Optimal Resource Management for Wireless Cooperative Edge Learning with Energy Harvesting

Francesco Binucci[1,2], and Paolo Banelli[1]
[1]Department of Engineering, University of Perugia, Via G. Duranti 93, 06128, Perugia, Italy
[2]Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Viale G.P. Usberti, 181/A, 43124, Parma, Italy

***Abstract*—This paper proposes a general framework, where energy-harvesting devices may cooperate with one another to perform their learning tasks, in a peer-to-peer fashion. Unlike the classical edge-inference scenarios, where inference tasks are split between an edge device and an edge server (ES), our model considers a peer-to-peer (P2P) wireless network, where each mobile device (MD) can operate both as a client and as a server for other nodes. The framework first establishes node pairs on the basis of the channel state information. Within each pair, nodes can either process their tasks locally or offload them to the associated node, allocating both transmission and computational resources through a Lyapunov optimization procedure. Simulation results, targeted to an image classification task, validate the effectiveness of the proposed algorithm and highlight its potential for broader applications in various scenarios and use cases.**

***Index Terms*—Cooperative Inference, resource allocation, Lyapunov optimization**

I. Introduction

Edge Intelligence [1] has recently emerged as a crucial technology for delivering cost-effective, low-latency machine learning, and artificial intelligence services in next-generation mobile networks. Several studies highlight how computational offloading at the network edge enables ML services with lower latency and reduced energy consumption across various scenarios and use cases, including the Internet of Things [2], Industry 4.0 [3], and vehicular communications [4].

Ensuring service sustainability, particularly for devices with limited energy and computational resources, requires dynamic resource management that balance inference accuracy, latency, and energy consumption. In recent years, significant research efforts have been dedicated to developing such strategies [5].

Related Works. Several resource allocation frameworks have been proposed to execute cooperative tasks at the network edge [6]–[9]. Authors in [10] considered dynamic resource management for Industrial IoT applications, to optimize learning tasks performed by deep neural networks (NNs), in mobile edge computing scenarios. Similarly, [11] explores resource allocation based on Lyapunov optimization (LO) [12], to achieve an optimal trade-off between energy consumption, latency, and inference accuracy at the network edge. In [13], a LO-based resource management is applied to decentralized estimation in energy-harvesting networks, while [14] investigates an edge-inference scenario designed for ultra-reliable and low-latency communications in vehicular networks. LO-based strategies, can also handle resource management for mobile edge-learning within Goal-Oriented Communications (GOC) [15], as in [16], [17], [18], and references therein.

Our contributions. Rather than assuming the use of edge servers (ESs) like in the aforementioned works, we consider a peer-to-peer (P2P) network where each mobile device (MD) is equipped with multiple inference models, each offering a different trade-off between computational complexity and learning accuracy. Furthermore we assume each MD can simultaneously act both as a client and a server. Simulations, for image classifications by Convolutional NNs (CNNs), validate the proposed framework, highlighting its potential and motivating further investigation also in other scenarios.

II. System Model

We consider a P2P wireless network with K MDs, where resource management evolves in a discrete time fashion, on time slots indexed by t, with a fixed duration τ.

In any time slot, a MD can generate $A^k(t)$ tasks, denoted as $\mathcal{B}^k(t) = \{\omega(i,t)\}_{i=1}^{A^k(t)}$, e.g., a set of images indexed by i.

Each MD can either locally process its own tasks, or partially offload the inference to another MD, or queuing the tasks to be processed/offloaded in a future slot $T_{\rm dec}(t) \geq t$. Each MD is equipped with a set $\mathcal{S}^k$ of inference models, each one characterized by a different trade-off between inference accuracy and complexity. We encode the decisions to offload some tasks during the t-th time slot by

$$I^k(t) = \begin{cases} 1, \text{ if MD } k \text{ offloads data at the } t\text{-th slot} \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

To simplify management, we assume that at each time slot the offloading can be performed among pairs of nodes, with a pairing that is encoded in the matrix $\mathbf{M}(t)$, defined as

$$[\mathbf{M}(t)]_{k,k'} = \begin{cases} 1, \text{ if MD } k \text{ is associated with MD } k' \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

The connectivity matrix $\mathbf{M}(t)$ may be established by either a centralized or distributed policy. We assume a centralized controller with perfect knowledge of all the network links. The controller updates the connectivity matrix once every S time slots according to the policy described in the next sections.

