Analyzing the Applicability of a Multi-Criteria Decision Method in Fog Computing Placement Problem

Isaac Lera, Carlos Guerrero and Carlos Juiz
dept. of Mathematics and Computer Science
University of Balearic Islands
Palma, Spain
isaac.lera@uib.es, carlos.guerrero@uib.es, cjuiz@uib.es

***Abstract*—In Fog computing the placement selection of services in network devices with computational and storage features is an NP-hard problem. We analyse the design of Fog scenarios and the placement problem using Electre III a multi-criteria decision method for outranking alternatives. In this study, we model a Fog environment to apply a decision model for determining which alternatives are the most suitable in each application deployment. We compare the results with the weighted average to analyse the applicability of Electre III method in the Fog placement problem. We design a dynamical scenario in which new users appear along the simulation and we use the latency, hop count, cost, deployment penalty and energy consumption criteria to rank placement alternatives. This approach enables the study of how the characteristics of the resources have to be distributed, that is, how to design Fog scenarios, in order to make the allocation of applications more efficient.**

***Index Terms*—fog placement, multi-criteria decision analysis, Fog Computing**

I. Introduction

Fog computing paradigm tries to improve the latency, network bandwidth, reliability and security enabling intermediate network devices between endpoint devices and cloud infrastructure to perform computational and storage capacities [1]. In this novelty paradigm, several issues are still a challenge. Among them, the conditioned selection of intermediate devices to deploy applications or services, called fog placement problem (FPP) [2]. It is a constraint satisfaction problem where there is a finite number of possibilities and a non-linear space. For example, we assume a network formed by thousands of entities that are able to host applications. These entities are called Fog nodes or Edge nodes according to the distance from the user or gateway location. The applications can be split by independent containers or modules following the microservice architecture designs. Thus, the relationships among modules of the same application can be represented as a graph. Each module is susceptible to being deployed in a different Fog node. This deployment affects to the criteria satisfaction, represented in service level agreements, associated to each endpoint requests such as latency, availability, power consumption, resource and deployment costs, application packaging, resource utilization, network congestions, software compatibility, user preferences, cost of violation constraints, and so on [1]. In addition, FPP happens in dynamic environments since there is an arrival of new module requests and previous decisions influence in current invocations: does the previous deployment satisfy several different types of requests? In case of negative answer, how many deployments are necessary to satisfy them and where? Thus, FPP solver's objective is to decide the best alternative with non-compensatory criteria.

TIN2017-88547-P (MINECO/AEI/FEDER, UE)

Electre methods are a family of outranking multi-criteria decision aiding (MCDA) methods based on pair-wise comparison [3], [4]. These methods aid in the decision process to assess if an alternative is at least as good as an alternative b (aSb). Different versions of Electre have been developed including Electre I, II, III, IV and TRI. Electre I is used to creating a partial prioritization; Electre II, for ranking the actions and Electre III establishes an outranking degree dealing with inaccurate, imprecise, and uncertain data. In Figueira et al. [5] there are more details about the Electre family. Electre III supports decision models that include more than five criteria and at least one of the following situations must be verified: I) The evaluations of alternatives are an ordinal scale or a weakly interval scale for each criterion, II) The criteria are sufficiently heterogeneous to make aggregation difficult on a single scale, III) The loss of a criterion is non-compensatory with the gain on another one, and finally, IV) Small differences of evaluations are not significant in terms of preferences, while the accumulation of small differences may become significant in at least one criterion (Chapter 4 [5]).

Fog environments are dynamic and heterogeneous domains. In each domain, different types of applications coexist, user workloads (request and movement distributions), fog providers, and various types of contract services (service level agreements). The decision process of bringing services from cloud providers to Fog nodes becomes a repeated NP problem. Under these premises, we evaluate the efficiency of a suitable MCDA method like Electre for the selection of Fog nodes. That is, in this work, we analyse the possibilities of applying optimization algorithms used in current literature on this type of flexible domains and the feasibility of Electre, as a multi-criteria non-compensatory method, to manage the uncertainty of performance of the alternatives and not exceeding QoS limits of some criteria such as the deployment costs or latency values. The evaluation using preference and indifference thresholds of the placement locations enables the analysis of weighting between the most optimal solution and those solutions that meet the user requirements. In addition, this analysis opens future analysis about the distribution of resources in Fog scenarios.

The remaining parts of this paper are organized as follows. Section II is structured in three descriptions along with state of the art of the Electre method, formalization of the FPP, and the decision model applied to the FPP. Section III and IV presentation of the tests and their evaluations. Finally, we conclude and explore the future lines in section V.

II. Related work and background

In this section we describe briefly the theory of the Electre III method, we present the fog placement problem in terms of our decision model, and we summarize related works.

