# Non-Cooperative Edge Server Selection Game for Federated Learning in IoT

Kinda Khawam[1,5], Hussein Taleb[2], Samer Lahoud[3], Hassan Fawaz[4], Dominique Quadri[5], and Steven Martin[5]
[1] Université de Versailles Saint-Quentin-en-Yvelines, 78000 Versailles, France
[2] Ecole Supérieure d'Ingénieurs de Beyrouth, Saint Joseph University of Beirut, Beirut, Lebanon#
[3] Dalhousie University, Halifax, Nova Scotia, B3H 4R2, Canada
[4] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France
[5] ROCS, LISN, Université Paris Saclay, 91190 Gif-sur-Yvette, France

***Abstract*—Computational offloading is an efficient way to help constrained IoT devices by performing heavy tasks on Edge servers, especially tasks related to Machine Learning. Moreover, due to their limited learning capacity and memory size, such devices can only store a limited amount of data as a training set for their learning. Consequently, learning prediction is bound to be smeared with relatively high error. To mend that issue, IoT devices can federate the learning process with their pairs via an Edge server. However, offloading repeatedly the learning model through a wireless access network is time consuming. Hence, although learning collectively can reduce the learned model variance, it inflicts a communication cost depending on the selected Edge server. Therefore, in this paper, we model the Edge Selection problem as a non-cooperative game where devices autonomously and efficiently select an Edge server to reduce both their learning error and their communication cost. Depending on the characteristics of the dataset, we discern two different types of games. For each game type, we implemented and compared a semi-distributed algorithm based on Best Response dynamics. We compared the obtained results with the optimal centralized approach and with a less computationally intensive meta-heuristics, to assess the price of anarchy. Our numerical analysis shows that the Best Response algorithm strikes a good balance between efficiency and swift convergence.**

***Index Terms*—Edge Computing, Non-cooperative game theory, Federated learning, Linear regression, IoT.**

## I. Introduction

Federated learning enables Machine Learning in a decentralized fashion while providing privacy and economical benefits [1]. It is particularly suitable in situations where data is naturally distributed among disparate devices and collecting that data on a Cloud server to reduce the learning error can be costly [2]. Moreover, as IoT devices are capacity and memory constrained, they cannot successfully perform learning on their own. Hence, each device keeps its own data while a shared learning model is trained on each device and aggregated centrally at an Edge server. We build on the model proposed in [3] where each IoT device seeks to obtain a linear regression learning model with minimal expected Mean Squared Error (MSE) on its data distribution. IoT devices combine their learned parameters with a group of devices that selected the same Edge server. Hence, devices need to select the most suitable Edge server in order to minimize both the learning error and communication cost incurred by federating the learning task. The Edge server selection is tackled as a non-cooperative game by autonomous IoT devices. Several games are defined depending on the data set characteristics.

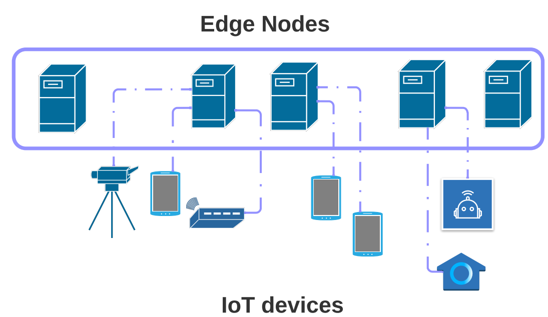



Figure 1: Edge Selection Game

The type of Federated Learning (FL) we consider in this work is coined as Edge-enabled in the thorough survey on Edge Computing found in [4]. Such a setting avoids resorting to the Cloud in order to ensure prompt computation and training of the learning model. In particular, the work in [5] explores the trade-off between communication cost and the convergence rate of the FL algorithm. We build on our previous findings in [6] where IoT devices form learning clusters via cooperative game theory, to increase both communication efficiency and learning accuracy. However, in this work, IoT devices are more constrained in term of capacity and can not rely on a simple IoT device for aggregation and signaling. Instead, we consider a more realistic assumption where FL is done through an Edge server and IoT devices need to autonomously choose the most suitable one via non-cooperative game theory. In [7], the impact of communication among federating devices was taken into account but far from game strategic considerations. Similarly, in [8] and [9], the main objective was to accelerate the convergence of federated learning in the context of an imperfect wireless channel, but not in the scope of a coalition formation game.