This work was supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP F83C22001690001/E83C22004640001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

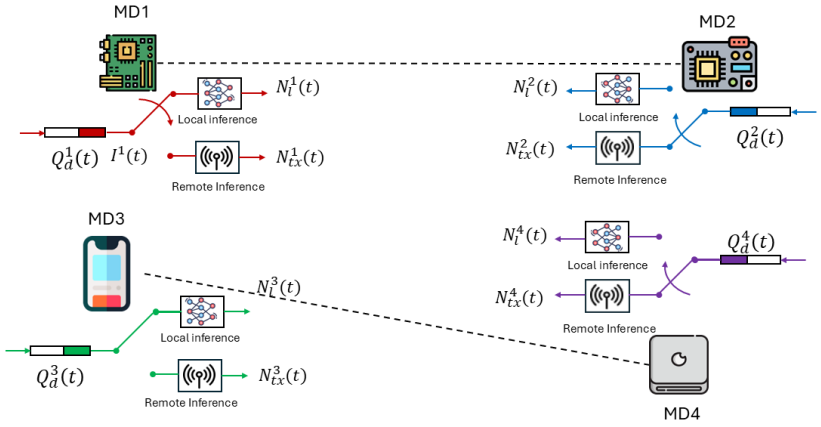

Fig. 1. System model. Edge Devices are associated in pairs. Each MD may decide to process the data locally or to offload a a task to its associated MD.

Notably, adjusting the refresh rate S enhances the algorithm's ability to adapt to non-stationary environments.

Offloading variables and connectivity matrix are linked by

$$I^k(t) + I^{k'}(t) \leq 1, \quad \forall k, k' : [\mathbf{M}(t)]_{k,k'} = 1, \tag{3}$$

to ensure that, at any time slot, the offloading between couple of nodes is mono-directional, e.g., either $k \to k'$ or $k' \to k$.

A. Latency Model

At any time slot, the k-th MD may transmit (offload) a number of tasks given by

$$N_{tx}^k(t) = \left\lfloor \frac{\tau R^k(t)}{W^k} \right\rfloor, \tag{4}$$

where W^k are the bits required to encode the task, while $R^k(t)$ is the bit rate assigned by the management policy, depending on the channel between MDs k and k', and the trade-offs between batteries energy, latency, and learning precision.

We assume that the remotely offloaded inference proceeds in parallel with the transmission, as new tasks are received. Furthermore, we assume that the offloaded tasks have to be immediately processed by the remote MD k', which can also process part of its previously buffered tasks, according to

$$N_l^{k'}(t) = \left\lfloor \frac{\tau f_d^{k'}(t) \beta^k}{F(\rho^{k'})} \right\rfloor, \tag{5}$$

where $f_d^{k'}(t)$ is the MD clock frequency, $\beta^{k'}$ converts number of FLOPs in clock cycles, while $F(\rho^{k'})$ are the FLOPs needed to process a task of the k'-th MD through the NN $\rho^{k'} \in \mathcal{S}^{k'}$.

Assuming that each k'-th MD may operate with a clock frequency $0 \leq f_d^{k'}(t) \leq f_{\max}^{k'}$ for all t, part of the computational capabilities of the MD can be assigned to complete tasks of the other MD it is paired to. We thus define $f_{k'}^k(t) = f_{\max}^{k'} - f_d^{k'}(t)$ the fraction of its clock frequency the k'-th MD may reserve for tasks possibly offloaded by the k-th MD. Thus, the *maximum* number $N_{k'}^k(t)$ of tasks the k'-th MD may process for the k-th one is given by

$$N_{k'}^k(t) = \left\lfloor \frac{\tau f_{k'}^k(t) \beta^{k'}}{F(\rho^k(t))} \right\rfloor. \tag{6}$$

In practice, it makes sense to impose the following constraints to the resource management policy:

1) $N_{tx}^k(t) \leq N_{k'}^k(t)$, to limit the number of transmitted tasks to the maximum number of tasks that the destination MD can process at the t-th slot.
2) The feasible transmission rates have to be high enough to guarantee that at least a task can be transmitted within the useful portion $\delta_{min}^k(t) = \tau - F(\rho^k)/(f_{k'}^k(t)\beta^{k'})$ of the time slot, where $F(\rho^k)/(f_{k'}^k(t)\beta^{k'})$ is the time the destination MD needs to take a decision on the task.