A. Electre III method

From Electre family, Electre II, III and IV are ranking methods. Version III includes the management of data uncertain with respect to version II, and version IV is addressed to problems where it is difficult to define the importance of the coefficients of criteria. We introduce Electre III method at a sufficient level to understand our decision model but there is more related information in the related bibliography references: [5], [6]

In Electre methods, the characterization of the assertion "a outranks b" (aSb) requires the definition of the evaluation functions of the performance of each alternative. Thus, n alternatives $A = a_1, a_2, ..., a_n$ are evaluated in terms of m criteria $(g_1, g_2, ..., g_m)$. The value from $g_j(a_i)$ represents the performance of the i-th alternative with respect to j-th attribute. a alternative is considered better than alternative b according to the criterion i when $g_i(a) > g_i(b)$. We can introduce a preference model (a pseudo-criterion) including two thresholds: a preference threshold $p_i(g_i(a))$ and an indifference threshold $q_i(g_i(a))$ and hence, $p_i(g_i(a)) > q_i(g_i(a)) \forall g_i(a) : a \in A$. Each criterion g_i has an importance, a weight $w_i > 0$: $\sum_i w_i = 1$. The acceptability of the assertion aSb is based in two concordance and discordance matrices. The concordance matrix $C(a,b)$ represents the level of majority among the criteria a favour of the assertion aSb: $C(a,b) = \sum_i w_i C_i(a,b)$ where:

$$C_i(a,b) = \begin{cases} 1 & \text{if } g_i(a) \geq g_i(b) - q_i(g_i(a)) \\ 0 & \text{if } g_i(a) \leq g_i(b) - p_i(g_i(a)) \\ \frac{p_i(g_i(a)) + g_i(a) - g_i(b)}{p_i(g_i(a)) - q_i(g_i(a))} & \text{otherwise} \end{cases}$$

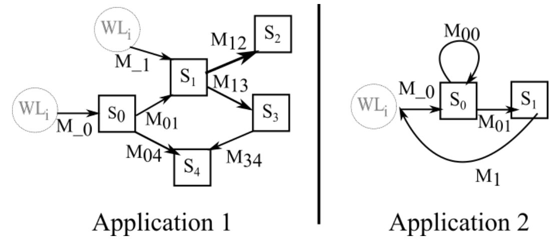


Fig. 1: Two application examples formed by a composition of services.

The discordance matrix depends on veto threshold (v_i). The veto value enables the rejection of an alternative under a criterion: $g_i(b) \geq g_i(a) - v_i(g_i(a))$. The discordance matrix $d(a,b)$ represents the weight of the number of unsatisfied criteria, where:

$$d_i(a,b) = \begin{cases} 1 & \text{if } g_i(a) \geq g_i(b) - v_i(g_i(a)) \\ 0 & \text{if } g_i(a) > g_i(b) - p_i(g_i(a)) \\ \frac{g_i(b) - g_i(a) - p_i(g_i(a)}{v_i(g_i(a)) - p_i(g_i(a))} & \text{otherwise} \end{cases}$$

Finally, aSb or S(a,b) can be defined like:

$$S(a,b) = \begin{cases} C(a,b) & \text{if } g_i(a) \geq g_i(b) - v_i(g_i(a)) \\ C(a,b) \prod_{i \in K} \frac{(1 - d_i(a,b))}{1 - C(a,b)} & \text{otherwise} \end{cases}$$

where K in the group of the criteria for which the $d_i(a_b) > C(a,b)$ is valid. In Electre III, we can define veto thresholds $v_i(g_i(a))$. These values define discordance thresholds. They can be so large that the discordance values are equal to zero, so $S(a,b) = C(a,b)$.

B. Fog placement model

We describe the fog environment modelling, the description of previous statements in the Fog placement model. Let's assume k applications $(app_a, app_b, ..., app_k)$, each application is a graph $app_k = G_k$ where vertices represents dependencies between $m_{k,i}$ modules. A module offers a containerized service of the application. The number of modules in each application can be different. In Fig. 1, we show two examples of applications with characteristics such as several workloads, a loop message and services calls from different modules. Using this representation each module could be deployed in several Fog nodes according to the strategies planed in our decision model.

The network is also defined using a graph-based approximation. In our model, we have nodes n_i that are connected through links. Nodes can represent network entities such as routers, hubs, etc. and Fog or Edge nodes. Vertices among nodes represent network connections. Workloads or users endpoints are allocated in nodes according to the scenario definition and the decision to allocate application modules is called the fog placement problem (FPP). This decision is triggered when a new module is requested from a workload endpoint or from another module $r_{n_i, m_{k,j}}$, where n_i

represents the network node where this request appears and $m_{k,j}$ the application module that is requested. In this point, it is necessary to decide the node where the service will be deployed. Multiple factors have to be taken into account, mainly compliance with service level agreements. The choice of the location of the service deployment affects in the latency, bandwidth, availability, response time, node packaging, billing, user preferences, availability, network congestion, software constraints, and other factors [1].