The main contributions of our work can be summarized as follows:

- We model the Edge Selection problem as a non-cooperative game where each IoT device strives to minimize, on its own, its cost function. The latter depends on the MSE resulting from federating the learning task and on the communication cost incurred from sending iteratively the learning model to the selected Edge server.

- We identify two distinct types of games depending on the dataset characteristics. This is important because it allows us to choose the appropriate algorithm for the specific problem we are trying to solve.
- We implement a semi-distributed Best Response dynamics for the devised games. We evaluated their performance by comparing them to the optimal solution using an exhaustive search to gauge the Price of Anarchy (PoA). As the exhaustive search is computationally intensive, we resorted for large scenarios to a meta-heuristics solution based on Simulated Annealing (SA).
- Intensive simulation results showed that the Best Response (BR) algorithm can find the best edge very promptly, without sacrificing efficiency, making it suitable for real-world scenarios.

The rest of the paper is organized as follows. We provide in section II the system model. In section III, we portray the Edge selection problem as a non-cooperative game and identify two types of games depending on the dataset characteristics. For each type of game, we characterize pure Nash Equilibriums in sections IV and V, respectively. In section VII, we give practical numerical examples for an in-depth analysis by assessing the price of anarchy given in section VI. We conclude the paper in section VIII.

## II. The System Model

We consider a set $\mathcal{L} = \{1, .., M\}$ of $M$ constrained IoT devices that aim to perform learning through linear regression on a limited data set. Devices can predict future values on the metrics they are sensing (temperature, humidity, etc.) and which constitute their dataset.

Every device $j$ has access to $n$ samples as a training dataset. The samples are denoted by $X_j$ and the predicted variable $Y_j$ depends linearly on $X_j$ such as $Y_j = \theta_j \cdot X_j + \eta_j$, where $\theta_j$ is the slope of the linear regression and $\eta_j$ is the random bias of the regression, of mean $\mu_e$. Based on the $n$ data samples collected, each device $j$ seeks to estimate the mean $\hat{\theta}_j$. However, due to the small size of the dataset and the limited learning capacity of IoT devices, this model has a relatively high error rate.

Thus, devices can federate the learning process through an Edge server $E_k$ to reduce the learning error. The training is done using an iterative process, such as stochastic gradient descent (adopted in this paper), where the model is updated in multiple rounds, based on the local data of each participating device.

We consider a set $\mathcal{K} = \{1, .., K\}$ of $K$ Edge servers. Edge servers are uniformly distributed in the network. Any IoT device $j$ that selected an Edge server $E_k$ send to the latter its estimated parameter $\hat{\theta}_j$ and Edge server $E_k$ sends back a single model to all devices that selected it as follows:

$$\hat{\theta}_{E_k} = \frac{1}{|\mathcal{E}_k|} \sum_{j \in \mathcal{E}_k} \hat{\theta}_j, \tag{1}$$

where $\mathcal{E}_k$ is the set of devices federating the learning through Edge server $E_k$ and $|\mathcal{E}_k|$ is the cardinal of $\mathcal{E}_k$ such as:

$$|\mathcal{E}_k| = \sum_{j \in \mathcal{L}} \mathbb{1}_{\{device\, j\, selected\, E_k\}}$$

$i.e.$, the number of devices that selected Edge server $E_k$.