This way we ensure that the time required to offload and process a task will not exceed the duration of a time slot. Thus, defining $R_{min}^k(t) = W^k/\delta_{\min}^k(t)$ and $R_{\max}^{k+}(t) = (W^k N_{k'}^k(t))/\tau$, the transmission rate will be subject to

$$R_{\min}^k(t) \leq R^k(t) \leq R_{\max,s}^k(t). \tag{7}$$

Lets define the quantity

$$N_d^k(t) = I^k(t) N_{k'}^k(t) + (1 - I^k(t)) N_l^k(t), \tag{8}$$

which quantifies the number of tasks the k-th MD processes either remotely, or locally, during the t-th slot.

According to Lyapunov optimization [12], we model the latency terms considering a set of service queues. Specifically,

$$Q_d^k(t+1) = \max(0, Q_d^k(t) - N_{tx}^k(t) I^k(t) - N_l^k(t) \overline{I^k}(t)) + A^k(t) \tag{9}$$

denotes the evolution of the queue for the k-th user.

Assuming that the arrival process $A^k(t)$ has a stationary arrival rate λ^k, and defining $\overline{A^k} = \lambda^k/\tau$, exploiting the Little's Law [19], the edge-to-edge latency can be expressed as $D_{\text{avg}}^k = Q_{\text{avg}}^k/\overline{A^k}$. Thus, the long-term latency constraint can be formalized as a constraint on the average queue length

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{Q_d^k(t)\} \leq Q_{\text{avg}}^k. \tag{10}$$

B. Energy Model

The energy model considers the energy spent both for transmissions and computations. The matrix $[\mathbf{H}(t)]_{k,k'}$ stores the wireless channel $H_{k,k'}(t)$ between each pair of nodes.

Assuming that each pair of MDs do not interfere with the other pairs, the energy spent for transmission can be derived from Shannon capacity in AWGN as [20]

$$E_{tx}^k(t) = \frac{\tau B^k N_0}{|H_{k,k'}(t)|^2}\left(\exp\left(\frac{R^k(t)\ln(2)}{B^k}\right)-1\right). \quad (11)$$

The energy spent for computation at the t-th slot for the k-th MD can be modeled by [21]

$$E_l^k(t) = \tau\kappa_k f_k(t)^3, \quad (12)$$

where $f_k(t)$ is the clock frequency of the k-th MD, and κ_k denotes the effective switched capacitance of the processor.

Thus, the total energy spent by a MD is $E_{\text{tot}}^k(t) = E_{tx}^k(t) + E_l^k(t)$, and the overall system consumes an energy

$$E_{\text{tot}} = \sum_{k=1}^{K} E_{\text{tot}}^k(t). \quad (13)$$

We further assume that each MD can harvest energy from the environment, and the battery level evolves as

$$J^k(t+1) = \min(J_{\max}^k, \max(J^k(t) - E_{tot}^k(t), 0) + \chi^k(t)), \quad (14)$$

where $J_{\max}^k$ is the maximum battery level, and $\chi^k(t)$ is the energy harvested at the t-th slot by the k-th MD. We impose the following long-term constraint on the average battery level

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{J^k(t)\} \geq J_{\text{avg}}^k \quad (15)$$

C. Inference Performance

Once the inference decisions are taken, we assume to have access to a feedback G^k related to a specific performance metric (e.g., inference accuracy). Specifically, we are interested in maximizing the following average objective

$$G_{\text{tot}}(t) = \sum_{k=1}^{K}\frac{1}{N_d^k(t)}\sum_{i=1}^{N_d^k(t)} G^k(\omega(O(i), T_{\text{gen}}(t)) \quad (16)$$

where $O(i)$ is the task id at the top of queue $Q_d^k(t)$, while $T_{\text{gen}}(t)$ is the birth time of the task decided at the t-th slot.

