Efficient Offloading in UAV-MEC IoT Networks: Leveraging Digital Twins and Energy Harvesting

Mehak Basharat*, Muhammad Naeem†, Asad M. Khattak‡, and Alagan Anpalagan*,
* Dept. Electrical, Computer, and Biomedical Engineering, Toronto Metropolitan University, Toronto, ON, Canada.
† Department of Electrical Engineering, COMSATS University Islamabad, Wah Campus, Pakistan.
‡ College of Technological Innovation, Zayed University, Abu Dhabi, U.A.E.
(e-mail: {mehak.basharat;alagan}@torontomu.ca, muhammadnaeem@gmail.com, asad.khattak@zu.ac.ae).

***Abstract*—Digital twin technology leverages a real-time simulated environment to optimize unmanned aerial vehicles (UAVs)–mobile edge computing (MEC) networks. Considering unpredictable MEC environments and low-power Internet of things (IoT) devices, this paper proposes a digital twin-assisted task offloading scheme in UAV-MEC networks with energy harvesting. The goal is to minimize latency and maximize the number of associated IoT devices by optimizing UAV placement and IoT device association. The constraints on computing, caching, energy harvesting, latency, and the maximum number of IoT devices a UAV can serve are considered. To solve the formulated problem, we employ a branch and bound algorithm to obtain optimal results. Additionally, we propose a relaxed heuristic algorithm to solve the problem with reduced computational complexity. This approach provides efficient alternatives to obtain near-optimal solutions. Through simulations, we demonstrate the effectiveness of the heuristic algorithm and validate the benefits of leveraging digital twin technology in UAV-MEC networks with energy harvesting.**

***Index Terms*—Digital twin, energy harvesting, mobile edge computing, task offloading, and unmanned aerial vehicles.**

I. Introduction

Unmanned aerial vehicles (UAVs) equipped with mobile edge computing (MEC) capabilities can be used to provide coverage to ground Internet of things (IoT) devices by enabling real-time data processing and analysis at the edge of the network [1]. Thus, IoT devices can offload their tasks to UAV-MEC to operate more efficiently and extend their battery life by harvesting energy. This is especially important as IoT devices often have limited resources, such as computing power, caching, and energy. However, ensuring the effective and reliable operation of UAV-MEC networks requires a comprehensive understanding of the complex interactions and dependencies between all system components. Moreover, the future sixth-generation (6G) networks are expected to be highly dynamic and heterogeneous, with complex demands from emerging applications, such as ultra-low latency and real-time network topology changes [2]. This complexity and management costs increase as the number of connected devices grows, which is expected in future wireless networks.

Digital twin is an emerging technology that creates a virtual layer of the physical network and its components, enabling real-time simulation, monitoring, and optimization of UAV-MEC networks. Digital twins can make data and task offloading decisions in a simulated environment [3]. Computing, caching, and communication resources can be allocated more quickly and accurately based on IoT device requests with the help of digital twins [4]. The digital twin model constantly monitors the physical network, providing IoT devices with perceptual data to make more timely and accurate offloading decisions [5]. This ultimately improves the energy efficiency and performance of the system, especially when combined with energy harvesting in a digital twin-assisted UAV-MEC network. In [6], the digital twin technology is explored for task offloading in the MEC network to reduce time overhead and power consumption. The digital twin concept can also enable edge networks in the IoT environment [7].

IoT devices in practical systems randomly generate computing tasks, and the digital twin can simulate the behaviour and performance of IoT devices and UAV-MECs to predict energy consumption and task completion time accurately. This prediction can reduce energy consumption and extend mission time duration by offloading tasks between IoT devices and UAV-MECs based on computing infrastructure. UAV operating efficiency is further increased with energy harvesting technology, and energy can be used to power IoT devices. Nevertheless, there are challenges with deploying digital twins, such as estimation errors and choosing appropriate scenarios for IoT devices and UAV-MEC platforms. Therefore, when the number of IoT devices increases, it is imperative to explore the UAV-MEC networks functioning with digital twins to guarantee the quality of service. This paper investigates the concept of a digital twin consisting of IoT devices and UAV-MECs, where IoT devices randomly generate computing tasks. The objective is to minimize latency and associate the maximum number of IoT devices with UAVs by optimizing UAV placement and IoT device connection.

