Cloud Computing*, 2020, pp. 176–183.

[32] G. Baranwal, R. Yadav, and D. P. Vidyarthi, "QoE aware IoT application placement in fog computing using modified-TOPSIS," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1816–1832, 2020.

[33] R. Du, K. Xu, and X. Liang, "Multiattribute evaluation model based on the KSP algorithm for edge computing," *IEEE Access*, vol. 8, pp. 146 932–146 943, 2020.

[34] S. Harika and B. C. Krishna, "Multi-objective optimization-oriented resource allocation in the fog environment: A new hybrid approach," *IJITWE*, vol. 17, no. 1, pp. 1–25, 2022.

[35] A. A. Gad-Elrab, A. S. Alsharkawy, M. E. Embabi, A. Sobhi, and F. A. Emara, "Adaptive multi-criteria-based load balancing technique for resource allocation in fog-cloud environments," *IJCNC*, vol. 16, no. 01, 2024.