As proven by [3], the MSE recorded for a device federating learning through linear regression with $|\mathcal{E}_k| - 1$ other devices, is given by:

$$MSE_k = \frac{\mu_e}{(n-2) \cdot |\mathcal{E}_k|} + \sigma^2 \cdot \frac{|\mathcal{E}_k| - 1}{|\mathcal{E}_k|}, \tag{2}$$

where $\sigma^2$ is the average variance in the true parameters between federating devices. Hence, we denote by $MSE_k$ the MSE rate of devices learning through the same Edge server $E_k$.

Further, when the learning model is computed through Edge server $E_k$, each device joining the latter inflicts an additional communication cost among devices in $\mathcal{E}_k$. We adopt a Time Division Multiple Access (TDMA) channel access as we consider constrained 5G cellular IoT networks (such as Narrow Band IoT (NB-IoT) [10] or 5G NR-Light (RedCap) [11]). In such a setting, devices that select Edge server $E_k$ endure a delay proportional to $|\mathcal{E}_k|$ as devices send in turn (in a given time slot) their estimated parameters to $E_k$ that will operate the aggregated estimation (roughly after $|\mathcal{E}_k|$ time slots). We consider that one packet is sufficient to send the learned parameter (one metric is learned) and is sent in one time slot. Furthermore, orthogonal channels are re-allocated among Edge servers to cancel out interference. Thus, the communication cost of device $j$ via $E_k$ is denoted by $\delta_k$ and given by:

$$\delta_k = |\mathcal{E}_k| \tag{3}$$

Finally, the global cost of device $j$ through Edge $E_k$ is:

$$C_j(a_j = E_k) = MSE_k + \alpha \delta_k, \tag{4}$$

where $\alpha$ is a normalizing factor and $a_j$ is the action of device $j$ which is selecting Edge server $E_k$.

In Table I, we summarize the adopted variables for the system model.

Table I: Notation for the system model

| Notation | Definition |
|---|---|
| $X_j$ | Samples for IoT device $j$ |
| $Y_j$ | Predicted variables for IoT device $j$ |
| $\theta_j$ | Slope of the regression model for IoT device $j$ |
| $\eta_j$ | Random bias of the regression model for IoT device $j$ |
| $\mu_e$ | Mean of the random bias of the regression models |
| $n$ | Number of data samples collected for IoT device $j$ |
| $MSE_k$ | MSE rate of devices learning through the Edge server $E_k$ |
| $\lvert E_k \rvert$ | Number of devices that selected Edge server $E_k$ |
| $\sigma^2$ | Average variance in the true model parameters between federating devices |
| $\delta_k$ | Communication cost of device $j$ via $E_k$ |
| $\alpha$ | Normalizing factor |
| $a_j$ | Action of device $j$ (which consists of selecting an Edge server) |

## III. Non-Cooperative Edge Selection Game

The formulation of the Edge Selection non-cooperative game $G = \langle \mathcal{L}, \mathcal{K}, C \rangle$ can be described as follows:

- A set of players $\mathcal{L} = (1, ..., M)$ which is the set of IoT devices.
- A set of strategies for any player $j$ is $S_j = \mathcal{K} = (1, ..., K)$ which is the set of Edge servers.
- The space of pure strategies $S$ formed by the Cartesian product of each set of pure strategies $S = S_1 \times S_2 \times ... \times S_M$.
- A set of cost functions $\{C_1, C_2, ..., C_M\}$ that quantify the players' preferences over the possible outcomes of the game. Outcomes are determined by the particular action $a_j$ chosen by device $j$ and actions chosen by all other devices $a_{-j}$. Action $a_j$ corresponds to selecting a particular Edge server. Hence, it is a vector such that $a_j = (a_j^k, k = 1, .., K)$ where $a_j^k = \{0, 1\}$ is a binary variable that equals 1 if device $j$ selects Edge server $E_k$, and 0 otherwise. Thus, the cost of device $j$ is $C_j(a_j, a_{-j})$ given by (4).