The rest of the paper is structured as follows: Section II presents the system model and problem formulation. Section III describes the solution approaches for the formulated optimization problem. Section IV presents simulation results to show the effectiveness of the proposed scheme compared to optimal results. Finally, Section V concludes the paper.

II. System Model and Problem Formulation

We consider a UAV-MEC network that consists of a real physical layer with M number of UAV-MECs providing coverage to N number of IoT devices distributed randomly

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00094

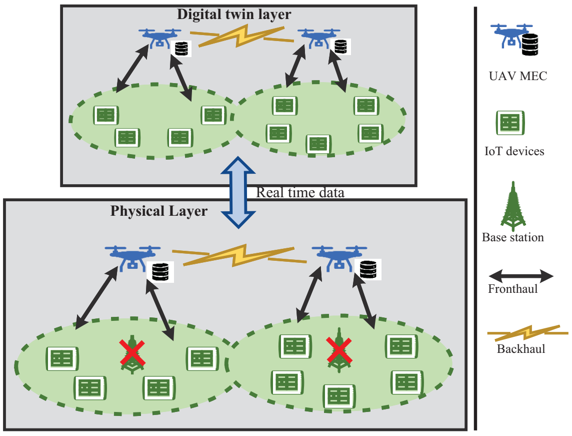


Fig. 1. System model for digital twin-assisted task offloading.

on the ground as illustrated in Fig. 1. To account for the dynamic nature of UAV-MEC networks, we integrate a digital map of the UAV-MECs, IoT devices, and the communication environment, which constitutes the digital layer. The digital twin system stores network entity parameters and monitors the network's current operating state. Real-time channels allow the network entities (i.e., UAV-MECs and IoT devices) to send their current running status and topology to the digital twin [8], [9]. The digital twin representation of IoT devices and UAV-MECs may have a deviation from their actual state values, which can be either positive or negative. For example, the digital twin server provides an estimated CPU frequency to define the deviation between UAV-MEC and its digital twin [5], [8]. The task from *n-th* IoT device is defined as $\mathcal{T}_n = (\mathcal{S}_n, \mathcal{D}_n, \mathcal{W}_n, \xi_n)$, where $\mathcal{S}_n$, $\mathcal{D}_n$, $\mathcal{W}_n$, and ξ_n represent the data size (in bits) to be computed, computation deadline, computation workload (in CPU cycles/bit), and residual energy of *n-th* IoT device, respectively.

The coordinates of *m-th* UAV and *n-th* IoT device are represented as $L_m = (x_m, y_m, h_m)$ and $L_n = (x_n, y_n, 0)$ in a three-dimensional (3D) Cartesian coordinate system respectively, where x and y are horizontal coordinates and h is the height of the UAV. The distance between the *n-th* IoT device and *m-th* UAV is denoted by $d_{n,m}$. When *n-th* IoT device offloads data to hovering *m-th* UAV, it will need communication resources. The instantaneous data rate of *n-th* IoT device can be written as:

$$R_{n,m} = B \log_2 \left(1 + \frac{P_n^{IoT} g_{n,m}}{\sigma_n^2}\right), \ \forall \ n, m, \tag{1}$$

where B represents channel bandwidth, P_n^{IoT} denotes the transmit power of *n-th* IoT device, σ_n^2 is the average power of the Gaussian noise at *n-th* IoT device, and $g_{n,m} = (\Upsilon_{n,m})^{-1}$ is the channel gain between the *n-th* IoT device and *m-th* UAV in which $\Upsilon_{n,m}$ is the average path loss consisting of the probability of line of sight (LOS) and non-line of sight (NLOS) links between *n-th* IoT device and *m-th* UAV [10], [11]. It is assumed that the available spectrum is divided into orthogonal resource blocks to avoid user interference.

