A Safe Genetic Algorithm Approach for Energy Efficient Federated Learning in Wireless Communication Networks

Lina Magoula*, Nikolaos Koursioumpas*, Alexandros-Ioannis Thanopoulos*,
Theodora Panagea*, Nikolaos Petropouleas*,
M. A. Gutierrez-Estevez**, Ramin Khalili**

Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece
*** Munich Research Center, Huawei Technologies Duesseldorf GmbH, Munich, Germany*
{lina-magoula*, nkoursioubas*, gthanopoulos*, dpanagea*, nipet*}@di.uoa.gr
{m.gutierrez.estevez**, ramin.khalili**}@huawei.com

Abstract—Federated Learning (FL) has emerged as a decentralized technique, where contrary to traditional centralized approaches, devices perform a model training in a collaborative manner, while preserving data privacy. Despite the existing efforts made in FL, its environmental impact is still under investigation, since several critical challenges regarding its applicability to wireless networks have been identified. Towards mitigating the carbon footprint of FL, the current work proposes a Genetic Algorithm (GA) approach, targeting the minimization of both the overall energy consumption of an FL process and any unnecessary resource utilization, by orchestrating the computational and communication resources of the involved devices, while guaranteeing a certain FL model performance target. A penalty function is introduced in the offline phase of the GA that penalizes the strategies that violate the constraints of the environment, ensuring a safe GA process. Evaluation results show the effectiveness of the proposed scheme compared to two state-of-the-art baseline solutions, achieving a decrease of up to 83% in the total energy consumption.

Index Terms—Wireless Networks, Beyond 5G, 6G, Energy Efficiency, Genetic Algorithm, Federated Learning

I. INTRODUCTION

The Information and Communication Technology (ICT) industry represents an important energy consumer, utilizing 4% of the world's electricity [1]. The ICT industry is expected to reach 10-20% of the world's electricity by 2030 [2], [3]. Many ambitious climate activities are already taking place by European Commission across the ICT industry, in line with the GSMA's commitment towards net-zero carbon emissions by 2050 [4], [5].

In future wireless networks, the use of Artificial Intelligence (AI) as a key enabler has been recognized at European and global level [6]. Centralized AI approaches require a vast amount of data transfer to centrally located and energy hungry data centers, raising concerns with regard to both data privacy and energy consumption [7]. Alternative solutions such as Federated Learning (FL) have surfaced, where devices with typically low power profiles perform a model training in a collaborative manner, exploiting locally stored datasets and avoiding any raw data transmissions. The 3rd Generation Partnership Project's (3GPP) in Rel. 18, tries to adopt FL in Network Data Analytics Function (NWDAF) [8], with the introduction of the Model Training Logical Function (MTLF) in Rel. 17 that is responsible of training Machine Learning (ML) models and exposing new training services [9].

Several critical energy related challenges have been raised related to the application of FL to wireless networks [10]. Firstly, the model training process requires from the computing devices constant model update transmissions. Such model updates could consist of thousands or even millions of parameters comprising complex neural networks. This implies a significant communication overhead, making the uplink transmissions particularly challenging. Secondly, ML model architectures require complex calculations, tightly related to the size of the ML model, resulting in even more challenges in terms of the energy aspect, since significant computational work will take place on device level during training. Another critical challenge of major importance includes computation and data heterogeneity across devices, in conjunction with strict ML model performance requirements. More specifically, critical services and applications often require stringent ML model performance [11], which is directly translated to higher training times and as a result to an increase in the overall energy consumption. Such deterioration could be accelerated in case of devices differing in both available resources and statistical data distribution.

Taking into consideration the aforementioned challenges, it becomes apparent that new solutions are required in order to mitigate the environmental impact of FL in wireless communication networks, while guarantying a certain ML model performance. A number of state-of-the-art works focus on energy efficient FL, tackling the problem in terms of resource orchestration, computation offloading, as well as load balancing strategies (e.g. [12]–[18]).