C. Related work

In the literature, there are diverse strategies to find the most suitable allocation. Some of them are addressed the problem in the use of optimization algorithms such as Genetic algorithms [7], [8], Swarm optimization [9], [10], Monte Carlo simulations [11]; using subregions of the topology through concepts such as fog colonies [2], [12], [13], or using linear programming [14]–[18].

Most of them work with a statical vision of the scenario. They have to decide which is the best place knowing the type of workload, the location of the initial requests and the invariable characteristics of the devices. Its application in changing environments would have a high computational cost. Electre methods are greedy proposals but they can be applied in dynamical scenarios for the following reasons: I) there is no need to find the best location of a module but a location that is within of the tolerance thresholds; II) the requirements of an application or a user can vary dynamically modifying the thresholds; III) the criteria change for each user or application, they do not have to be equal for all; IV) the vast majority of the valuations of the nodes remain unchanged in successive assessments so they are subject to be searched; V) it can be used to complement or use optimization algorithms since it only values the options.

Regarding the applicability of MCDA methods in computational problems, there is the Marzouk [19] et al. approach where they use Electre III in the Value Engineering to increase in the efficiency of the resolution process. And [20] and [21] analyse the selection of cloud providers through different MCDA methods.

III. Experiment description

In order to apply Electre III method in Fog Placement context, we have to define: which are the alternatives, how will the evaluation process be, and the preference, the indifference and the veto values for each scenario. In addition, we represent a dynamical scenario where new users arrive along the simulation execution and they request services under multi-criteria of preference.

The alternatives are the options on which we can deploy the service, that is, the Fog nodes. Thus, we can define how good a node is with respect to another in the deployment of a type of service: $n_i S n_j(app_i)$.

The evaluation of this pairwise comparison depends on the criteria that determine the goodness of a node to host a service. Note that MCDA models enable a flexible definition of placement strategies since they can be personalized for each user and application request. It means, one user may require a small latency time and another one prioritizes the cost of deployment. In the experimentation of this study, we have included the following criteria:

- g_1. Latency is a time delay between the sending and receiving of an application message located in two network nodes. The latency is computed according to the link characteristics (the bandwidth and link propagation). In each estimation, we use the theoretical value without considering the current network traffic.
- g_2. Hop Count is the number of intermediate network devices through the request pass between the source and the destination node. We use the shortest path between both nodes to compute this value. The hop count and the latency will be related criteria according to the characteristics of the links.
- g_3. Energy consumption refers to the total watts to perform a task: application deployment, service of a requests, and undeployment. We only consider the energy consumption associated to the deployment.
- g_4. Cost is the price for deploying the application in the node.
- g_5. Deployment penalty represents the overhead for deploying the application in a node. This penalization has a value directly related to the number of instructions per second that the Fog node can perform.

The objective for any user is to minimize all of the criteria, although having different preference, indifference and veto thresholds.

An effective way to define the three thresholds is through the service level agreement in real environments where the range of evaluation of these criteria is known approximately. It means, it is possible to know the distribution of the energy consumption of an application and depending on these values determine the value of preference, indifference or veto. For example, we assume that the execution model of an application follows a distribution similar to the consumption of a CPU depending on its utilization [22], we could establish which values are tolerable in each threshold. Fig. 2 shows an example of a selection of three thresholds according to the distribution. Alternatives below the preference threshold will be chosen by those that are in another threshold. Alternatives in the preference threshold are similar. Alternatives between the preference and the indifference are equals but are not preferred. Alternatives above the veto value are discarded for that criterion.

In our experiment, we use the percentile rank as a reference point for the evaluation of each criterion. Through several experiments, we select the following value of the thresholds: the preference value is the 20th percentile rank, the indifference value is one-third of the 10th percentile (indifference threshold is 20th + 10th/3), and the veto is the 40th percentile rank.

In order to rank each alternative, it is also necessary to determine that criterion is more important than the others. In Electre III method, weights are defined for each criterion. The

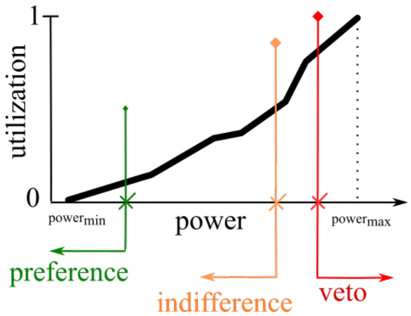


Fig. 2: Preference, indifference and veto thresholds in an energy consumption distribution.

use of the weights enables the comparison of this approach with another method that is the weighted average. The use of the weighted average to select alternatives under several criteria is widely used. Therefore, we analyse the contributions of this Electre method in our study with respect to the weighted average.

We use the Electre III implementation available in the package “outranking tools” of R[1].