### A. Nash Equilibrium

The main objective of non-cooperative game is to find an effective solution called a Nash Equilibrium (NE). The NE is a profile of strategies that self-interested players adhere to and from which any unilateral deviation cannot lead to a profit. As such, it can be defined as a strategy profile where each player's strategy is an optimal response to the other players' strategies:

$$C_j(a_j, a_{-j}) \leq C_j(a'_j, a_{-j}), \forall j \in \mathcal{L}, \forall a'_j \in S_j \tag{5}$$

### B. Analyzing the Cost Function

We denote by $L = |\mathcal{E}_k|$ the number of NB-IoT devices that selected the same Edge device $E_k$. We take the first and second derivative of the cost function in (4) with respect to $L$:

$$\frac{dC_j(L)}{dL} = -\frac{\beta}{L^2} + \alpha \tag{6}$$

$$\frac{d^2C_j(L)}{dL^2} = \frac{2\beta}{L^3} \tag{7}$$

where $\beta = \left(\frac{\mu_e}{n-2} - \sigma^2\right)$.

We distinguish two cases: Case I where $\beta \leq 0$ and Case II where $\beta > 0$.

## IV. Devised Game for Case I

It is straightforward to verify that the cost function is increasing in the number of devices that chose the same strategy $E_k$ as $\frac{dC_j(L)}{dL} > 0$ when $\beta \leq 0$. Hence, we are in presence of the so-called *congestion games* [12]. In our game, we only want to allow deterministically chosen actions, called pure strategies. Finite games possess at least one NE but are not guaranteed to have pure NEs. In our setting, mixed NE are precluded as it is not feasible to federate the learning through various Edge servers. Hence, we are only interested in a Pure NE (PNE).

### A. Best Response dynamics to reach PNE

Congestion games are particularly interesting as they have Pure Nash Equilibrium (PNE) and Best response dynamics permit attaining those PNE as proven in [13]. A semi-distributed Best Response dynamics where devices play in turn, and at each iteration, a centralized unit communicates the cost of each strategy (the cost realized in every Edge server), the playing device selects the strategy with the lowest cost. The algorithm terminates when no device changes its strategy from the previous round as sketched in Algorithm 1.

**Algorithm 1** Best Response Algorithm

**Initialize** The $M$ IoT devices are assigned randomly to the $K$ Edge servers.

**repeat**

  For $j = 1, ..., M$

    Device $j$ selects Edge server $k = arg\,min\, C_j(a_j, a_{-j})$

**until** *All devices have the same strategy as in the previous round*;

## V. Devised Games for Case II

In case II, we have that $\frac{d^2C_j(L)}{dL^2} > 0$ as $\beta > 0$. Thus, the cost function is convex in $L$. We denote by $L^* = argmin\, C_j(L)$. Equating $\frac{dC_j(L)}{dL}$ to zero gives $L^* = \sqrt{\frac{\frac{\mu_e}{n-2} - \sigma^2}{\alpha}}$.

It's much more tedious to find PNE for Case II. Therefore, we identify various instances of the game depending on the number of IoT devices as follows.

### A. Game II.1: $M \leq L^*$

In this game, the total number of players $M$ is lower than the argument of the minimum $L^*$. This particular scenario corresponds to a sparse IoT deployment. Given that the cost function is decreasing before the minimum, The NE consists in grouping all devices together at any Edge device. In fact, any unilateral movement will strictly increase any device cost. We conclude that in this case, only one Edge server is needed except if IoT devices are geographically very wide apart.

### B. Game II.2: $M > L^*$

In this scenario, the number of devices is greater than $L^*$. Thus, if $L^*$ devices choose the same Edge, they can realize minimal cost. This scenario can be again divided into 2 sub-scenarios: one where the total number of devices $M$ is a multiple of $L^*$ and one where $M$ is a not a multiple of $L^*$.

*1) $M = e \cdot L^*$:* The total number of devices can be written as $M = e \cdot L^*$ where $e \in \mathbb{N}$. We argue that for this scenario, a NE is where $e$ Edges are chosen by exactly $L^*$ IoT devices each. In fact, if any device decides to selfishly choose another Edge with already $L^*$ devices or an un-selected Edge where it is on its own, it would face a strictly higher cost as it is no longer at the optimal ($L \neq L^*$). Hence, no device will change its action unilaterally.

