Energy-Efficient UAV-Aided Computation Offloading on THz Band: A MADRL Solution

Yuanjian Li[1], A.S. Madhukumar[1],
Tan Zheng Hui Ernest[2], Gan Zheng[3], Walid Saad[4], and Hamid Aghvami[5]

1, College of Computing and Data Science
Nanyang Technological University, Singapore

2, Agency for Science, Technology and Research, Singapore

3, the University of Warwick, the UK

4, Virginia Tech, the US

5, King's College London, the UK

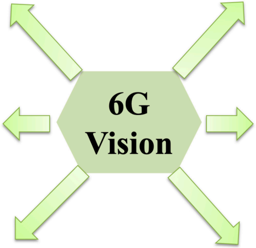
NANYANG TECHNOLOGICAL UNIVERSITY

The Next-Gen (6G) Wireless Systems

Global coverage
- Satellite and UAV communications
- Terrestrial communications
- Maritime communications
- Underwater, underground communications

Full applications
- Integration of communications, computing, storage, control, sensing, localization, robotics, AI, and big data
- Terminal-network-cloud
- Cloud/fog/edge computing

All digital
- Digital twins: mapping between physical world and virtual world
- Intelligent connection of "human-machine-things-environment"



All spectra
- Sub-6 GHz (including short wave and acoustic wave)
- cmWave + mmWave + THz
- Optical wireless

All senses
- Holographic communications/storage
- Truly immersive XR: fusion of virtuality and reality
- Tactile Internet

Strong security
- Physical layer security and network layer security
- Reliable communications
- Intelligent endogenous security

C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas et al., "On the road to 6G: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

Motivations

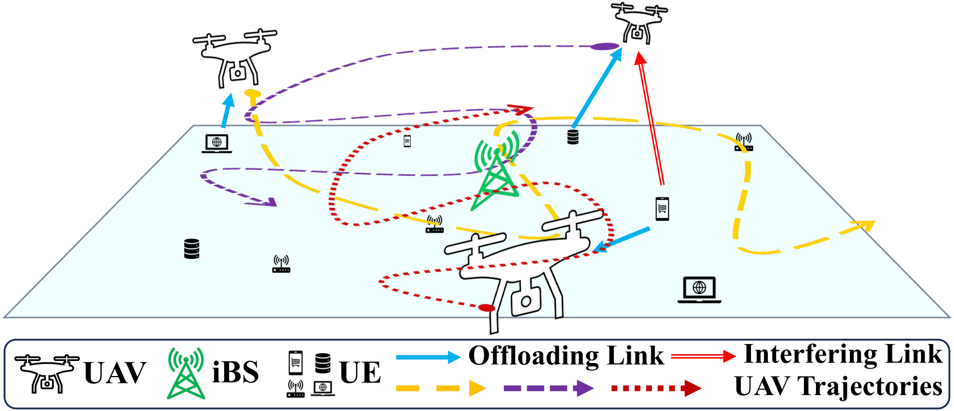
- IoT user equipments (UEs) have **limited power and computing resources**, yet require prolonged operation, emphasizing **energy efficiency**.
- **THz technology** enables **low-latency** and **high data rate** multi-access edge computing (MEC) services, such as task offloading.
- **UAV-aided MEC on THz** mitigates propagation limits, blockages, and coverage issues by leveraging UAVs' **mobility** and short-range **LoS links**.

Contributions

- **The Gap:** Limited research exists on energy-efficient UAV-aided MEC systems operating on the THz band.
- **The Difficulty:** AI-native solutions for adapting to dynamic wireless environments remain inherently challenging and lacking.
- **Core Contribution:** This work addresses the gap by proposing an AI-native algorithm for energy efficiency maximization in UAV-aided MEC networks over the THz band.

System Model Diagram of Multi-UAV Computation Offloading for Many IoT UEs

Multiple UAVs are deployed to provide multiple energy-limited computation-scarce terrestrial IoT user equipments (UEs) with accessible task offloading services on THz band.