Finally, the comparison between both methods and the scenario modelling is through the use of a Fog simulator. As far as we know there are different Fog simulators such as Fog-Torch [23], EmuFog [24], EdgeCloudSim [25], iFogSim [26], and YAFS [2]. We choose YAFS since it is the only one that supports dynamic changes during execution, a flexible definition of the environment using JSON files and the results are generated in a raw format for later analysis. In addition, we include the parametrization of this study (configuration files and source) in the YAFS repository[2].

A. Scenario parametrization

The scenario is defined by a topology, the parametrization of the Fog nodes and network links, the workload, and the applications. To the best of our knowledge, there is no real and open scenario with such characteristics and with the data available in the literature. Thus, we choose common distributions such as uniform and exponential that they facilitate the interpretation of the results.

The topology is created using aSHIIP tool [27]. It is an autonomous generator of random Internet-like topologies with Inter-domain Hierarchy. It includes several topology generators. In our case, we use the GLP (Generalized Linear Preference) model *inspired in the power law exponent and the clustering behaviour of the Internet*. We use the aSHIIP tool with the following parameters: GLP option, 200 nodes and these default values: $m0$ 20, m 1, p 0.45, and $beta$ 0.64. Parameter explanations are in [28]. In this type of topology, the nodes have a level according to their degree. Nodes with a very low level are nodes with the highest number of connections with other nodes. The resulting random topology has the following measures: the average degree is 3.48, the diameter is 7, and the graph density is 0.017. We place the location of the cloud in a non-centric node, therefore we choose its position randomly with those nodes with a degree lower than 2 (nodes with 5 and 6 level). The topology used is shown in Fig. 3.

[1]Electre3_SimpleThresholds function - https://cran.r-project.org/web/packages/OutrankingTools/

[2]https://github.com/acsicuib/YAFS/tree/master/src/examples/MCDA/

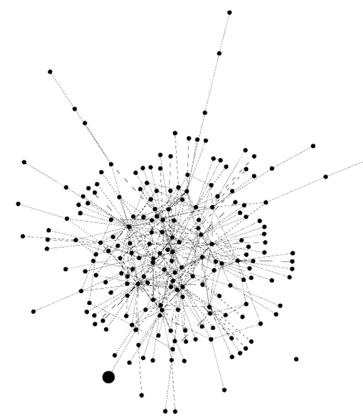


Fig. 3: GLP-based topology created with the aSHIIP tool. The bigger node represents the cloud location.

Regarding the parametrization of the Fog nodes, we simplify the topology considering all nodes like Fog nodes and we fix the parameters in various experiments. The computational speed of the nodes is fixed with a random distribution between 50 and 1000 instruction per seconds. Each Fog node only supports the deployment of a single service, except for the cloud node where all the services of all the applications are deployed. Regarding with previous criteria, the power consumption (g_3) is modelled through a random distribution, the power minimum value is fixed between a range of 50 and 300 power units and the power maximum value between 400 and 1000 power units. The cost (g_4) is modelled considering sigmoid function according to the degree value of the node (node levels between a range of 0 and 4). Thus, the first 100 nodes as more expensive than the next 100 nodes. Central nodes are more expensive than external nodes.

The propagation time of the network links is fixed with a random distribution between 10 and 90, and the bandwidth between 100 and 1000.

The number of applications is 10. The graph of each application is generated through a growing network process following a random selection of services between 2 and 10. The number of computed instructions of each service follows a random distribution between 20000 and 60000 instructions; and the message size, between 1500 KBytes and 4500 KBytes.

The workload of each application is generated following the next rule: a node can be a workload source of one application with a probability of 0.02. It means the possible number of workload sources is 40 (with 10 applications and 200 nodes). The generation ratio of requests is fixed with an exponential distribution (lambda) with random average between 100 and 1000 seconds. All workload sources are activated during the

TABLE I: Six cases with non-normalized weights used in the experimentation.

Case	Hop count	Latency	Power	Cost	D.Penalty
A	3	3	3	3	3
B	**1**	4	3	3	3
C	4	**1**	3	3	3
D	4	4	**1**	3	3
E	4	4	3	**1**	3
F	4	4	3	1	**1**

TABLE II: Cost and deployment penalty counter values.

Case	Cost		Deployment penalty	
	E.III	WA	E.III	WA
A	0	5	75	0
B	0	9	54	4
C	12	14	55	1
D	0	0	65	0
E	0	0	74	0
F	0	0	75	0

simulation following another exponential distribution (lambda) with a random average between 100 and 1000 seconds.

The simulation time is 10000 seconds and the generation of the workload is always the same in each simulation.

IV. Results

We compare Electre III method and weighted average using different cases where the weights of the criteria are varied. The first case (*Case A* of the Table I) is the reference case where all the criteria have the same weights. In the rest of cases (from B to F of the Table I), a criterion presents more importance to the rest. The non-normalized weights of Table I are in a range of 1 to 4, where the lower value is the one with greater weight. For example, the case where the hop count criterion has more importance is the *Case B*. As the latency and the hop count have an implicit relationship for the topology, we have penalized their weights more than the rest of the criteria (it is the value of 4). With all these cases, we present some results about the behaviour of the solutions in each model and its impact on the criteria.