The transmission delay in offloading data can be represented as: $\mathcal{L}_{n,m}^{O} = \frac{\mathcal{S}_n}{R_{n,m}}$, $\forall \ n, m$. When offloading a task, it may require computation resources, i.e., $\mathcal{D}_n \neq 0$ and $\mathcal{W}_n \neq 0$. The estimated time to perform the task on UAV-MEC can be written as: $\tilde{\mathcal{L}}_{n,m}^{UAV} = \frac{\mathcal{S}_n \mathcal{W}_n}{f_m^{UAV}}$, where f_m^{UAV} is the estimated CPU frequency (cycles per second) of *m-th* UAV.

We assume that the deviation $\overline{f_m^{UAV}}$ can be obtained beforehand to calculate the computation latency gap between the real value and estimation of the digital twin. This can be calculated as:

$$\Delta\mathcal{L}_{n,m}^{UAV} = \frac{\mathcal{S}_n \mathcal{W}_n \overline{f_n^{UAV}}}{f_m^{UAV}(f_m^{UAV} - \overline{f_m^{UAV}})}. \tag{2}$$

Then, the actual computation latency for performing the task at UAV-MEC is given as: $\mathcal{L}_{nm}^{UAV} = \tilde{\mathcal{L}}_{n,m}^{UAV} + \Delta\mathcal{L}_{n,m}^{UAV}$.

The energy harvested by the *n-th* IoT device from *m-th* UAV can be written as: $\mathcal{E}_{n,m} = \left[(P_m^{UAV} \times g_{n,m}) \times \gamma_{n,m}\right] \times \eta_{n,m}$, where the efficiency of energy harvesting is denoted by $\eta_{n,m}$ and is dependent on the harvesting circuit. $\gamma_{n,m}$ represents the harvesting duration and P_m^{UAV} is the transmit power of the UAV for energy harvesting. We consider energy harvesting latency $\mathcal{L}_{n,m}^{H}$ to be inverse of residual energy and equal to the duration of harvesting duration $\gamma_{n,m}$. An IoT device's residual and harvested energy must be greater than the threshold E_{TH} to offload data. We calculate the deviation in residual energy $\tilde{\xi}_n \zeta_n^E$, where $\tilde{\xi}_n$ denotes the percentage deviation from the residual energy and ζ_n^E represents is a random variable that varies from -1 to 1. This can be written as: $(\xi_n + \tilde{\xi}_n \zeta_n^E) + a_{n,m}\mathcal{E}_{n,m} \geq \beta_n E_{TH}, \forall n, m$. This represents infinite possible constraints due to the vale of ζ_n^E. One way to transform this is to consider worst case scenario, i.e, setting $\max\limits_{\zeta_n^E}\{\tilde{\xi}_n \zeta_n^E\}$.

In our proposed scheme, *n-th* IoT device can perform its computational task denoted by $\mathcal{T}_n$ locally or offload it to one of the UAV-MECs. Tasks from the set of tasks $\mathcal{T}_E$ ($\mathcal{T}_n \in \mathcal{T}_E$) can be assigned to a given UAV-MEC based on harvesting and offloading requirements and parameters associated with UAV-MECs [1]. The priority level of the task denoted by W_n requested by *n-th* IoT device is determined based on the residual energy. The digital twin can then reallocate UAV-MECs in real-time based on the priority of IoT devices. Different UAVs may have different CPU processing capabilities, meaning that one UAV may have low computation cost but high latency. When $\mathcal{D}_n \neq 0$ and $\mathcal{W}_n \neq 0$, the *n-th* IoT device can choose to perform computation task locally or offload it to one of the UAV-MECs. We define a binary decision variable for offloading as $a_{n,m}$, where $a_{n,m} = 1$ if *n-th* IoT device offloading to *m-th* UAV-MEC and $a_{n,m} = 0$ otherwise.