*2) $M = e \cdot L^* + e'$:* The total number of devices can be written as $M = e \cdot L^* + e'$ with $e, e' \in \mathbb{N}$ with $e' < L^*$. In this case, we have two possible scenarios and we will identify for each a PNE:

*a) If $e' \leq e$:* We argue that a NE is where $e - e'$ Edges are chosen by exactly $L^*$ devices each and $e'$ Edges are chosen $L^*+1$ devices each. Any movement among the Edges will either deteriorate the cost (movement from an Edge selected by $L^*$ devices to an Edge selected by $L^* + 1$ devices, as the cost is increasing for $L > L^*$) or leaves it unchanged (movement from size $L^* + 1$ Edge to size $L^*$ Edge) which proves the Nash stability. Moving to an un-selected Edge server is precluded as the cost is prohibitive and Federated learning looses its purpose (the device endures the communication cost without harvesting the gains from FL).

*b) If $e < e' < L^*$:* In that case, $e'$ can be re-written as $e' = k \cdot e + f$, where $f < e$ (corresponding to the Euclidean division of $e'$ by $e$). We argue that a Nash Equilibrium is attained when the $e'$ remaining devices spread evenly among the $e$ Edges already selected by $L^*$ devices. In fact, a NE is attained when $e - f$ Edges are each chosen by exactly $L^* + k$ devices and $f$ Edges are each chosen $L^* + k + 1$ devices. Any movement among the Edges will either deteriorate the cost (movement from an Edge selected by $L^* + k$ devices to an Edge selected by $L^* + k + 1$ devices, as the cost is increasing for $L > L^*$) or leave it unchanged (movement from size $L^* + k + 1$ Edge to size $L^* + k$ Edge) which proves the Nash stability. Other PNE exist, but this particular one is a good compromise between the devices interest (realizing an acceptable cost, as close as possible from the optimal cost at $L^*$) and the system interest as we limit the number of operating Edge servers.

Extensive numerical simulations carried out show the validity of the proposed pure Nash Equilibriums. Any unilateral deviation from the proposed PNE deteriorates the cost or leave it unchanged for the deviating device.

## VI. Price of anarchy PoA

To evaluate the efficiency of the non-cooperative game approach, we compare it against the optimal solution where we seek to minimize the total cost of devices. The corresponding optimizing problem $(\mathcal{P})$ can be written as follows:

$$\underset{\boldsymbol{a}}{\text{minimize}} \qquad C_{tot}(\boldsymbol{a}) = \sum_{j=1}^{M} C_j(a_j, a_{-j}) \tag{8}$$

$$\text{subject to} \qquad a_j^k = \{0, 1\},\ \forall j \in \mathcal{L},\ \forall k \in \mathcal{K} \tag{9}$$

$$\sum_{k=1}^{K} a_j^k = 1,\ \forall j \in \mathcal{L} \tag{10}$$

$$\sum_{j=1}^{M} a_j^k = 0 \vee \sum_{j=1}^{M} a_j^k > 1,\ \forall k \in \mathcal{K} \tag{11}$$

where $\boldsymbol{a} = \{a_j, j = 1, .., M\}$.

Constraints (10) impose that each device is attached to only one Edge server. Constraints (11) impose that either an Edge server is unselected or selected by at least one device.

Problem $(\mathcal{P})$ is a binary non-linear optimization problem. Such a problem can be solved using an exhaustive search algorithm. However, the complexity is in $O(M^K)$. This makes the exhaustive search computationally intensive, and rapidly becomes intractable for modest sized networks. Therefore, we resort to the well-known Simulated Annealing heuristic (SA).

The SA heuristic includes an acceptance probability, which can prevent the algorithm from terminating at local minima [14]. Moreover, the SA algorithm is quite effective in comparison with the exhaustive search.