Contrary to the existing efforts, the current work proposes a Genetic Algorithm (GA) approach, targeting the minimization of both the overall energy consumption of an FL process and any unnecessary resource utilization, by orchestrating the

computational and communication resources of the involved devices, while guaranteeing a certain FL model performance target. The proposed solution considers the FL model's complexity (proposed only in [17], [18]), as well as device and data heterogeneity, as part of the overall energy consumption. Towards a safe decision making, a penalty function is introduced in the offline phase of the GA that penalizes the strategies that violate the constraints of the environment, ensuring a safe GA process. Finally, considering the potentially prohibited energy cost that is required for repetitive FL executions, while exploring for optimal solutions, this is the only work that proposes a simulated and computationally cost effective FL environment that emulates a real FL process, solely in the offline phase.

The rest of the paper is organized as follows. Section II provides the system model. Section III provides the problem formulation. Section IV describes the proposed solution that is evaluated in Section VI using the simulation setup of Section V. Finally, section VII concludes the paper.

II. SYSTEM MODEL

This section provides the system model of an AI-enabled wireless communication network using FL. FL enables collaborative decentralized training of a Deep Neural Network (DNN), across multiple and heterogeneous network devices - acting as workers-, without exchanging local data samples. The set of workers, denoted by $\mathcal{K}$, is orchestrated by a coordinator node.

An FL process is comprised of a number of rounds. Each round is named as *global* iteration and is denoted by $n \in \mathbb{N}$. Let $\mathbf{w}_n$ be a vector containing the model parameters at *global* iteration n. In each iteration n, the coordinator node distributes the $\mathbf{w}_n$ of a global DNN model of size m, in bits, to the worker nodes that are involved in the FL process. Let α denote the complexity associated with the global model and measured in Floating Point Operations (FLOPs). Each worker $k \in \mathcal{K}$, after successfully receiving the vector with the global parameters $\mathbf{w}_n$, performs a local training using its locally stored data samples $\mathcal{D}_k$.

Device Computational Capability: Let $f_{k,n}$ denote the available computational capacity (i.e. Central Processing Unit (CPU) speed) of worker k to execute a local training at the nth *global* iteration (it is assumed that the coordinator node has enough computational capacity to execute any computational task). Based on $f_{k,n}$, the worker k can complete a certain number of FLOPs per cycle, denoted by $c_{k,n}$. The CPU of each worker k has also an effective switched capacitance, denoted by ς_k, which depends on the hardware architecture. The local training of a worker is considered complete when a pre-selected performance target η is reached at each worker. The performance target η could refer to error metrics (e.g. mean squared error), accuracy, etc. Let $I_{k,n}$ be the total number of *local* iterations required in order for the locally trained model of worker k to reach the pre-selected performance target η at *global* iteration n. By $\tau_{k,n}$ is denoted the time required by k to complete a local training process.

Device Communication Capability: After $I_{k,n}$ *local* iterations, each k produces a locally trained model with parameters $\mathbf{w}_{k,n}$ that needs to be transmitted to the coordinator. The communication channel between worker k and the coordinator at *global* iteration n is modeled as a flat-fading channel with Gaussian noise power density N_0 and channel gain g_k, where the fading is assumed constant during the transmission of the model. Also, let $b_{k,n}$ and $p_{k,n}$ be the assigned bandwidth and transmission power to k (it is assumed that enough bandwidth has been assigned to k to transmit its model updates). $r_{k,n}$ and $tr_{k,n}$ are the achievable data rate and the required time to upload $\mathbf{w}_{k,n}$ to the coordinator, respectively (see Section III for more details).

FL Rounds: The FL process is realized in a synchronized manner. A *global* iteration is finished when the coordinator receives updates from all workers or when a preselected time threshold H is reached. Workers therefore should complete their computational and communication tasks within this time threshold (Eq. (4)): Updates from those who have not met the time threshold are considered invalid and are not used. The coordinator node produces an updated global model $\mathbf{w}_{n+1}$, using all the received model updates, and transmits it back to the workers to start the next *global* iteration $(n + 1)$. This procedure is repeated until the global model reaches a pre-selected performance target ϵ_0. Table I summarizes the notations.