For space reasons, we could not include all the figures along this study, but all the material is available on-line[3] including the code, the scenario generator and the analyser of the results.

The first result that we analyse is the response time of the applications. In the Fig. 4 we present the response time of each application and for each user or gateway. They represent entry points of the workloads in the scenario. In total, there are 10 applications and, for example, in application number two, there are tree users. The above image presents the *Case A*, where all the criteria have the same weight. The below image represents the *Case C* where the latency criterion has more importance. Both methods do not present a global superiority with respect to the other in terms of response time. In the *Case C*, we should expect a better response time since we try to improve the latency, but the congestion of the network influences the response time. We fixed that a fog node is able to allocate only one service, to maximize the influence in the results of different placement policies. Thus, the distribution of services increases traffic. A more efficient criterion to evaluate alternatives will be consist on to identify the current network traffic and the future one due to the possible deployment of the new service, considering in this measurement the latency.

We can observe some differences between both methods in the analysis of the distribution of placements according to the hop count criterion. In Fig. 5, we represent the hop count distribution of each case for both methods. This value cannot be zero since the services cannot be deployed in the gateway node. The Case B represents the case where the hop count criterion is the most important. In this case, the distribution tends to small values, reaching a peak in the value 2. On the other hand, Electre III presents a very similar behaviour in the rest of the cases since, as we mentioned, the method is not looking for the most optimal but the alternatives that are within the thresholds of preference. In the case of Electre III, even the cases A, E, and F have a small increase in the value 1 and 3. This is due to the fact that the topology of the network and the rest of the characteristics influence the homogeneous valuation of resources.

The analysis of the distribution of energy consumption shows really the effect of the homogenization of the infrastructure in the selection of alternatives in both methods. As we mentioned, the power is generated using a uniform distribution. Thus, the effect of this criterion with respect to the other is minimal under similar consumptions. Even, this distribution is similar in the *Case D* which represents the case where the energy consumption is more relevant.

In Table II, we present the results for the case of the cost (*Case E*) and the deployment penalty (*Case F*). Note that Electre III method practically ignores the penalty criterion due to a homogeneous distribution of values with irrelevant consequences for the percentiles thresholds.

V. Conclusion and future work

In this work, we proposed the use of Electre III, a multi-criteria decision analysis (MCDA) method, to rank Fog node alternatives in the Fog placement problem. Thus, the deployment of applications can be defined through a service level agreement in terms of preference, indifference and veto thresholds. We perform a comparative evaluation between Electre III method and the weighted average method. We analyse six cases with regard to the different weights of each criterion. The criteria are the latency, the hop count, the energy consumption, the cost and the deployment penalty. In the results, there is a slight variation and selection of alternatives when there is greater heterogeneity of Fog node characteristics. In the case of Electre III method, we observed that all the criteria are considered within the goodness of the alternatives and in the weighted average method, it influences, as expected, the greater weight in each criterion. The definition