When $\mathcal{D}_n = 0$ and $\mathcal{W}_n = 0$, the *n-th* IoT device only needs resources for offloading and caching. This can be represented as $\alpha_n = \{0, 1\}$, where $\alpha_n = 1$ if *n-th* IoT device is caching only and $\alpha_n = 0$ otherwise. The UAVs may or may not be able to serve the IoT devices depending on the available resources.

This can be written as $b_n = \{0, 1\}$, where $b_n = 1$ if *n-th* IoT device is selected for operation and $b_n = 0$ otherwise. When the residual energy of *n-th* IoT device denoted by ξ_n is less than a certain threshold E_{TH}, then it needs to perform energy harvesting before offloading. The energy harvesting binary indicator can be represented as β_n, where $\beta_n = 1$ if $\xi_n < E_{TH}$ and $\beta_n = 0$ otherwise.

We formulate the optimization problem for task offloading that considers the latency, including digital twin synchronization latency and priority of IoT devices. Here, we formulate an optimization problem to minimize the utility as:

$$\begin{aligned} \min_{\mathbf{a},\mathbf{b}} : U = \sum_n \sum_m \omega [a_{n,m} (\mathcal{L}^O_{n,m} + (1-\alpha_n)\mathcal{L}^{UAV}_{nm}) \\ + \beta_n \mathcal{L}^H_{n,m} + L^s_n)] - (1-\omega)\sum_{n=1}^{N} W_n b_n, \end{aligned}$$

Subject to:

$$\begin{aligned} & C1 : \sum_m a_{n,m} = b_n, \forall n, m \\ & C2 : \sum_n a_{n,m} \leq Z_m, \forall m \\ & C3 : \sum_n a_{n,m} \mathcal{S}_n \leq \mathcal{A}_m, \forall m \\ & C4 : a_{n,m}(\mathcal{L}^O_{n,m} + (1-\alpha_n)\mathcal{L}^{UAV}_{nm}) + \\ & \quad \beta_n \mathcal{L}^H_{n,m} + L^s_n) \leq \mathcal{D}_n, \forall n, m \\ & C5 : (\xi_n + \tilde{\xi_n}\zeta^E_n) + a_{n,m}\mathcal{E}_{n,m} \geq \beta_n E_{TH}, \forall n, m, \\ & C6 : 0 \leq f^{UAV}_m \leq f^{UAV-MAX}_m, \forall m \\ & C7 : a_{n,m} = \{0, 1\}, \forall n, m \\ & C8 : b_n = \{0, 1\}, \forall n, \end{aligned} \tag{3}$$

where ω is the weight associated with objectives and L^s_n is the synchronization latency between the physical layer and digital twin layer. Constraint C1 ensures that *n-th* IoT device can only connect to one UAV. C2 limits the number of IoT devices that a UAV can serve to a maximum Z_m. C3 pertains to the caching capacity, i.e., the available cache capacity of *m-th* UAB denoted by $\mathcal{A}_m$ should be more than the offloading data by all the corresponding IoT devices. C4 ensures that the latency for the task of *n-th* IoT device composed of offloading, computation, harvesting, and digital twin synchronization latency must not exceed the computation deadline $\mathcal{D}_n$. C5 requires an IoT device's residual and harvested energy to be greater than the threshold E_{TH} to offload data. There is a possibility that the digital twin may not have updated information about residual energy. Therefore, C5 also includes the perturbing effect of residual energy, i.e., $\tilde{\xi_n}\zeta^E_n$. C6 restricts the maximum CPU frequency of UAV-MECs that can deviate in digital twin.

III. Solution Approach

The problem presented in (3) is a binary integer problem. We first use a branch-and-bound algorithm to obtain optimal results [12]. The algorithm divides the solution space into branches and bound the objective function in each branch. However, the worst-case complexity of this algorithm is exponential, making it unsuitable for large-scale IoT networks.