$\mathcal{K}$	Set of worker nodes
n	Index of the *global* iteration (FL round)
N_0	White Gaussian noise power spectral density
g_k	Gain of the wireless channel the worker k has access to
$f_{k,n}$	Available computational capacity of worker k at the nth *global* iteration
$b_{k,n}$	Bandwidth assigned to worker k at the nth *global* iteration
$p_{k,n}$	Transmission power of worker k at the nth *global* iteration
$r_{k,n}$	Achievable transmission data rate of worker k at the nth *global* iteration
$\mathcal{D}_k$	Local data samples of worker k
$c_{k,n}$	Total number of FLOPs per cycle that the worker k can complete at the nth *global* iteration
ς_k	Effective switched capacitance of worker k
$\mathbf{w}_n$	Global FL model produced at the nth *global* iteration
α	Complexity of the global FL model in terms of total number of FLOPs
m	Size of global FL model in bits
$I_{k,n}$	Number of *local* iterations required to reach η at the worker k at the nth *global* iteration
$\tau_{k,n}$	The time required by a worker k to complete a local training process at the nth *global* iteration
$tr_{k,n}$	The time required by worker k to transmit its model updates at the nth *global* iteration
$\mathbf{w}_{k,n}$	Model parameters produced by worker k at the nth *global* iteration
η	Pre-selected performance target imposed to the training process of all workers
ϵ_0	Pre-selected performance target imposed to the global FL model

TABLE I: Notation Table

III. PROBLEM FORMULATION

The objective is to achieve energy efficiency in the system, i.e minimizing both the overall energy consumption of the workers and any unnecessary resource utilization, while guarantying a certain global model performance ϵ_0. In each *global* iteration n, all workers consume a specific amount of energy from the power grid, denoted by E_n^{gr}, in order to complete their tasks. Two types of tasks are considered in an FL process, the computation tasks, referring to the local training processes, and the communication ones, related to the transmission of the model parameters. In case a worker does not complete its tasks in the given time threshold H, its allocated resources are considered wasted (unnecessary resource utilization), since there is no contribution to the *global* model update by this worker.

The amount of computation and transmission energy required by a worker k during *global* iteration n are denoted by $E_{k,n}^C$ and $E_{k,n}^T$, respectively, given by:

$$E_{k,n}^C = \frac{\varsigma_k \cdot I_{k,n} \cdot \alpha \cdot \mathcal{D}_k \cdot f_{k,n}^2}{c_{k,n}}, \tag{1}$$

and

$$E_{k,n}^T = \frac{m \cdot p_{k,n}}{b_{k,n} \cdot \log_2 \left(1 + \frac{g_k \cdot p_{k,n}}{N_0 \cdot b_{k,n}}\right)}. \tag{2}$$

Since the number of *global* iterations required for the model to converge to ϵ_0 is not known a priori, a greedy approach is followed where, at each iteration n, we attempt to minimize the total energy consumption required by all workers to complete their computational and communication tasks, while a certain local accuracy η is achieved at each worker. This greedy approach is repeated until the global model reaches the pre-selected performance target ϵ_0. Although an FL process lacks theoretical convergence guarantees [19], for the task that we study and the data distribution considered, the convergence to ϵ_0 is always achieved (Section VI). The parameters of the optimization are the computational capacity $f_{k,n} \in \mathbb{R}_+$ and transmission power $p_{k,n} \in \mathbb{R}_+$ of each worker k at each iteration n of the FL process. The objective function is defined as:

$$\min_{\mathbf{f}_n, \mathbf{p}_n} E_n^{gr} = \sum_{k=1}^{\mathcal{K}} (E_{k,n}^C \cdot \Omega_{k,n} + E_{k,n}^T) \tag{3}$$

$$\text{s.t.} \quad \tau_{k,n} + tr_{k,n} < \mathsf{H}, \forall k \in \mathcal{K} \tag{4}$$