[3]https://github.com/acsicuib/YAFS/tree/master/src/examples/MCDA/

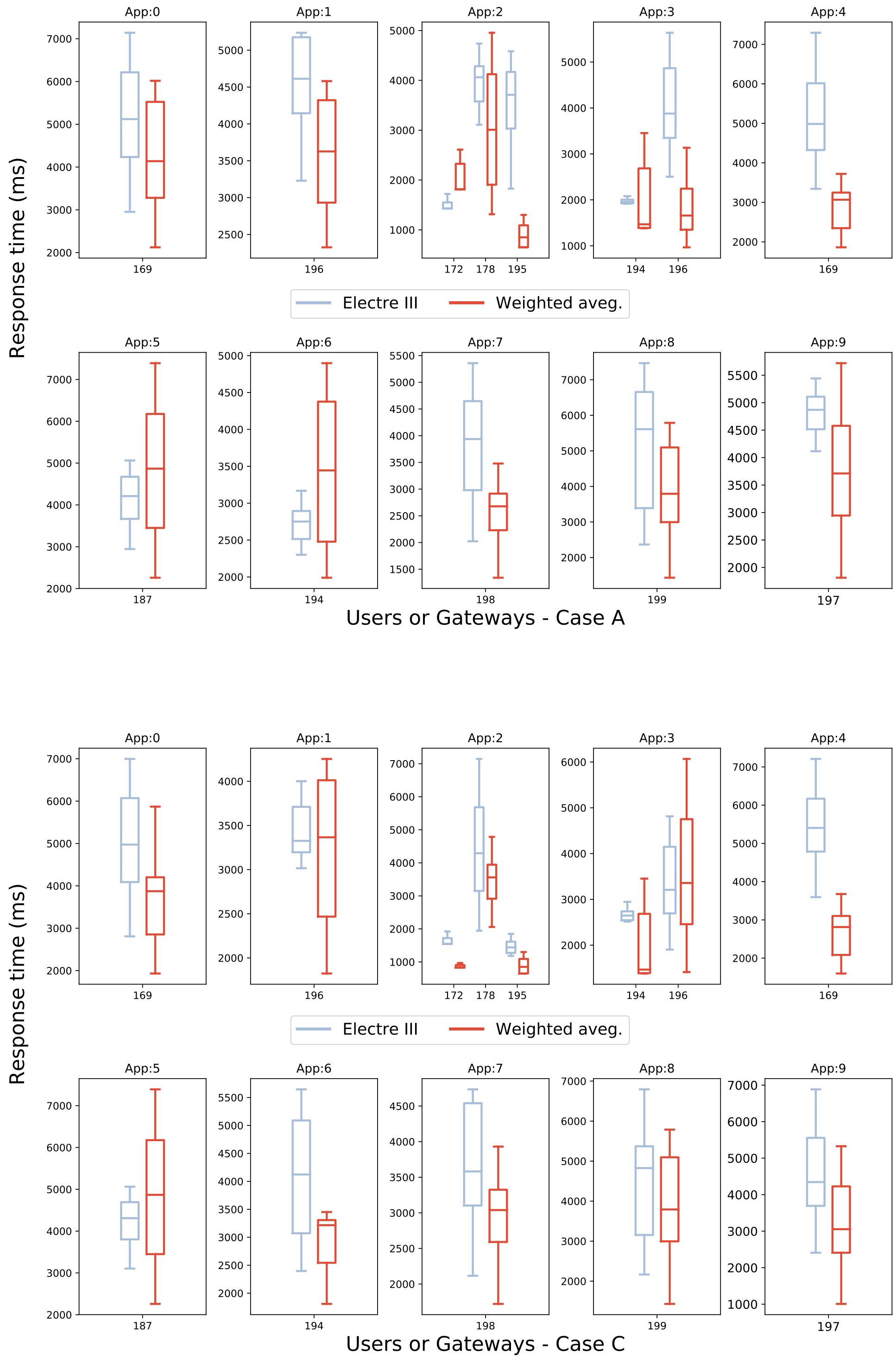
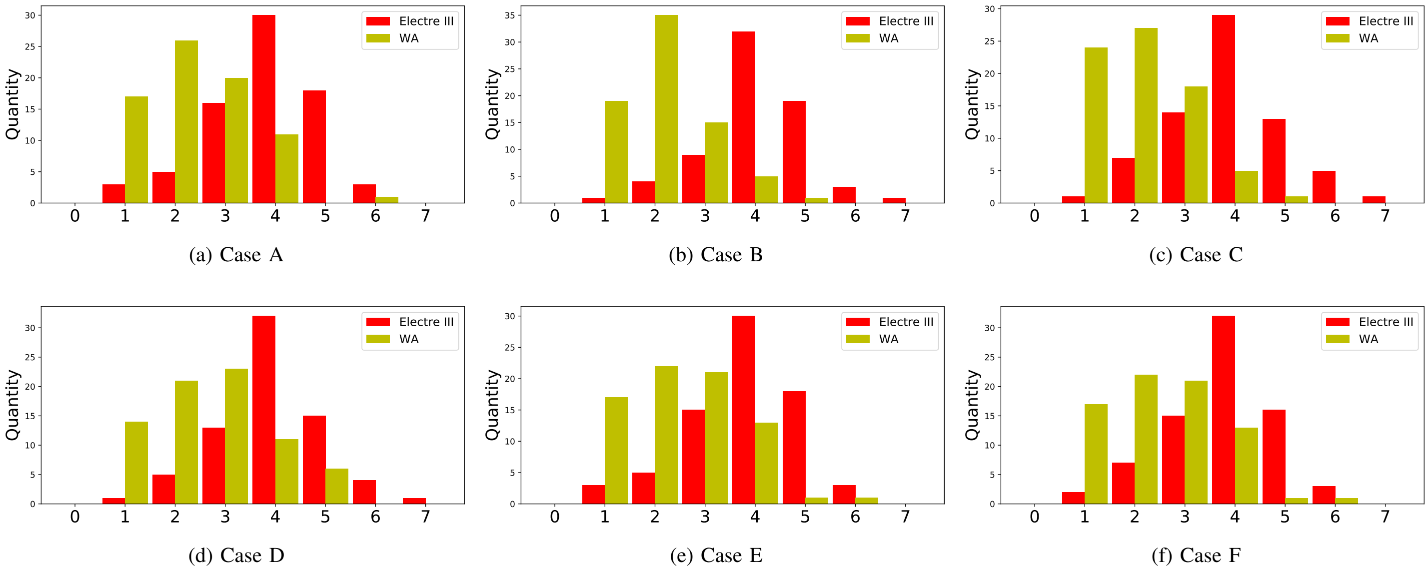