III. Problem Formulation and Solution

The system model we described so far is then employed in the following long-term resource allocation problem

$$\begin{aligned}
\underset{\mathbf{\Phi}(t)}{\text{maximize}} \quad & \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{G_{\text{tot}}(t)\} \\
\text{subject to} \quad & \text{(a) long-term latency (10) } \forall k, \\
& \text{(b) long-term battery level (15) } \forall k, \\
& \text{(c) } I^k(t) + I^{k'}(t) \leq 1, \forall k, k' : \mathbf{M}_{k,k'}(t) = 1, \\
& \text{(d) } 0 \leq E_{\text{tot}}^k(t) \leq J^k(t)\ \forall k, t, \\
& \text{(e) } R_{\min}^k(t) \leq R^k(t) \leq R_{\max}^k(t),\ \forall k, t, \\
& \text{(f) } 0 \leq f_d^k(t) \leq f_{\max}^k,\ \forall k, t
\end{aligned} \quad (17)$$

where $\mathbf{\Phi}(t) = \{R^k(t), f_d^k(t), I^k(t), \rho^k(t)\}_{k=1}^K$ is the set of the optimization variables at each time slot. Our aim is to maximize the average inference performance under (a) average latency and (b) battery level constraints. Constraint (c) imposes a unique direction for the flow of the offloaded tasks, while constraint (d) limits the energy consumption in each time slot. Finally, constraints (e) and (f) define the feasible transmission rates and clock frequencies.

A. MD Association Algorithm

The first aim of our resource allocation policy is defining the association matrix $\mathbf{M}(t)$. To this end, we firstly organize the time slots in frames, indexed by $f \in \mathbb{N}_0$ and characterized by a fixed duration S. We assume that the association matrix is constant within each time frame, i.e., $\mathbf{M}(t) = \mathbf{M}(fS+1)$ for each $t \in [fS+1, (f+1)S]$. Then, at the end of each time frame, the centralized controller establishes the MD pairs on the basis of Algorithm1. At each step, Algorithm 1 associates the pair of MDs characterized by the best channel. Of course, many other resource allocation policies may be considered, and their investigation is left for future studies.

Algorithm 1: MD association algorithm

Input: Channel State Matrix $\mathbf{S} = \mathbf{H}(t)$;
Initialize $\mathbf{M} = \mathbf{0}$
for $k = 0 \dots K/2$ **do**
  (i^*, j^*)=$\max_{i,j}\mathbf{S}$
  set $\mathbf{M}(i^*, j^*) = \mathbf{M}(j^*, i^*) = 1$
  null rows and columns i^* and j^* of $\mathbf{S}$
end for
return $\mathbf{M}$

B. Lyapunov Based Solution

Following standard Lyapunov optimization [12], we associate to each long-term constraint a virtual queue, that evolve according to

$$\begin{aligned}
Z^k(t+1) &= \max(0, Z^k(t) + \mu^k(Q_d^k(t+1) - Q_{\text{avg}}^k)) \\
H^k(t+1) &= \max(0, H^k(t) + \nu^k(J_{\text{avg}}^k - J^k(t+1))),
\end{aligned} \quad (18)$$

where μ^k and ν^k are step-sizes used to control the convergence speed of the algorithm. Overall violations of the long-term constraints are captured by the Lyapunov function [12]

$$L(t) = \frac{1}{2}\sum_{k=1}^{K}[Z^k(t)^2 + H^k(t)^2], \quad (19)$$

whose expected change between consecutive time slots gives the Lyapunov drift plus penalty function (LDPP)

$$\Delta_p(t) = \mathbb{E}\{L(t+1) - L(t)|\mathbf{\Theta}(t)\} - V\mathbb{E}\{G_{\text{tot}}(t)|\mathbf{\Theta}(t)\}, \quad (20)$$

where $\mathbf{\Theta}(t) = \{H^k(t), Z^k(t)\}_{k=1}^K$. Exploiting some upper bounds [12], herein omitted due lack of space, and defining $\tilde{Q}_d^k(t) = (\mu^k)^2 Q_d^k(t) + \mu^k Z^k(t)$, and $\tilde{H}^k(t) = \mu^k H^k(t) -$