$$0 \leq f_{k,n} \leq f_k^{max}, \forall k \in \mathcal{K} \tag{5}$$

$$0 \leq p_{k,n} \leq p_k^{max}, \forall k \in \mathcal{K} \tag{6}$$

$$\sum_{k=1}^{\mathcal{K}} f_{k,n} > 0, \tag{7}$$

where:

$$\tau_{k,n} = \frac{I_{k,n} \cdot \alpha \cdot M_k}{c_{k,n} \cdot f_{k,n}}, \tag{8}$$

$$tr_{k,n} = \frac{m}{r_{k,n}}, \tag{9}$$

$$r_{k,n} = b_{k,n} \cdot \log_2 \left(1 + \frac{g_k \cdot p_{k,n}}{b_{k,n} \cdot N_0}\right). \tag{10}$$

More precisely, $\mathbf{f}_n = [f_{1,n}, f_{2,n}, ..., f_{\mathcal{K},n}]^T$, $\mathbf{p}_n = [p_{1,n}, p_{2,n}, ..., p_{\mathcal{K},n}]^T$, and $\Omega_{k,n}$ (Eq. 11) is an indicator function ensuring that there will be no local training at worker k at the iteration n, in case of zero allocated transmission power:

$$\Omega_{k,n} = \begin{cases} 1, & p_{k,n} > 0 \\ 0, & otherwise \end{cases} \tag{11}$$

Constraint (4) ensures the synchronization of the FL process, by upper bounding the total time required by each worker to complete a computation and transmission task at the time threshold H. Constraints (5) and (6) ensure that the computational capacity along with the transmission power of a worker are within its maximum capabilities, denoted by f_k^{max} and p_k^{max}, respectively. Finally, constraint (7) ensures that at least one worker should be involved in the FL process.

IV. PROPOSED GENETIC ALGORITHM SOLUTION

In this section, a safe GA meta-heuristic approach is proposed. GA provides a feasible solution to strategically perform a global search by means of many local searches, generating high-quality solutions relying on biologically inspired operations, such as parent selection, crossover and mutation [20]. A GA in each generation, constructs a population of *chromosomes*, which is a set of candidate solutions to the optimization problem. The target of the GA is to provide higher quality solutions over the generations using as criterion the fitness score of each chromosome, which represents the target metric of the optimization problem. The main definitions of the GA solution are provided below:

Generation: A *global* iteration n of an FL process.

Gene: A resource assignment to worker k at *global* iteration n, i.e. $[f_{k,n}, p_{k,n}]$, that is bounded according to constraints (5) and (6).

Chromosome: A vector consisting of genes equal to the number of workers, representing a candidate solution to the optimization problem.

Elites: The number of best-scored chromosomes in a generation that are included unaltered in the next generation.

Population: A fixed-size set of chromosomes.

Fitness Function: A function that serves as a score for each chromosome and is formulated based on the objective function (3), in conjunction with constraints (4) and (7). Specifically:

$$FF_n = -[\sum_{k=1}^{\mathcal{K}} (E_{k,n}^C \cdot \Omega_{k,n} + E_{k,n}^T) + v_n] \tag{12}$$

The first part of Eq. (12) is the objective function of our problem formulation. The second part v_n is a penalty term defined to guarantee a safe GA process [21]. As a result, the chromosomes that violate the constraints, wasting resources, will be penalized. We define this penalty as follows:

$$v_n = \sum_{k=1}^{\mathcal{K}} (E_{k,n}^W + \mu_1 \cdot P_{k,n}^{(1)}) + \mu_2 \cdot P_n^{(2)} + P_n^{(3)} \tag{13}$$

where $E_{k,n}^W$ is the wasted energy consumption of worker k, μ_1 and μ_2 are constant penalty weights of each constraint violation, and $P_{k,n}^{(1)}$, $P_n^{(2)}$ are two indicator functions, related to constraints (4) and (7). Considering also the total energy