Algorithm 1 : Relaxed Heuristic Algorithm (RHA).

Inputs: N, $[x_n, y_n, 0]$, M, W_n, β_n, Z_m.
Output: UAVs coordinates $[x_m, y_m, h_m]$, $a_{n,m}$, b_n.
UAVs initial locations are random as centroids for clusters.
while UAVs locations converge **do**
  Assign each IoT device to the nearest UAV.
  Calculate mean of each cluster to update UAV location.
end while
Set the UAV height using optimized UAV coordinates.
Initialize the primal variables i and j, and set small positive value ϵ for termination criteria.
while Duality gap is greater than ϵ **do**
  To solve the linear system $\Lambda i = v$, we need to compute the matrix Λ and the vector v, where Λ is the matrix of constraints given in (3) and v is vector of constraints.
  Compute the matrix χ (KKT conditions) and r (vector of residuals) for the linear system $\chi_j = r$.
  Solve the linear system $\chi_j = r$ for j.
  Computer the search direction di and dj for the primal and dual variables using the KKT conditions.
  Compute the step size using a line search method.
  Update the primal variables i and dual variables j with the step size and search directions di and dj.
  Apply iterative rounding to obtain a feasible solution.
  Assign the nearest *m-th* UAV that can cover a given IoT device by setting binary variable $a_{n,m} = 1$.
end while

We then develop a relaxed heuristic algorithm (RHA) given in Algorithm 1 that utilizes K-mean clustering for UAVs deployment and interior point method for user association to solve the digital twin-assisted task offloading problem in (3). The algorithm considers the location of ground IoT devices for the placement of UAVs. The input parameters are the number of IoT devices N, their coordinates $[x_n, y_n, 0]$, and the number of available UAVs M. The K-mean unsupervised learning divides the IoT devices into M clusters. Initially, UAVs are placed randomly as centroids for the clusters. The algorithm then assigns IoT devices to the nearest UAV and calculates the mean for each cluster to update the UAV's location. This process continues until the UAV locations converge and there is no longer significantly change. Once the optimized locations of UAVs are obtained, the height of the UAVs is set based on the surrounding environment.

Next, we use the interior point method with iterative rounding to assign UAVs based on the objective and constraints in (3) [13]. To solve the optimization problem, we use the primal-dual interior point method which is not sensitive to polynomial time complexity [14]. We approximate the optimization problem by adding slack variables to a sequence of subproblems and obtain the primal-dual search direction using modified Karush-Kuhn-Tucker (KKT) conditions. We solve the KKT conditions using Newton's method [15] to a sequence of modified versions of KKT conditions [14]. Our algorithm solves linear programming problems iteratively, with gradually relaxed constraints, until an optimal solution is obtained. We use iterative rounding at each iteration to round the fractional solution to an integer solution that satisfies the constraints given in (3). Finally, we apply a threshold to the output of the interior point method; if it exceeds the threshold value, the *n-th* IoT device will establish a connection with *m-th* UAV by setting $a_{n,m} = 1$; otherwise, $a_{n,m} = 0$.