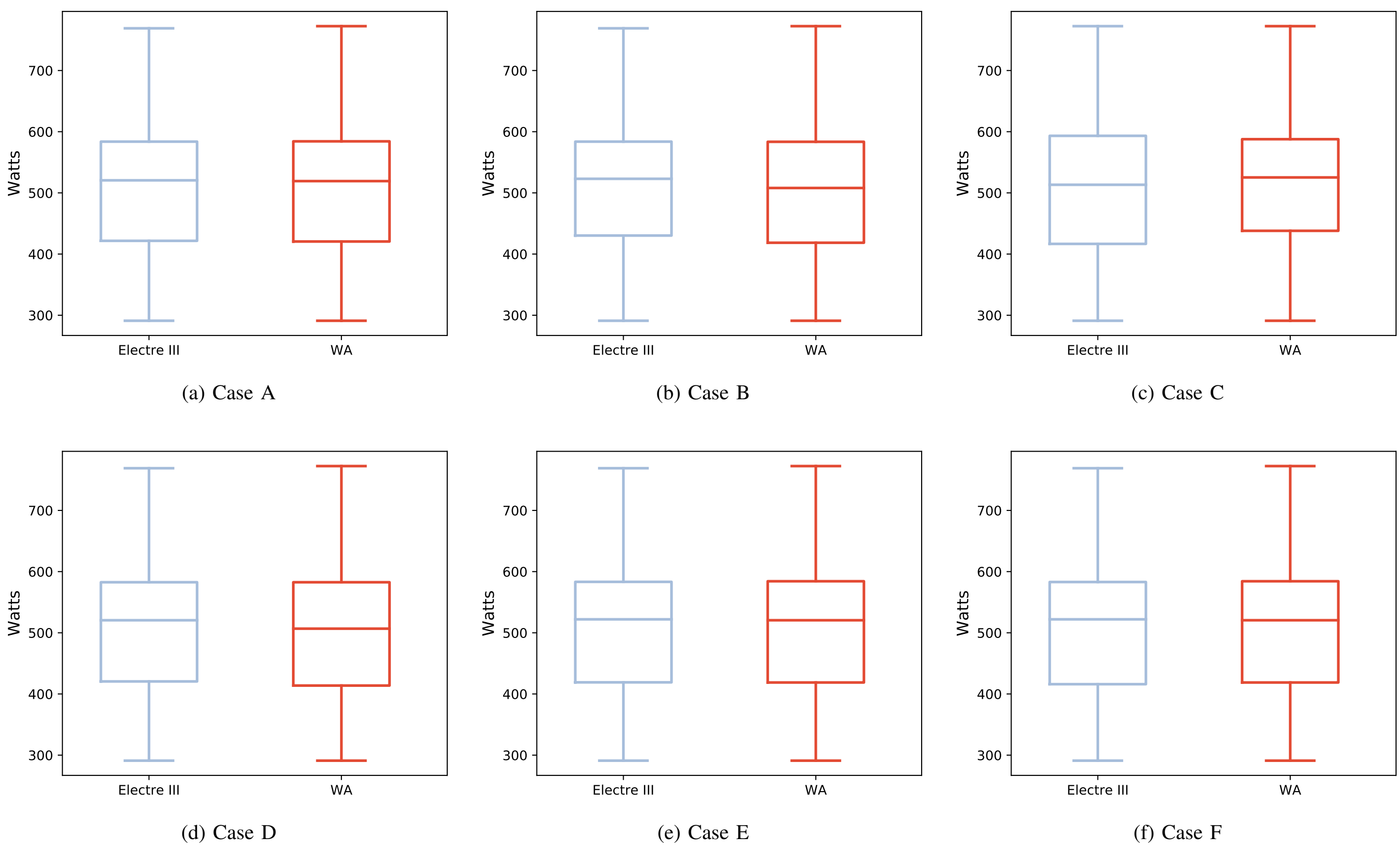
Fig. 4: The response time of each user according to its application. The above image represents the *Case A* and the below image, the *Case C*.



(a) Case A (b) Case B (c) Case C

(d) Case D (e) Case E (f) Case F

Fig. 5: Hop count distribution of both methods.



(a) Case A (b) Case B (c) Case C

(d) Case D (e) Case E (f) Case F

Fig. 6: Energy consumption distribution of both methods.

of the Fog placement problem using these methods enables the use of flexible criteria. The number of criteria can vary depending on each type of deployment. The thresholds can be adapted according to the evaluation of the indicators along the life cycle of a Fog scenario. This flexibility enables more sophisticated studies on how resources are distributed in an ecosystem and, also, the use of machine learning techniques for an adaptive approach to the placement of applications.