consumption of a complete FL process, apart from the energy consumed in individual FL rounds, an indicator function $P_n^{(3)}$ is introduced as part of the penalty function. This function penalizes the strategies that result in higher total energy consumption for a complete FL process l, denoted by $E_{FL}^{(l)}$, compared to the previous one ($E_{FL}^{(l-1)}$):

$$P_{k,n}^{(1)} = \begin{cases} 0, & \tau_{k,n} + tr_{k,n} - \mathsf{H} < 0 \\ 1, & otherwise, \end{cases} \quad (14)$$

$$P_n^{(2)} = \begin{cases} 0, & \sum_{k=1}^{\mathcal{K}} f_{k,n} > 0 \\ 1, & otherwise, \end{cases} \quad (15)$$

$$P_n^{(3)} = \begin{cases} -\frac{E_{FL}^{(l-1)} - E_{FL}^{(l)}}{E_{FL}^{(l)}}, & E_{FL}^{(l)} < E_{FL}^{(l-1)} \\ \frac{E_{FL}^{(l)} - E_{FL}^{(l-1)}}{E_{FL}^{(l-1)}}, & otherwise \end{cases} \quad (16)$$

The GA starts with a randomly selected population in the first generation and targets to maximize (12) across generations, which results in minimizing the total energy consumption and the wasted resources, with respect to the constraints of the system.

GA Operations: In each generation a Roulette Wheel method is used for parent selection, along with a uniform crossover and mutation [22] operations, with pre-selected probabilities, named as crossover and mutation rate. Apart from the standard operations, the GA is enhanced with a hybrid operation [23] that combines the Triggered Hyper-mutation (TH) and Fixed Memory (FM). When a change in the system is detected, the TH increases the mutation rate for a certain number of generations, while the FM that uses a fixed-length memory, replaces the worst-scored chromosome from the population with the best individual from the memory. This hybrid operation is triggered when the distance of the fitness score from the best-scoring chromosomes of two consecutive generations is higher than a pre-selected percentage threshold D.

Early Stopping: A termination criterion is applied to GA that interrupts the searching procedure, when the best solution during the evolution process does not improve for a certain number of generations.

Offline and Online Phase: During the offline phase of the GA, in each generation, an FL *global* iteration is executed as many times as the population size, since different candidate solutions are tested. Note that, embedding periodical executions of FL rounds, is of high complexity and time consuming, but most importantly, contrary to the objective of the current work; it requires a significant amount of energy consumption. In order to overcome this issue and as it will be described in Section V, a simulated and significantly simpler FL environment was designed for the GA's offline phase, emulating a real FL process for a given performance target. Finally, the online phase will use real FL processes, in order to evaluate the effectiveness and the robustness of the GA.

V. SIMULATION SETUP

A. Network Environment Setup

The wireless communication network, considered in all performed experiments, is comprised of one coordinator and up to 40 ($|\mathcal{K}| \leq 40$) heterogeneous (in terms of resource capabilities) workers. Of all workers, the 20% are of reduced capabilities and are considered as low-end devices, while the rest are considered as high-end. Each low-end device i, operates with $f_i^{max} = 1$ GHz, $C_i = 4$ [18] and $p_i^{max} = 28$ dBm. Each high-end device j, operates with $f_j^{max} = 3$ GHz, $C_j = 2$ and $p_j^{max} = 33$ dBm. The ς_k is fixed for all workers and equal to 10^{-28} Watt/Hz^3 [16]. We consider non mobile workers and as such all channel gains can be considered as constant. The channel gain is modeled as $g_k = 127 + 30log_{10}(d_k)$, where d_k is the distance of worker k from the coordinator, randomly selected in the range of $[10, 500]$ m and $N_0 = -158$ dBm/Hz [24]. All workers are assigned with a fixed bandwidth $b_k = 20$ MHz.