$(\nu^k)^2 J^k(t)$ we end up with the following instantaneous problem, where we omit the time index t to ease the notation

$$\min_{\mathbf{\Phi}} \sum_{k=1}^{K} -\tilde{Q}_d^k[N_{tx}^k I^k + N_l^k \overline{I^k}] - VG^k + \tilde{H}^k \tau \kappa_k (f_d^k)^3 + \tilde{H}^k \frac{\tau B^k N_0}{|H_{k,k'}|^2} \left(\exp\left(\frac{R^k \ln 2}{B^k} \right) - 1 \right) \quad \text{s.t. } (17d) - (17g) \tag{21}$$

The solution to the problem presents significant challenges due to two primary reasons:

1) The lack of a closed-form expression for the term G^k, capturing the performance of the inference model.
2) The decision variables I^k and the learning model selection variable ρ^k introduce mixed-integer complexity to the problem.

To address the first challenge, following an approach similar to [17], we introduce a surrogate function $\tilde{G}(\rho^k)$ that models the average inference performance for each inference model ρ^k used by the k-th MD. This surrogate function drives the optimization process, while still leveraging the actual feedback $G^k(t)$ to asses the algorithm performance. Regarding the second challenge, we note that the problem is separable across connected user pairs (k, k'). For each pair, the integer variables can be evaluated exhaustively. Specifically, assuming a fixed inference model ρ^k, we determine the optimal local clock frequencies for the MD pair (k, k') when local inference is used (i.e., $I^k = I^{k'} = 0$):

$$f_d^{k^*}(\rho^k) = \left[\sqrt{\frac{\tilde{Q}_{k'}^d \beta^k}{3\kappa^k F(\rho^k) \tilde{H}^k}} \right]_0^{f_{\max}^{k+}}, \tag{22}$$

where $f_{\max}^{k+}$ is the minimum between the maximum allowable clock frequency for the MD and the maximum sustainable clock frequency given the battery level $J^k(t)$.

When $I^k = 1$, we assume offloading follows the direction $k \to k'$ (a similar approach applies for $I^{k'} = 1$), and the optimal transmission rate for the k-th MD is given by

$$R^{k*}(\rho^k) = \left[\frac{B^k}{\ln 2} \ln\left(\frac{\tilde{Q}_d^k |H_{k,k'}|^2}{\ln(2) \tilde{H}^k W^k N_0} \right) \right]_{R_{\min}^k}^{R_{\max}^k} \times \mathbb{I}\{\tilde{Q}_d^k > 0\}, \tag{23}$$

where $R_{\max}^k$ is the minimum between $R_{\max,s}^k$ and the maximum transmission rate allowed by the energy budget $J^k(t)$.

The optimal solution of the problem can be evaluated by testing all the possible inference models employable by both the devices in case of no-offloading, $k \to k'$ offloading, and $k' \to k$ offloading, and then selecting the offloading decisions I^k and $I^{k'}$ and learning models ρ^k, $\rho^{k'}$ leading to the lowest joint cost for the MD pair.

IV. Simulation Results and Conclusion

We run the optimization algorithm for $N = 10,000$ time-slots, with a time slot duration $\tau = 10$ ms, and considering $K = 4$ MDs in the scenario, all with the same maximum battery level $J_{\max}^k = 80$ mJ, and the same long-term battery constraint $J_{\text{avg}}^k = 50$ mJ, and average latency constraint $D_{\text{avg}}^k = 20$ ms. The MD association matrix $\mathbf{M}(t)$ is computed every $S = 10$ slots.

Computing model: the maximum clock frequency $f_{\max}^k$ is set to 2.8 GHz for MD2 and MD3 and to 2 GHz for MD1 and MD4. The number of FLOPs per clock cycles β^k is set to $\{10, 30, 30, 10\}$ for the 1st, the 2nd, the 3rd and the 4th MD respectively. We assumed an effective switched capacitance $\kappa^k = 1.097 \times 10^{-27}$ $[\frac{s}{cycles}]^3$ for all the MDs.

Channel Setting: we set for all the MDs the same bandwidth $B^k = 20$ MHz, noise power spectral density $N_0 = -174$ dBm/Hz, and maximum transmit power $p_{\max}^k = 3.5$ W. The channel $|H_{k,k'}(t)|$ between each device pair is assumed to be stationary and Rayleigh distributed, with average path losses $E\{|H_{k,k'}(t)|^2\}$ reported in Tab. I. For all the MDs, we assumed images encoded in 8-bit format, composed $128 \times 128 \times 3$ pixels, with a resulting size of $W^k = 48$ KB.