Acknowledgment

This research was supported by the Spanish Government (Agencia Estatal de Investigación) and the European Commision (Fondo Europeo de Desarrollo Regional) through grant number TIN2017-88547-P (MINECO/AEI/FEDER, UE)

References

[1] O. Consortium, "Openfog reference architecture for fog computing," 2017, [Online; accessed 18-December-2018]. [Online]. Available: https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL-1.pdf

[2] I. Lera, C. Guerrero, and C. Juiz, "Availability-aware service placement policy in fog computing based on graph partitions," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[3] B. Roy, "Classement et choix en présence de points de vue multiples," *RAIRO - Operations Research - Recherche Opérationnelle*, vol. 2, no. V1, pp. 57–75, 1968.

[4] K. Govindan and M. B. Jepsen, "Electre: A comprehensive literature review on methodologies and applications," *European Journal of Operational Research*, vol. 250, no. 1, pp. 1 – 29, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221715006529

[5] J. Figueira, S. Greco, and M. Ehrgott, *Multiple Criteria Decision Analysis: State of the Art Surveys*, ser. International Series in Operations Research & Management Science. Springer, 2005. [Online]. Available: https://books.google.es/books?id=YqmvlTiMNqYC

[6] A. Papadopoulos and A. Karagiannidis, "Application of the multi-criteria method electre iii for the optimisation of decentralised energy systems," *Omega*, vol. 36, pp. 766–776, 10 2008.

[7] C. Guerrero, I. Lera, and C. Juiz, "Genetic algorithm for multi-objective optimization of container allocation in cloud architecture," *Journal of Grid Computing*, vol. 16, no. 1, pp. 113–135, Mar 2018. [Online]. Available: https://doi.org/10.1007/s10723-017-9419-x

[8] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, Mar 2017.

[9] V. Moysiadis, P. Sarigiannidis, and I. Moscholios, "Towards distributed data management in fog computing," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–14, 09 2018.

[10] S. K. Mishra, D. Puthal, J. J. P. C. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4497–4506, Oct 2018.

[11] A. Brogi and S. Forti, "Qos-aware deployment of iot applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, Oct 2017.

[12] C. Guerrero, I. Lera, and C. Juiz, "A lightweight decentralized service placement policy for performance optimization in fog computing," *Journal of Ambient Intelligence and Humanized Computing*, Jun 2018. [Online]. Available: https://doi.org/10.1007/s12652-018-0914-0

[13] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized iot service placement in the fog," *Service Oriented Computing and Applications*, vol. 11, no. 4, pp. 427–443, Dec 2017. [Online]. Available: https://doi.org/10.1007/s11761-017-0219-8

[14] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, Jan 2017.

[15] Z. Huang, K.-J. Lin, S.-Y. Yu, and J. Y. jen Hsu, "Co-locating services in iot systems to minimize the communication energy cost," *Journal of Innovation in Digital Ecosystems*, vol. 1, no. 1, pp. 47 – 57, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352664515000061

[16] K. Velasquez, D. P. Abreu, M. Curado, and E. Monteiro, "Service placement for latency reduction in the internet of things," *Annals of Telecommunications*, vol. 72, no. 1, pp. 105–115, Feb 2017. [Online]. Available: https://doi.org/10.1007/s12243-016-0524-9

[17] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1440–1452, May 2016.

[18] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3702–3712, Dec 2016.

[19] M. Marzouk, "Electre iii model for value engineering applications," *Automation in Construction*, vol. 20, no. 5, pp. 596 – 600, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580510002050

[20] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, "Cloud service selection," *J. Netw. Comput. Appl.*, vol. 45, no. C, pp. 134–150, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2014.07.019

[21] M. Whaiduzzaman, A. Gani, N. B. Anuar, M. Shiraz, M. N. Haque, and I. T. Haque, "Cloud service selection using multicriteria decision analysis," *TheScientificWorldJournal*, vol. 2014, pp. 459 375; 459 375–459 375, 02 2014. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/24696645

[22] C. Guerrero, I. Lera, B. Bermejo, and C. Juiz, "Multi-objective optimization for virtual machine allocation and replica placement in virtualized hadoop," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 11, pp. 2568–2581, Nov. 2018. [Online]. Available: doi.ieeecomputersociety.org/10.1109/TPDS.2018.2837743

[23] A. Brogi and S. Forti, "Qos-aware deployment of iot applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, Oct 2017.

[24] R. Mayer, L. Graser, H. Gupta, E. Saurez, and U. Ramachandran, "Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures," in *2017 IEEE Fog World Congress (FWC)*, vol. 1, no. 1, Oct 2017, pp. 1–6.

[25] C. Sonmez, A. Ozgovde, and C. Ersoy, "EdgeCloudSim: An environment for performance evaluation of Edge Computing systems," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, vol. 1, no. 1, May 2017, pp. 39–44.

[26] H. Gupta, A. Vahida 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. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2509

[27] J. Tomasik and M. Weisser, "ashiip: Autonomous generator of random internet-like topologies with inter-domain hierarchy," in *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Aug 2010, pp. 388–390.

[28] T. Bu and D. Towsley, "On distinguishing between internet power law topology generators," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, June 2002, pp. 638–647 vol.2.