Key Considerations

To enable energy-efficient multi-UAV-assisted MEC frameworks in IoT, the following challenges must be addressed:

- How to design UAV trajectories to establish high-quality ground-to-air (G2A) links for efficient task offloading in multi-UE scenarios?
- How to jointly optimize communication and computation resources, including transmit power, UAV-UE associations, CPU clock speeds, and time slicing, to enhance system metrics like energy efficiency?
- How to develop an agile multi-agent learning framework capable of handling non-stationarity and dynamically adapting to the challenges of MUME UAV-assisted MEC systems?

The Energy Efficiency Maximization Problem under Investigation

We seek to maximize expected energy efficiency for multi-UAV multi-UE computation offloading systems over the THz band.

$$\max_{\{\mathbf{v}_u[n], f_g[n], \psi_g^u[n], \tau_g^u[n], P_g[n]\}} \frac{1}{N}\sum_{n=1}^{N} \frac{d[n]}{E[n]},$$

$$\text{s.t.} \quad (1), (2), (3), (4), (5), (6), (8), (9)$$

$$d[n] = \sum_{g\in\mathcal{G}} d_g[n]$$

$$E[n] = \sum_{g\in\mathcal{G}} E_g^l[n] + \sum_{u\in\mathcal{U}} \left(E_u[n] + \Re E_u^p[n]\right)$$

We **jointly** optimize communication and computation resources, including:

- ⇒ **UAVs' trajectories** ($\mathbf{v}_u[n]$),
- ⇒ UEs' local central processing unit (**CPU**) clock speeds ($f_g[n]$),
- ⇒ **UAV-UE associations** ($\psi_g^u[n]$),
- ⇒ **time slot slicing factor** ($\tau_g^u[n]$),
- ⇒ **UEs' offloading powers** ($P_g[n]$).

Why AI-Aided Solution?

Solving the formulated multi-dimensional maximization problem with classical optimization techniques, such as game theory or convex optimization, is extremely **difficult** due to the following key challenges:

- **Non-convex objective function:** The objective consists of accumulated fractional functions with multiple summations, making it a on-convex mixed-integer non-linear programming (MINLP) problem that is **NP-hard**.
- **Coupled optimization parameters:** The parameters, including discrete binary variables $\psi_g^u[n]$, vector $\mathbf{v}[n]$, and ranged floats $f_g[n]$, $\tau_g^u[n]$, and $P_g[n]$, are intertwined in both the objective and constraints.

Why AI-Aided Solution?

- **Non-convex constraints:** Examples include:
  - Norm inequality in the mobility constraint,
  - Binary index-involved UAV-UE association constraint,
  - Time slot allocation constraints.

These lead to high computational and algorithmic overheads.

Alternative Solution: An AI-native solution from a data driven perspective, i.e., model-free DRL-aided algorithm, will be proposed to efficiently tackle the formulated optimization problem by training with raw experiences from interactions between DRL agents and the task offloading environment.

Why Multi-Agent Reinforcement Learning?

- Enables distributed implementation of wireless protocols at the edge.
- Facilitates experience sharing, allowing less-trained agents to learn from more skilled ones.
- Accommodates heterogeneous agents with diverse learning goals and device capabilities.

Main Difficulties in Solving the Joint Computation and Communication Resource Management Problem

- **High-dimensional spaces:** Continuous state and action spaces cause severe dimensionality challenges.
- **Exploration vs. Exploitation:** Balancing new strategy exploration with exploiting learned policies is complex in infinite action-state spaces.
- **Non-stationarity:** Dynamic environments cause state transitions and rewards to depend on joint actions, with agent behaviors evolving over time.
- **Scalability:** Complexity grows exponentially as the number of agents, e.g., thousands of devices, increases.
- **Sample inefficiency:** Multi-agent learning requires a large number of samples to establish effective policies or equilibria.