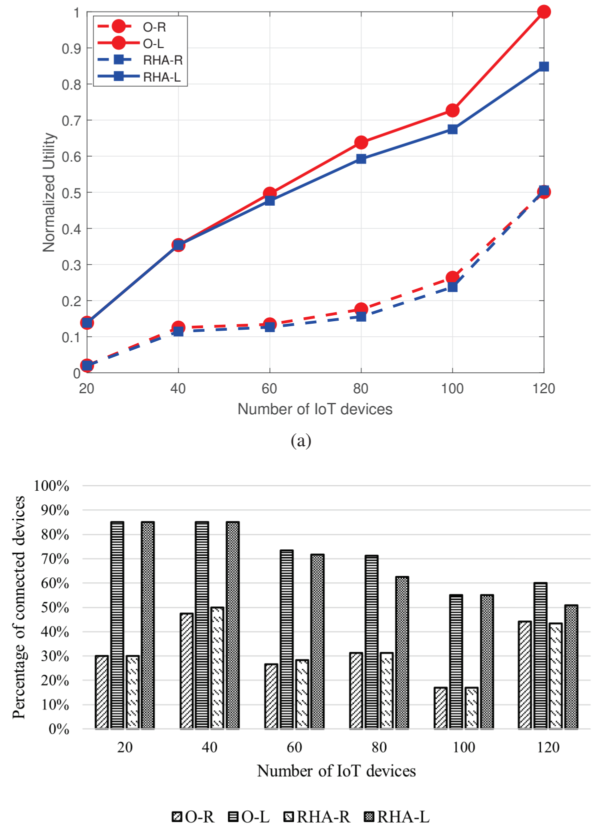
We have categorized the solution into four different types: (i) RHA-L, i.e., a relaxed heuristic algorithm with the K-mean learning algorithm for UAVs placement (ii) RHA-R, i.e., a relaxed heuristic algorithm that randomly places UAVs, (iii) O-L, i.e., the branch-and-bound algorithm with the K-mean learning algorithm for UAVs placement, and (iv) O-R, i.e., the branch-and-bound algorithm with UAVs placed randomly.

IV. Simulation Results

We perform simulations of digital twin-assisted task offloading in the UAV-MEC network with energy harvesting. The network consists of $M = 3-11$ UAVs and $N = 20-120$ IoT devices that are uniformly distributed in a 1000m $\times$ 1000m area. We prioritize IoT devices based on their residual energy by assigning them levels ranging from $W_n = 1-5$, with 5 being the highest priority and 1 being the lowest. We consider simulation parameters similar to [16]: size of data $\mathcal{S}_n$=10–20 Mbits, computation deadline $\mathcal{D}_n$=1–12 sec, and computation workload $\mathcal{W}_n$=452–737 cycles/bit.

Fig. 2 shows performance comparison for $M = 3$, $\tilde{\xi_n} = 40\%$ deviation from ξ_n, priority level $W_n = 5$, and $\Delta\mathcal{L}_n^{UAV} = 0.02$. Fig. 2(a) shows the normalized utility $\left(U_{NORM} = 1 - \frac{U-\min(U)}{\max(U)-\min(U)}\right)$ versus the total number of IoT devices. The normalized utility increases with the number of IoT devices for all algorithms. Algorithms with a learning approach have higher utility than random algorithms in all cases. The maximum normalized utility for O-L is higher than RHA-A, and the same trend is seen in algorithms with random UAV deployment. This is because the K-mean learning-based placement of UAVs can result in better system configuration. The proposed RHA-L has comparable utility to O-L with less complexity, emphasizing their scalability for large-scale IoT networks. Fig. 2(b) shows the percentage of connected IoT devices versus the total number of IoT devices. The number of associated IoT devices should generally increase with the increase in the total number of IoT devices and the number of UAVs. However, in this case, the number of UAVs is limited to 3, and each UAV can connect only a limited number of IoT devices Z_m. The percentage of connected devices in the cases of algorithms with learning placement of UAVs is higher than the random placement of UAVs, with RHA-L performing better than O-L. However, this increase in connected IoT devices comes at the cost of latency in the objective function.