**Algorithm 2** SA heuristic

**Initialize** Let $\boldsymbol{a}(t = 0)$ such as devices are spread on Edge servers.
**repeat**
  For $j = 1, ..., M$
  Select randomly an Edge server and compute $C_{tot}(t)$
    If $C_{tot}(t) < C_{tot}(t-1)$, update $\boldsymbol{a}(t)$
    Otherwise, update $\boldsymbol{a}(t)$ with probability $e^{\frac{C_{tot}(t) - C_{tot}(t-1)}{T}}$.
  ++t;
**until** $t < N$;

Our heuristic starts with an initial feasible solution where all devices are evenly spread across Edge servers, while making sure an Edge server serves more than one device. Then, at each iteration, a device is randomly chosen to change its Edge server which is selected uniformly from the available Edge servers. This is a candidate solution for which the total network cost is computed. The candidate solution is accepted if it decreases the total cost in comparison with the previous iteration; otherwise, it is accepted with a given probability. The algorithm iterates until a given stop criterion is satisfied. In our case, until a number $N$ of iterations is reached. We denote by $T$ the SA temperature that is kept constant.

## VII. Numerical Simulations

We evaluate in this section the performances of the devised games portraying Edge server selection by IoT devices. Numerical simulations were run in Matlab. Recall that $\sigma^2$ is the variance in the true model parameters between federating devices, $n$ is the number of samples in the training dataset for the devices and $\mu_e$ is the mean of the random bias of the linear regression for the dataset considered. Finally, $\alpha$ is the normalizing factor introduced in the definition of the device cost function. We considered two scenarios with 3 and 5 Edge servers respectively, as, in a practical setting, Edge servers are co-localized with Base Stations, and up to 5 Base Stations spans a reasonable geographically network area. Moreover, devices should limit their choice to the geographically closest Edge servers to curb the suffered communication latency.

In subsection VII-A, we mainly assess the various games performances through the Price of Anarchy by confronting the performances at NE against the Optimal scheme. We adopt the following numerical values: $\sigma = 2$, $n = 12$, $\mu_e = 10$ (for Case I) and $\mu_e = 50$ (for Case II). Finally, we consider $\alpha = 0.5$.

### A. Assessing the Price of Anarchy

We assess the Price of Anarchy for the various devised games in what follows.

*1) PoA for Case I:* First, we evaluate the PoA with 3 Edge servers by comparing the total network cost obtained by the optimal solution solved via exhaustive search, the SA heuristic, and the cost at NE via Best Response (BR) dynamics. The results are reported in Fig. 2. With only five devices, the three approaches perform identically with a total cost of 20. As the number of devices increases, the performances start to slightly vary with that of the SA remaining very close to the optimal solution. The cost is slightly higher at PNE.

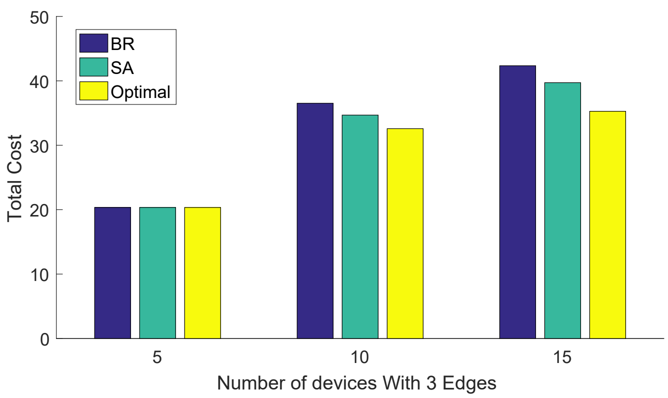



Figure 2: Total network cost vs the number of devices

With 5 Edge servers, we resort only to SA because the exhaustive search is computationally hard. In Fig. 3, we compared the cost resulting from using the Best Response dynamics to that of the SA heuristic. With 10 devices present, the performance of the two is near identical with a cost close to 40. Nonetheless, as the number of devices increases the gap between the performances of the SA heuristic and the Best Response dynamics further narrows. This gives precedence to the BR algorithm as it takes no more than two iterations to converge, contrarily to the SA as shown is Fig. 4. The latter takes around 5 iterations to converge with 10 devices present and close to 30 when there are 120 devices.