The Proposed MADRL Algorithm

Algorithm 1: The Proposed DSPAC-MN Solution

Initialization: Initialize online NNs' layers, as per OWI. Synchronize the exploratory actor networks and shadow networks via $\mathbf{\Theta}^{p_u} \leftarrow \mathbf{\Theta}^{a_u}$, $\mathbf{\Theta}^{a_u^-} \leftarrow \mathbf{\Theta}^{a_u}$ and $\mathbf{\Theta}^{c_j^-} \leftarrow \mathbf{\Theta}^{c_j}$. Initialize replay buffer $\mathcal{B}$ of size $\mathbf{B}$ and the mini-batch simpler $\mathcal{D}$ of size $\mathbf{D}$. Set total training step $n_t = 0$;
for $te = [1, te_{max}]$ **do**
  Reset time step $n = 0$, UAVs' locations to $\mathbf{q}_u[n]$ and queues to $\mathcal{Q}_g[n] = 0$, then the current state $s[n] = \{\mathbf{q}_u[n], \mathcal{Q}_g[n]\}$ is generated;
  repeat
    Perturb each exploratory actor via $\mathbf{\Theta}^{p_u} \leftarrow \mathbf{\Theta}^{a_u} + \mathbf{\Theta}^{p}$;
    Each UAV observes $s[n]$ and outputs $a_u[n] = \pi_u(s[n]|\mathbf{\Theta}^{p_u}) + \mathbf{N}$, then the joint action $\mathbf{a}[n] = [a_u[n]]_{u\in\mathcal{U}}$ is formulated;
    Execute the joint action $\mathbf{a}[n]$, observe the next state $s[n+1]$ and receive the immediate common reward $r[n]$;
    if $|\mathcal{B}| < \mathbf{B}$ **then**
      Archive experience $\langle s[n], \mathbf{a}[n], s[n+1], r[n]\rangle$ into $\mathcal{B}$;
    else
      Replace the earliest stored experiences in $\mathcal{B}$ with the new transition $\langle s[n], \mathbf{a}[n], s[n+1], r[n]\rangle$;
    if $|\mathcal{B}| \geq \mathbf{D}$ **then**
      Randomly sample a mini-batch of size $\mathbf{D}$ from $\mathcal{B}$ into $\mathcal{D}$, i.e., $(s_m, \mathbf{a}_m, s_{m+1}, r_m) \in \mathcal{D} \sim \mathcal{B}$;
      for $u \in \mathcal{U}$ **do**
        The shadow actor outputs $\pi_u\left(s_{m+1}|\mathbf{\Theta}^{a_u^-}\right) + \mathbf{N}^-$ to calculate the target Q value;
      Update the dual online critics' trainable parameters $\mathbf{\Theta}^{c_j}$ by batch gradient descent on MSE loss $\ell(\mathbf{\Theta}^{c_j})$ in (17);
      Increment the total training step $n_t \leftarrow n_t + 1$;
      if $n_t \% N_s == 0$ **then**
        for $u \in \mathcal{U}$ **do**
          The online actor generates $\pi_u(s_m|\mathbf{\Theta}^{a_u})$;
        Update all the online actors' tunable parameters $\mathbf{\Theta}^{a_u}$ by batch gradient ascent by the chain rule as per (19);
        Update shadow networks $\mathbf{\Theta}^{a_u^-} \leftarrow \tau_s \mathbf{\Theta}^{a_u} + (1-\tau_s)\mathbf{\Theta}^{a_u^-}$ and $\mathbf{\Theta}^{c_j^-} \leftarrow \tau_s \mathbf{\Theta}^{c_j} + (1-\tau_s)\mathbf{\Theta}^{c_j^-}$;
    Trigger time step incrementation $n \leftarrow n + 1$;
  until $\|\mathbf{q}_u - \mathbf{q}_{u'\in\{\mathcal{U}\setminus u\}}\| < D, \exists u, \exists u' \,||\, \mathbf{q}_u(n) \notin \flat, \exists u \,||\, n = N_{\max}$;

Update the trainable parameters of the **critics**:

$$\mathbf{\Theta}^{c_j} \leftarrow \mathbf{\Theta}^{c_j} - \alpha_c \nabla_{\mathbf{\Theta}^{c_j}} \ell\left(\mathbf{\Theta}^{c_j}\right),$$