Fig. 3 illustrates the performance comparison for $N = 120$, $\tilde{\xi_n} = 40\%$ deviation from ξ_n, priority $W_n = 5$, and $\Delta\mathcal{L}_n^{UAV} = 0.02$. Fig. 3(a) depicts the normalized utility versus the total number of UAVs. For all algorithms, the normalized utility



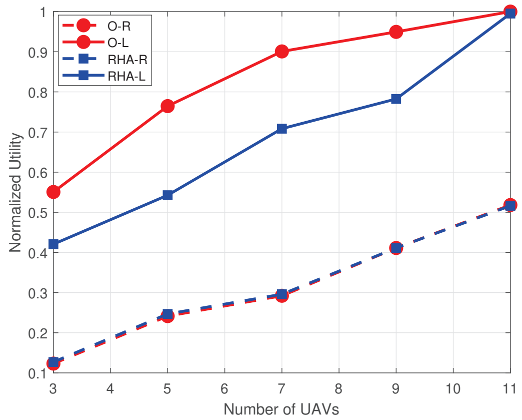
(a)

(b)

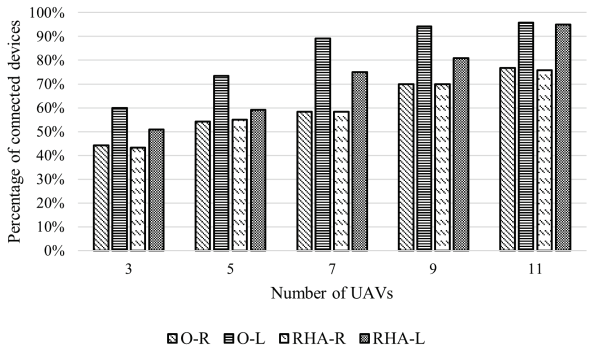
Fig. 2. Performance comparison for $M = 3$, $\tilde{\xi_n} = 40\%$ deviation from ξ_n, priority $W_n = 5$, and $\Delta\mathcal{L}_n^{UAV} = 0.02$ (a) utility versus IoT devices and (b) percentage of connected IoT devices versus total IoT devices.

increases with the number of IoT devices. Algorithms with a learning approach have a higher utility than random algorithms in all cases. The maximum normalized utility is observed with O-L compared to RHA-L, and a similar trend is seen in algorithms with random UAV deployment. This is because the K-mean learning-based placement of UAVs can result in a better system configuration. RHA-L has comparable utility to O-L with less complexity, highlighting their scalability for large-scale future IoT networks. Fig. 3(b) shows the percentage of connected IoT devices versus the total number of UAVs. The number of associated IoT devices generally increases with the total number of UAVs. In algorithms with learning placement of UAVs, the percentage of connected devices is higher than the random placement of UAVs, with RHA-L performing better than O-L.

Fig. 4 shows the number of connected IoT devices versus digital twin latency deviation for $N = 120$, $M = 11$, $\tilde{\xi_n} = 40\%$ deviation from ξ_n, and $W_n = 5$. When the number of IoT devices and UAVs remains constant, the number of connected devices decreases as the digital twin latency devi-



(a)



(b)

Fig. 3. Performance comparison for $N = 120$, $\tilde{\xi_n} = 40\%$ deviation from ξ_n, priority $W_n = 5$, and $\Delta\mathcal{L}_n^{UAV} = 0.02$ (a) utility versus number of UAVs and (b) percentage of connected IoT devices versus total UAVs.

ation increases. Conversely, for a given digital twin latency deviation, the number of connected IoT devices decreases as the task data $\mathcal{S}_n$ increases. In this case, the task data increases from $\mathcal{S}_n = 10 - 20$Mbits to $\mathcal{S}_n = 50 - 75$Mbits. This is because more data needs to be transmitted from IoT devices to UAV-MEC and computed on UAV-MEC as the computing tasks increase as a result, fewer IoT devices can be served.