B. Federated Learning Setup

The FL process considers a Convolutional Neural Network (CNN) for handwritten digit recognition, using the MNIST dataset [25]. The dataset contains 60K samples in total and each worker is assigned with randomly selected $\mathcal{D}_k$ data samples, with $\mathcal{D}_k \in [800, 1200]$. The CNN model comprises of $658,922$ trainable parameters, with size $m = 2.51$ MB and the complexity of the model is $\alpha = 1,800,348$ FLOPS [26]. All local training processes should meet the performance target $\eta = 0.5$. The FL process is considered complete, when the global CNN model reaches the performance target $\epsilon 0 = 0.04$. The time threshold selected for a synchronized FL process is set to $\mathsf{H} = 13$ sec.

C. Genetic Algorithm Setup

Table II includes the configuration of the GA. The maximum number of generations for the different number of workers is set to 5000, while the early stopping is set to 100 generations. The mutation rate, when the hyper-mutation is triggered, is increased by 50% and the size of the fixed memory is set to $[15, 35, 55, 85]$ for the case of $5, 10, 20$ and 40 workers, respectively. The hybrid operation is triggered when D is equal to $[0.4, 0.3, 0.25, 0.2]$ for the four respective cases.

No. of Workers	5	10	20	40
Population Size	40	120	210	220
Elites	10	20	30	60
Crossover Rate	0.3	0.3	0.3	0.3
Mutation Rate	0.1	0.05	0.1	0.05

TABLE II: GA Configuration

As already mentioned in Section IV, for the offline phase of the GA, a simulated FL environment is used, based on a real FL process. From equation (3), it becomes apparent that two FL and energy-related parameters are not known a priori and should be estimated, namely the number of *local* iterations per worker ($I_{k,n}$) and the total number of *global* iterations. Following a statistical analysis of over 100 independent FL executions, the $I_{k,n}$ is selected based on the worker's data samples variance and is in the range $[2, 11]$. Higher data samples variance, means increased data heterogeneity and as a result a higher number of *local* iterations is required to reach η. According to the statistical analysis, the number of *global* iterations is in the range $[10, 22]$, and it is inversely proportional to the number of *local* iterations.

Fig. 1: Offline phase of GA for different number of workers (5, 10, 20, 40)

VI. PERFORMANCE EVALUATION

This section provides numerical results to evaluate the performance of the proposed safe GA based solution. Two main performance metrics are considered in the evaluation phase: **1)** the total energy consumption of the FL (both computation and communication), **2)** the convergence speed of the FL (in terms of training time and *global* iterations).

A. Offline Phase

Figure 1 illustrates the performance of the GA, for different number of workers, as the number of generations increases. Figure 1a depicts the total energy consumption in each generation, that is the sum of the total computation (Fig. 1b) and transmission energies (Fig. 1c). As it can be inferred from Figures (1a - 1c), the amount of consumed energy, for the different number of workers, significantly decreases with the number of generations. More specifically, the GA achieves an overall approximate reduction in the energy consumption up to 75%, 76%, 66% and 56% in the case of 5, 10, 20 and 40 workers, respectively. As the number of workers increases, so does the time (in generations) the GA requires to converge. This relies on the fact that the number of genes increases proportionally to the number of workers, and as a result, the GA requires more exploration time.

In addition, it is intuitive that the higher the number of workers, the higher the total energy consumption of the system. As a result, the fitness score that the GA retrieves is lower for higher number of workers (Fig. 1f). However, as it is depicted in Figure 1d, the GA achieves an average per worker total energy consumption approximately equal to 0.44 Joules. This behavior showcases the scalability of our solution, since the increase in the total amount of energy consumption is mainly based on the number of workers.

As already stated, we introduce a penalty function towards reducing wasted resources of the system. Figure 1e validates our claims, since the total number of constraint violations during an FL process are eliminated. For the most challenging case of 40 workers our solution is still able to provide an efficient strategy reaching in total 1.9 violations, which is translated to 0.04 per worker.

B. Online Phase

The online phase includes the evaluation of the GA strategy produced by the offline phase. All statistical results are averaged, along with their standard deviation (±STD), over 100 independent real FL runs. Firstly, a comparison between our proposed solution and two baseline schemes, namely a Random and a Greedy Selection scheduler [17] is performed, for the 5, 10, 20 and 40 workers.