*2) PoA for Case II:* Here again, the same conclusion can be drawn as illustrated in Fig. 5. The discrepancies between the optimal, the SA heuristic and the BR algorithm are quite small. With three or five Edge servers and with up to 120 devices present, the SA approach exhibits almost the same cost with the optimal solution, whereas the BR algorithm renders a slightly higher cost. Nonetheless, the optimal solution is computationally intensive, and the SA heuristic needs a few hundred of iterations for convergence. As a result, the BR algorithm, which in this case converges in no more than three iterations, remains the most practical and efficient solution.

## VIII. Conclusion

Federated learning enables devices to learn collaboratively from information collected from numerous devices without sharing the original data, addressing privacy concerns. Furthermore, Federated learning overrides the need for costly data transfer to an Edge server, as only lightweight learning models are sent to the latter. Learning on a larger dataset reduces the variance in the learned model, and in turn its error. However, federating the learning process inflicts a communication cost among learning devices that must be taken into account. In this paper, devices engage in a non-cooperative game by identifying through which Edge server to best federate the learning process. Autonomous devices make their selection in order to reduce both their learning error and communication costs. Conclusions are consolidated through intensive numerical simulations showing that Edge selection depends mainly on the dataset characteristics and that semi-distributed Best Response dynamics ensure prompt convergence to highly efficient results. In future work, we intend to apply the proposed framework to a heterogeneous setting where we factor in the distance between the IoT devices and the Edge servers, as it directly impacts the communication cost.