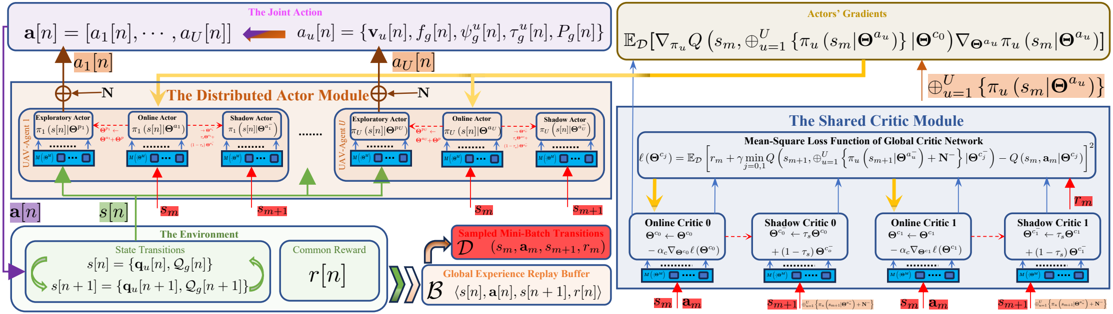
$$\ell\left(\mathbf{\Theta}^{c_j}\right) = \mathop{\mathbb{E}}_{(s_m, \mathbf{a}_m, s_{m+1}, r_m)\in\mathcal{D}\sim\mathcal{B}} \left[\mathbf{y}_m - Q\left(s_m, \mathbf{a}_m | \mathbf{\Theta}^{c_j}\right)\right]^2,$$

$$\mathbf{y}_m = r_m + \gamma \min_{j=0,1} Q\left(s_{m+1}, \oplus_{u=1}^{U}\left\{\pi_u\left(s_{m+1}|\mathbf{\Theta}^{a_u^-}\right) + \mathbf{N}^-\right\} | \mathbf{\Theta}^{c_j^-}\right)$$

Update the trainable parameters of the **actors**:

$$\mathbf{\Theta}^{a_u} \leftarrow \mathbf{\Theta}^{a_u} + \alpha_a \mathop{\mathbb{E}}_{s_m\in\mathcal{D}} \left[\nabla_{\pi_u} Q\left(s_m, \oplus_{u=1}^{U}\left\{\pi_u\left(s_m|\mathbf{\Theta}^{a_u}\right)\right\} | \mathbf{\Theta}^{c_0}\right) \times \nabla_{\mathbf{\Theta}^{a_u}} \pi_u\left(s_m|\mathbf{\Theta}^{a_u}\right)\right]$$

Workflow of the Proposed DSPAC-MN Algorithm



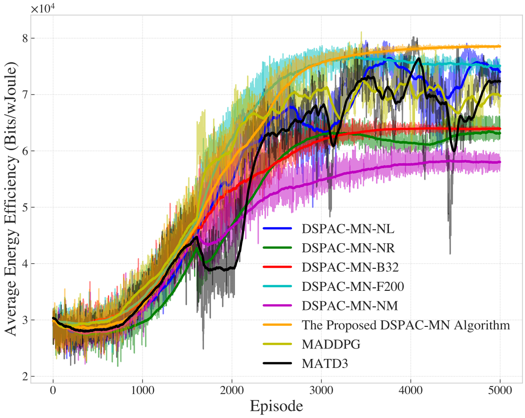
- *Distributed Agents*: explore in parallel
- *Shared Critic*: cooperative learning
- *Modularized Inputs*: balanced dimension
- *Perturbed Actors*: enhanced exploration

Setups for System Parameters and Hyperparameters of the Learning Process

Table I: Setups for System Parameters and Hyperparameters of the Learning Process