- **Random Selection Scheduler (RSS):** In each *global* iteration, the RSS orchestrates randomly the resources of each worker, based on their available capacities.
- **Greedy Selection Scheduler (GSS):** In each *global* iteration, the GSS chooses the resource capacities of all workers that led to the best outcome so far, in terms of the total energy consumption.

Table III includes the performance comparison of our solution against the baseline schedulers, with regard to: **1)** the total energy consumption of a complete FL process, **2)** the training duration of each *global* iteration and **3)** the total number of *global* iterations. As it can be deduced from Table III, our proposed solution outperforms all baseline schedulers, in terms of the energy aspect, while also resulting in the least amount of training time per *global* iteration. More precisely, our solution for the case of 5 workers, achieves a significant reduction in the total energy consumption of [83, 56] %, compared to the RSS and GSS, respectively. Similarly, for the case of 10, 20 and 40 workers, our solution results in a similar percentage decrease of [81, 68] %, [76, 66] % and [68, 57] %, respectively. As it appears in the total training time per *global* iteration of the FL process, our solution is on average 20% and 12% faster than the RSS and GSS, respectively. Finally, it should be highlighted that the simulated FL environment (offline phase), achieves a similar performance in the total energy consumption (complete FL process), compared to the real FL executions (online phase) for the same number of *global*

Total Avg. (±STD)	5 Workers			10 Workers			20 Workers			40 Workers		
Schedulers	GA	RSS	GSS	GA	RSS	GSS	GA	RSS	GSS	GA	RSS	GSS
Total Energy (J)	19.3 (±4.5)	116.4 (±23.2)	51.5 (±14.6)	52.8 (±7)	271.1 (±38)	168 (±27.9)	136.7 (±13)	576.4 (±69.4)	400.7 (±52.3)	367.7 (±33.7)	1135.7 (±101.2)	850.2 (±85.9)
Computation Energy (J)	7.5 (±1.7)	51.1 (±12)	25.3 (±7.5)	30.5 (±4.4)	120.6 (±19.7)	112.2 (±20)	61.4 (±6.5)	264.4 ±35.2)	194.2 (±27.8)	166.2 (±17.7)	526.9 (±5)	452.9 (±50.7)
Transmission Energy (J)	11.8 (±3.2)	65.3 (±15.4)	26.2 (±8.4)	22.3 (±3.3)	150.5 (±24.1)	55.8 (±12.3)	75.4 (±8.5)	312 (±43.9)	206.5 (±31.9)	201.6 (±19.9)	608.8 (±59.1)	397.3 (±43.9)
Training Time per Global Iteration (s)	8.1 (±2.5)	9 (±1.2)	8.9 (±2.1)	7.2 (±1.7)	11 (±0.8)	7.6 (±1.5)	8.8 (±1.6)	12.5 (±0.4)	11.9 (±1.1)	12 (±1.4)	12.9 (±0.1)	12.9 (±0.1)
Global Iterations	15.4 (±4)	15.6 (±3.5)	15.7 (±4.4)	17.6 (±2.6)	18.1 (±2.7)	17.4 (±2.8)	18 (±2)	18.7 (±2.5)	18.4 (±2.5)	18.4 (±1.8)	18.3 (±1.7)	18.5 (±2)

TABLE III: Comparison of our solution (GA) against two baseline schedulers (RSS and GSS)

iterations. More specifically, the distance in the total energy consumption between the offline and online phase, is equal to 2%, 8%, 8% and 7%, in favor of the online phase (lower total energy consumption), for the case of 5, 10, 20 and 40 workers, respectively, confirming the effectiveness of the simulated FL environment.