TABLE I
AVERAGE PATH LOSS FOR CHANNELS BETWEEN DIFFERENT MDS.

	MD1	MD2	MD3	MD4
MD1	N/A	90 dB	120 dB	130 dB
MD2	90 dB	N/A	120 dB	110 dB
MD3	120 dB	120 dB	N/A	100 dB
MD4	130 dB	110 dB	100 dB	N/A

Inference task description: we considered an image classification task based on the Intel Image Classification dataset [22], composed of 17,000 RGB images belonging to 6 different landscapes, and divided into 11,000 images for training and 3,000 images for validation and test sets. Tab. II reports validation accuracy and complexity of the CNNs employed at the different MDs [1][2].

TABLE II
AVERAGE ACCURACY AND COMPLEXITY OF THE INFERENCE NNS.

Model Name	Validation Accuracy [%]	Complexity [MFLOPs]
MobileNetv3 small	89.5	21.23
MobileNetv3 large	91.5	79.54
EfficientNet	93.2	137.76

Experimental Evaluation: Figure 2 shows the trade-off between average latency and accuracy for all the MDs. The curves have been obtained for increasing values of the trade-off parameter $V \in \{1 \times 10^1, 1 \times 10^2, \ldots, 1 \times 10^6\}$ (cf (20)), averaging latency and accuracy over the last 1,000 slots. We compared the proposed cooperative strategy (solid lines) with a full local computation scenario (dashed lines) considering the arrival rates $A_{\text{avg}}^k \in \{3, 4\}$ tasks per slot. We note that both the strategies satisfy the average latency constraint, represented by the red dashed line. Improving inference performance requires models with higher computational complexity, leading to increased latency. Furthermore, as the arrival rate increases, average accuracy decreases, since the strategies are pushed to

[1]Data in Tab. II model the terms $F(\rho^k)$ in (5), (6), and the surrogate objective function $\tilde{G}(\rho^k)$.

[2]CNNs implementations are provided by https://pytorch.org/hub/

select less powerful models to satisfy the latency constraint. The cooperative approach achieves better latency vs inference accuracy trade-offs, as indicated by the higher correct classification rates of the solid curves. Indeed, thanks to offloading, MDs with less computational power can exploit more powerful CNNs, deployed at other MDs, improving the accuracy. This is confirmed by Table III, reporting the offloading percentages of the various MDs for the realizations with the highest inference performance (i.e., the rightmost points in Figure 2). Notably, the majority of offloading decisions are made by MD1 and MD4, as they are equipped with the least powerful processors.

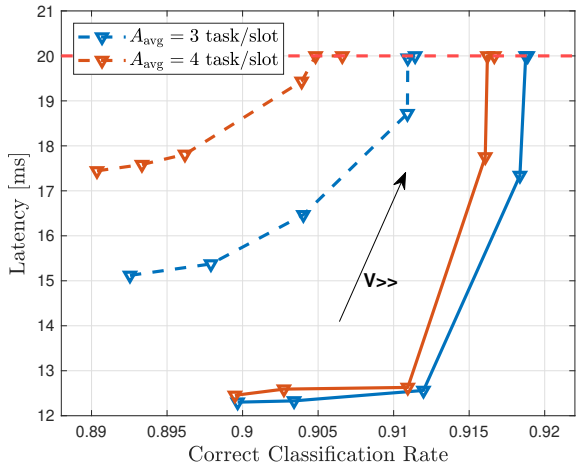

Fig. 2. Accuracy vs Latency trade-off for cooperative strategy (solid) and full local computation (dashed).

TABLE III
OFFLOADING PERCENTAGES FOR THE DIFFERENT MDS ($V = 1 \times 10^6$).