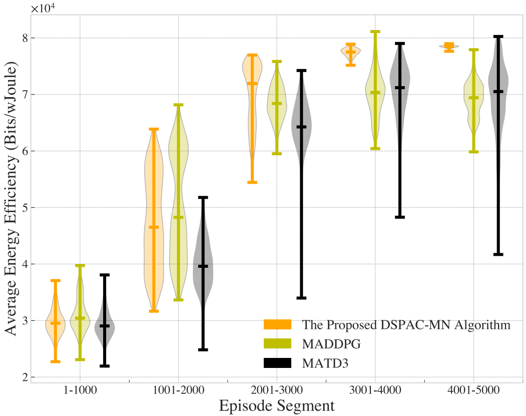
Parameters	**Values**	**Parameters**	**Values**	**Parameters**	**Values**
Number of terrestrial UEs $\|\mathcal{G}\| = G$	6	Number of UAVs $\|\mathcal{U}\| = U$	2	Replay buffer $\mathcal{B}$'s capacity $\mathbf{B}$	10^5
Length of time slot δ_t	0.5 s	Queue capacity A_g^*	5×10^7 bits	Mini-batch sampler $\mathcal{D}$'s size $\mathbf{D}$	256
Safety distance for avoiding collision D	8 m	UEs' computation intensity c_g	10^3 cycles/bit	Exploration noise $\mathbf{N}$	Normal $(0, 5)$
UEs' maximum CPU-cycle frequency f_g^*	0.5 GHz	UEs' maximum transmit power P_g^*	30 dBm	Exploration noise variance decaying rate	0.999/episode
UEs' transmission bandwidth B	20 GHz	AWGN variance σ^2	-90 dBm	Staggered policy renewal frequency N_s	2
Computation overhead c_o	2	UAVs' computation intensity c_u	10^3 cycles/bit	Shadow policy tempering noise $\mathbf{N}^-$	Normal $(0, 1)$
UAVs' CPU-cycle frequency budget f_u^*	10 GHz	New task instances' variance A_g	2×10^6 bits	Outbound/collision penalty p_o/p_c	100; 100
Minimum/maximum UAV speed $\mathbf{v}^-/\mathbf{v}^*$	10 m/s; 50 m/s	UAV altitude z_u	200 m	Dropout rate for online actors/critics	0.2
UEs' effective capacitance coefficient γ_g	10^{-28}	UEs' non-CPU power cost E_g^*	0 Joule	Learning rates α_a/α_c	10^{-4}; 10^{-3}
UAVs' effective capacitance coefficient γ_u	10^{-28}	UAVs' non-CPU power cost E_u^*	0 Joule	Critic's/actors' Exponential learning rate scheduler factor	0.9999; 0.9999
Fuselage drag ratio ϱ_0/Rotor solidity ϱ_2	0.6; 0.05	Air density ϱ_1	1.225 kg/m^3	Discount factor γ; Parameter-wise noise variance σ_p^2	0.99; 0.1
Rotor disc area ϱ_3	0.503 m^2	Blade angular velocity ϱ_4	300 radians/s	Polyak averaging coefficient τ_s	10^{-6}
Rotor radius ϱ_5	0.4 m	Profile drag Coefficient ϱ_6	0.012	Maximum training episodes $te_{\max}$	5000
Incremental correction factor to induced power ϱ_7	0.1	UAV weight ϱ_8	20 Newton	Step threshold $N_{\max}$	50
Average rotor induced velocity v_0	4.03 m/s	Rotor blade tip speed v_{tip}	120 m/s	Direction-aware collision penalty triggering factor T_c	1
Relative pressure	1013.25 hPa	Carrier frequency of THz channel f_c	0.3 THz	Dimension of each module's output $\|M_i\|$	10
Speed of light C	3×10^8 m/s	Antenna gains G_t/G_r	20 dBi; 0 dBi	Number of modules inside each actor	$3\,(U + 1)$
Relative humidity; Energy regulation factor $\mathfrak{R}$	0.5; 0.02	Relative temperature	296.15 $^\circ$K	Number of modules inside the shared cirtic	$4\,(U + 1 + 2U)$

Average Energy Efficiency versus Training Episode



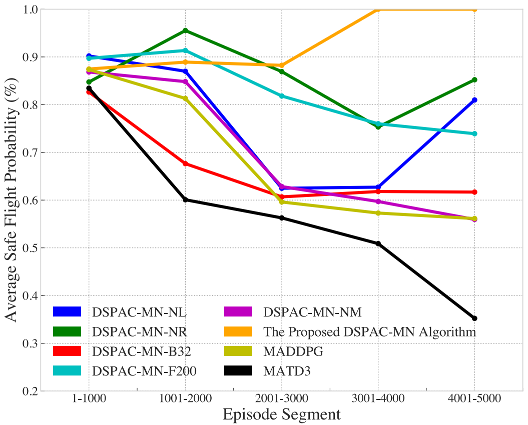
- MADDPG: an extension of deep deterministic policy gradient (DDPG) to handle multi-agent scenarios
- MATD3: an extension of twin-delayed DDPG (TD3) to reduce overestimation bias of MADDPG
- DSPAC-MN-NM: DSPAC-MN without modular networks
- DSPAC-MN-NR: Regularization-less DSPAC-MN
- DSPAC-MN-NL: DSPAC-MN without learning rate scheduling
- DSPAC-MN-B32: DSPAC-MN with batch size of 32
- DSPAC-MN-F200: DSPAC-MN with policy renewal frequency of 200