VII. CONCLUSIONS

This paper proposes a safe GA based solution, targeting the minimization of the overall energy consumption of an FL process in a wireless communication network. A simulated FL environment is designed for the GA's offline phase, targeting lower complexity and faster convergence. A penalty function is introduced, towards a safe GA process that almost minimizes the wasted resources of the system. Evaluation results showcase a significant reduction in the overall energy consumption, achieving up to 76% for the offline phase. For the online phase, a decrease of up to 83% in the total energy consumption is achieved compared to two state-of-the-art baseline solutions.

REFERENCES

[1] S. Lange, J. Pohl, and T. Santarius, "Digitalization and energy consumption. does ict reduce energy demand?" *Ecological Economics*, vol. 176, p. 106760, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921800919320622

[2] C. Fabrice, F. Eva, P. Lisa, and inno tsd. 5g and energy efficiency. [Online]. Available: https:\/\/global5g.5g-ppp.eu\/sites\/default\/files\/BookletA4_EnergyEfficiency.pdf

[3] A. S. G. Andrae and T. Edler, "On global electricity usage of communication technology: Trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, 2015. [Online]. Available: https://www.mdpi.com/2078-1547/6/1/117

[4] Mobile net zero: State of the industry on climate action 2022 report. [Online]. Available: https://www.gsma.com/betterfuture/wp-content/uploads/2022/05/Moble-Net-Zero-State-of-the-Industry-on-Climate-Action-2022.pdf

[5] A new industrial strategy for a globally competitive, green and digital europe. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/fs_20_425

[6] A. Kaloxylos, A. Gavras, D. Camps Mur, M. Ghoraishi, and H. Hrasnica, "Ai and ml – enablers for beyond 5g networks," Dec. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4299895

[7] S. Savazzi, V. Rampa, S. Kianoush, and M. Bennis, "An energy and carbon footprint analysis of distributed and federated learning," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2022.

[8] 3GPP, "Technical Specification Group Services and System Aspects; Study of Enablers for Network Automation for the 5G System; Phase 3 (Release 18)," Technical Report 3GPP TR 23.700-80, December 2022.

[9] ETSI, "5G; Architecture enhancements for 5G System (5GS) to support network data analytics services," Technical Specification ETSI TS 123 288 V17.4.0, May 2022.

[10] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[11] Test case definition and test site description part 1. [Online]. Available: https://5gcroco.eu/images/templates/rsvario/images/5GCroCo_D2_1.pdf

[12] J. Ren, J. Sun, H. Tian, W. Ni, G. Nie, and Y. Wang, "Joint resource allocation for efficient federated learning in internet of things supported by edge computing," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[13] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4188–4200, 2019.

[14] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[15] Q. Wang, Y. Xiao, H. Zhu, Z. Sun, Y. Li, and X. Ge, "Towards energy-efficient federated edge intelligence for iot networks," in *2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2021, pp. 55–62.

[16] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *Trans. Wireless. Comm.*, vol. 20, no. 3, p. 1935–1949, mar 2021. [Online]. Available: https://doi.org/10.1109/TWC.2020.3037554

[17] Y. Zhan, P. Li, L. Wu, and S. Guo, "L4l: Experience-driven computational resource control in federated learning," *IEEE Transactions on Computers*, vol. 71, no. 4, pp. 971–983, 2022.

[18] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," *Journal of Communications and Information Networks*, vol. 6, no. 2, pp. 110–124, 2021.

[19] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HJxNAnVtDS

[20] S. Mirjalili, *Genetic Algorithm*. Cham: Springer International Publishing, 2019, pp. 43–55. [Online]. Available: https://doi.org/10.1007/978-3-319-93025-1_4

[21] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, p. 1437–1480, jan 2015.

[22] B. Thomas, "Evolutionary algorithms in theory and practice," p. 120, 1996.

[23] J. Macedo, E. Costa, and L. Marques, "Genetic programming algorithms for dynamic environments"," in *"Applications of Evolutionary Computation*, G. Squillero and P. Burelli, Eds. Cham: Springer International Publishing, 2016, pp. 280–295.

[24] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. M. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1517–1530, 2022.

[25] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[26] Flops library. [Online]. Available: github.com/tokusumi/keras-flops