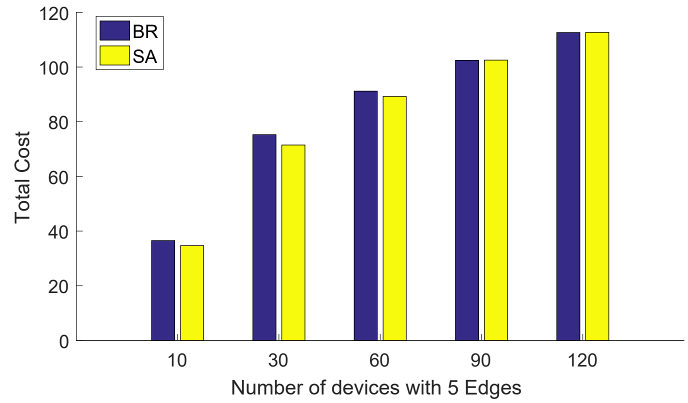



Figure 3: Total network cost vs the number of devices for BR and SA

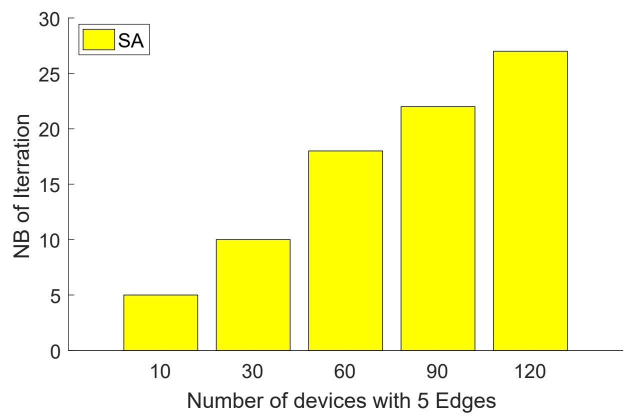
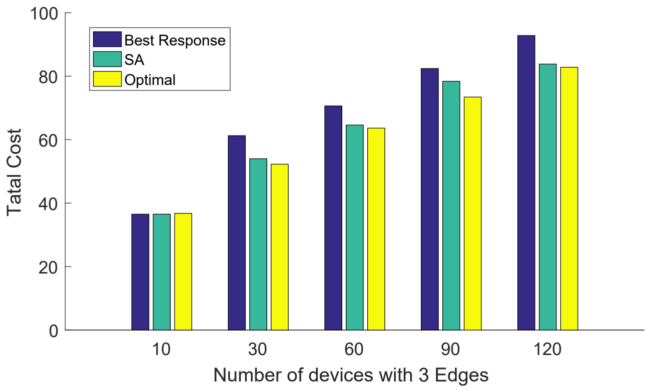
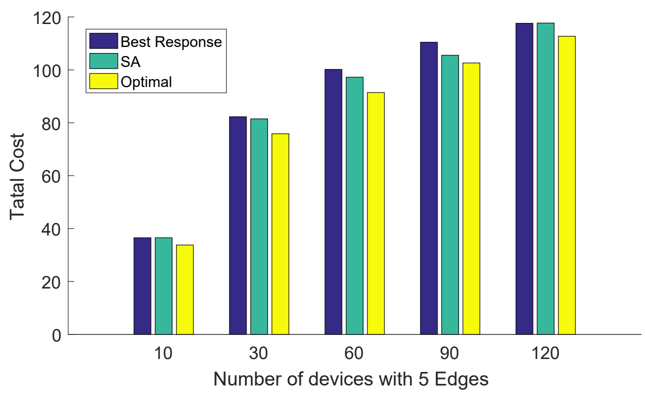



Figure 4: Mean number of iteration until convergence for SA




(a) 3 Edge servers




(b) 5 Edge servers

Figure 5: Performance as a function of the number of IoT devices

## References

[1] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Transactions on Eeing Topics in Computing*, 2021.

[2] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, "From federated learning to fog learning: Towards large-scale distributed machine learning in heterogeneous wireless networks," *CoRR*, vol. abs/2006.03594, 2020.

[3] K. Donahue and J. M. Kleinberg, "Model-sharing games: Analyzing federated learning under voluntary participation," *CoRR*, vol. abs/2010.00753, 2020.

[4] H. G. Abreha, M. Hayajneh, and M. A. Serhani, "Federated learning in edge computing: A systematic survey," *Sensors*, vol. 22, 2022.

[5] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[6] K. Khawam, S. Lahoud, C. Adjih, S. Makhoul, R. E. Tawil, and S. Martin, "Edge learning as a hedonic game in lorawan," *In Proc. of the 2023 IEEE International Conference on Communications (ICC)*, 2023.

[7] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," 2021.

[8] P. Liu, J. Jiang, G. Zhu, L. Cheng, W. Jiang, W. Luo, Y. Du, and Z. Wang, "Training time minimization for federated edge learning with optimized gradient quantization and bandwidth allocation," 2021.

[9] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE Journal on Selected Areas in Communications*, vol. PP, 2021.

[10] E. M. Migabo, K. D. Djouani, and A. M. Kurien, "The narrowband internet of things (nb-iot) resources management performance state of art, challenges, and opportunities," *IEEE Access*, vol. 8, 2020.

[11] S. Moloudi, M. Mozaffari, S. N. K. Veedu, K. Kittichokechai, Y.-P. E. Wang, J. Bergman, and A. Höglund, "Coverage evaluation for 5g reduced capability new radio (nr-redcap)," *IEEE Access*, vol. 9, pp. 45 055–45 067, 2021.

[12] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave n-person games," *Econometrica*, vol. 33, 1965.

[13] R. R. W., "A class of games possessing pure-strategy nash equilibria," *In International Journal of Game Theory*, 1973.

[14] D. T. Pham and D. Karaboga, "Intelligent optimisation techniques: Genetic algorithms, tabu search, simulated annealing and neural networks," *Springer-Verlag New York, Inc., USA, 1st edition*, 1998.