V. Conclusion

In this paper, we presented an efficient offloading scheme in UAV-MEC IoT networks by leveraging digital twins and energy harvesting. We examine the impact of uncertainties that arise from the UAV-MEC environment and low-power IoT devices. Our proposed scheme utilizes a real-time simulated environment to optimize the placement of UAVs and the association of IoT devices to minimize latency and maximize the number of associated IoT devices while considering various constraints. To obtain optimal results, we have employed a branch and bound algorithm. Moreover, we have proposed a relaxed heuristic algorithm to reduce computational complexity while providing near-optimal solutions. Simulation results

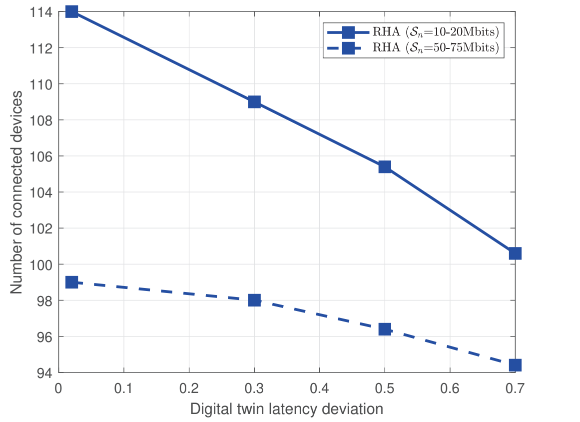


Fig. 4. Number of connected IoT devices versus digital twin latency deviation for $N = 120$, $M = 11$, $\tilde{\xi_n} = 40\%$ deviation from ξ_n, and $W_n = 5$.

are presented to evaluate the effectiveness of these algorithms and demonstrate the benefits of digital twin technology in UAV-MEC networks that incorporate energy harvesting.

References

[1] M. Basharat, M. Naeem, and A. Anpalagan, "Latency and cost minimization for task offloading with energy harvesting in UAV-MEC network," in *21th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt 2023)*, IEEE, 2023.

[2] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong, *et al.*, "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 22–27, Nov. 2022.

[3] D.-Y. Jeong, M.-S. Baek, T.-B. Lim, Y.-W. Kim, S.-H. Kim, Y.-T. Lee, W.-S. Jung, and I.-B. Lee, "Digital twin: Technology evolution stages and implementation layers with technology elements," *IEEE Access*, vol. 10, pp. 52609–52620, May 2022.

[4] B. Li, W. Xie, Y. Ye, L. Liu, and Z. Fei, "FlexEdge: Digital twin-enabled task offloading for UAV-aided vehicular edge computing," *IEEE Transactions on Vehicular Technology*, 2023.

[5] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, "Digital twin assisted task offloading for aerial edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10863–10877, Oct. 2022.

[6] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng, "Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1427–1444, Jan. 2022.

[7] T. Do-Duy, D. Van Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in mobile edge computing," *IEEE Wireless Communications Letters*, vol. 11, no. 4, pp. 806–810, Apr. 2022.

[8] L. Zhang, H. Wang, H. Xue, H. Zhang, Q. Liu, D. Niyato, and Z. Han, "Digital twin-assisted edge computation offloading in industrial Internet of things with NOMA," 2022.

[9] A. Khan, F. Shahid, C. Maple, A. Ahmad, and G. Jeon, "Toward smart manufacturing using spiral digital twin framework and twinchain," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1359–1366, Feb. 2022.

[10] A. Alzidaneen, A. Alsharoa, and M.-S. Alouini, "Resource and placement optimization for multiple UAVs using backhaul tethered balloons," *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 543–547, Apr. 2020.

[11] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *IEEE global communications conference (GLOBECOM)*, pp. 2898–2904, IEEE, 2014.

[12] N. Thakoor and J. Gao, "Branch-and-bound for model selection and its computational complexity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 5, pp. 655–668, May 2011.

[13] R. Masroor, M. Naeem, T. Akram, A. Anpalagan, and W. Ejaz, "Priority-based efficient deployment of UAVs with resource constraints in mission-critical networks," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2023.

[14] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[15] O. Mangasarian, "A Newton method for linear programming," *Journal of Optimization Theory and Applications*, vol. 121, pp. 1–18, 2004.

[16] A. Ndikumana, N. H. Tran, T. M. Ho, Z. Han, W. Saad, D. Niyato, and C. S. Hong, "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1359–1374, Jun. 2020.