Violin Plot versus Episode Segments



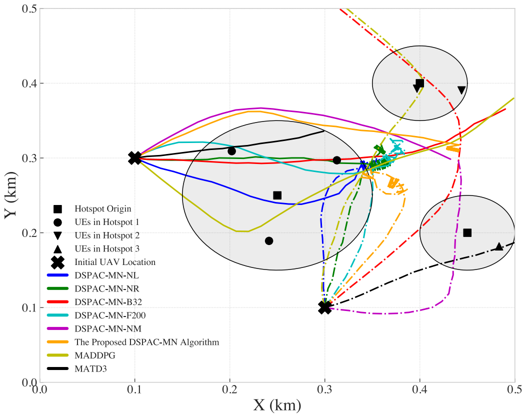
- The metric is measured in **bits/wJoule**, where wJoule accounts for a weighted sum energy consideration, ensuring magnitude **fairness** between computation/offloading and propulsion energy costs.
- The proposed DSPAC-MN **significantly outperforms** other baselines, demonstrating its **effectiveness** and **efficiency** through tailored components such as perturbed actors, a shared critic, and modularized inputs.

Average Safe Flight Probability



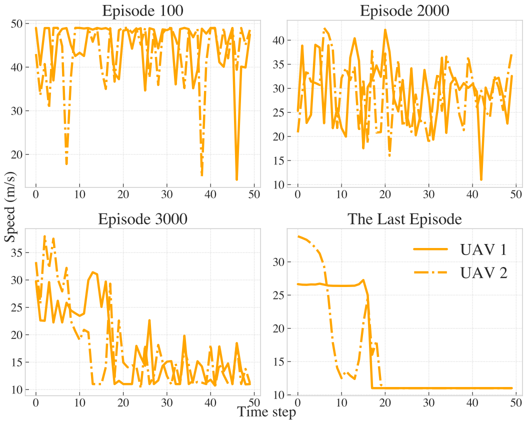
- A safe flight criterion is violated if any UAV flies **out of bounds** or if any pair of UAVs **collides**.
- The proposed DSPAC-MN approach is the only method achieving **100% safe flight navigation**, while other baselines have a **higher likelihood** of violating the rules imposed by the optimization problem.

Visualization of and Comparison on Devised Trajectories over Various Algorithms



- The proposed DSPAC-MN solution generates trajectories that are **well-separated** and **clear of borders**.
- Baselines such as DSPAC-MN-B32, MADDPG, and MATD3 fail to prevent UAVs from **crashing into borders**, **violating mobility constraints**.
- Benchmarks like DSPAC-MN-NL, DSPAC-MN-NR, DSPAC-MN-F200, and DSPAC-MN-NM produce trajectories that result in **collisions**, **breaching collision constraints**.

Safe Flight Probability and Designed Propulsion Speed



Comparison on Safe Flight Probability

Algorithms	MADDPG	MATD3	DSPAC-MN-NM	DSPAC-MN-NR
The last 1000 episodes	0.5729	0.59992	0.73998	0.62
The last 200 episodes	0.5853	0.6015	0.74	0.62
The last 10 episodes	0.62	0.598	0.74	0.62

Algorithms	DSPAC-MN-NL	DSPAC-MN-B32	DSPAC-MN-F200	**DSPAC-MN**
The last 1000 episodes	0.73734	0.77556	0.90374	**1.0**
The last 200 episodes	0.8205	0.78	0.9905	**1.0**
The last 10 episodes	0.848	0.78	**1.0**	**1.0**

The End

Thanks for your attentions

Dr. Yuanjian Li, Research Fellow at Nanyang Technological University (NTU), Singapore Email: yuanjian.li@ntu.edu.sg