A^k_{avg} [task/slot]	MD1 [%]	MD2 [%]	MD3 [%]	MD4 [%]
3	75.9	2.1	1.1	73
4	77.7	4.5	3.2	71.3

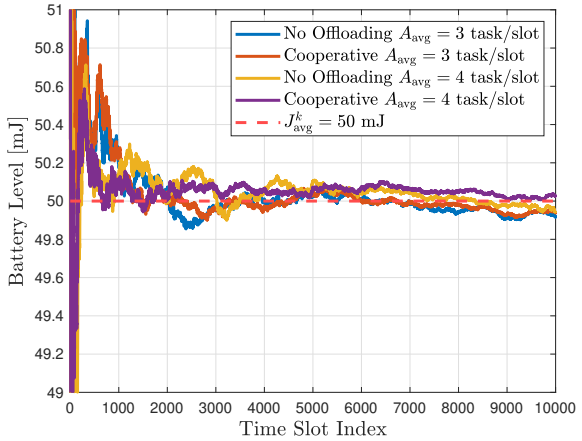

Fig. 3. Battery level as a function of the time slot ($J^k_{\text{avg}} = 50$ mJ).

Figure 3 shows the long-term battery level over time in the cooperative and the full local scenarios for the best accuracy realizations (i.e., the rightmost points in Figure 2). We note that also the long-term constraint in (15) is always satisfied.

To conclude, we presented a dynamic resource allocation strategy for cooperative edge-inference with energy harvesting devices. Simulations testified its effectiveness in optimizing latency vs accuracy trade-offs. Future research may explore non-ideal communication scenarios and specific applications, such as goal-oriented communications.

REFERENCES

[1] Z. Zhou, X. Chen, E. Li, *et al.*, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[2] S. U. Amin and M. S. Hossain, "Edge intelligence and internet of things in healthcare: A survey," *IEEE access*, vol. 9, pp. 45–59, 2020.

[3] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi, and X. Guan, "Industrial edge computing: Enabling embedded intelligence," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 48–56, 2019.

[4] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246–261, 2019.

[5] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–37, 2019.

[6] X. Ye, Y. Sun, D. Wen, G. Pan, and S. Zhang, "End-to-end delay minimization based on joint optimization of dnn partitioning and resource allocation for cooperative edge inference," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pp. 1–7, IEEE, 2023.

[7] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *2017 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7, IEEE, 2017.

[8] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE transactions on wireless communications*, vol. 16, no. 9, pp. 5994–6009, 2017.

[9] Z. Ning *et al.*, "Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution," *IEEE Tran. on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2212–2225, 2021.

[10] W. Wu, P. Yang, W. Zhang, *et al.*, "Accuracy-guaranteed collaborative dnn inference in industrial iot via deep reinforcement learning," *IEEE Tran. on Industrial Informatics*, vol. 17, no. 7, pp. 4988–4998, 2020.

[11] M. Merluzzi, P. Di Lorenzo, and S. Barbarossa, "Wireless edge machine learning: Resource allocation and trade-offs," *IEEE Access*, vol. 9, pp. 45377–45398, 2021.

[12] M. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Springer Nature, 2022.

[13] C. Battiloro, P. Di Lorenzo, *et al.*, "Dynamic resource optimization for decentralized estimation in energy harvesting iot networks," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8530–8542, 2020.

[14] Z. Wang, A. E. Kalør, *et al.*, "Ultra-low-latency edge inference for distributed sensing," *arXiv preprint arXiv:2407.13360*, 2024.

[15] E. C. Strinati and S. Barbarossa, "6g networks: Beyond shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107930, 2021.

[16] P. Di Lorenzo, M. Merluzzi, F. Binucci, *et al.*, "Goal-oriented communications for the iot: System design and adaptive resource optimization," *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 26–32, 2023.

[17] F. Binucci, P. Banelli, *et al.*, "Adaptive resource optimization for edge inference with goal-oriented communications," *EURASIP Jour. on Adv. in Sig. Proc.*, vol. 2022, no. 1, p. 123, 2022.

[18] F. Binucci, P. Banelli, *et al.*, "Multi-user goal-oriented communications with energy-efficient edge resource management," *IEEE Tran. on Green Comm. and Networking*, vol. 7, no. 4, pp. 1709–1724, 2023.

[19] J. D. Little, "A proof for the queuing formula: L= λ w," *Operations research*, vol. 9, no. 3, pp. 383–387, 1961.

[20] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[21] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 13, no. 2, pp. 203–221, 1996.

[22] P. Bansal, "Intel image classification: Image scene classification of multiclass," 2019.