Trustworthy Aerial Edge Computing with Robust, Priority-Aware, and Learning-Based Optimization for 6G Networks

by

Muhammad Omair Butt

A dissertation

presented to Lakehead University

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the program of

Electrical and Computer Engineering

Lakehead University

August 2026

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A DISSERTATION

I hereby declare that I am the sole author of this dissertation. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners.

I authorize Lakehead University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Lakehead University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Examining Committee Membership

The following served on the Examining Committee for this thesis.

Supervisor:	Dr. Waleed Ejaz LU Research Chair & Associate Professor Department of Electrical and Computer Engineering, Lakehead University Barrie STEM Hub
External Examiner:	Dr. Gagangeet Aujla Associate Professor Department of Computer Science Durham University, Durham, United Kingdom
Internal Examiner:	Dr. Salama Ikki Professor Department of Electrical and Computer Engineering Lakehead University, Thunder Bay
Internal Examiner:	Dr. Abdulsalam Yassine Associate Professor Department of Software Engineering Lakehead University, Barrie STEM Hub

Abstract

Aerial edge computing serves remote and disaster-affected regions where terrestrial coverage is sparse or unavailable. Trust in these deployments reflects the consistency with which an uncrewed aerial vehicle (UAV) fulfills the tasks it accepts. The existing literature generally develops trust measures without explicitly connecting them to the admission of Internet of things (IoT) devices. These measures typically associate trust with regions, data contributions, or device populations. IoT device admission meanwhile focuses on UAV resources such as coverage, residual energy, and computing capacity. This resource-oriented view describes what a UAV can offer rather than what it has reliably delivered. Offloading commits each device to the service the selected UAV provides, which makes trust an important consideration in aerial edge computing.

This thesis proposes a trust-aware aerial edge computing framework that delivers trustworthy mobile edge computing (MEC) from UAVs to IoT devices. The framework evaluates each UAV through multiple trust dimensions and admits an IoT device only when the selected UAV meets the trust requirement of its task. We formulate joint UAV deployment, device association, and resource allocation as a multi-criteria mixed-integer nonlinear program (MINLP) balancing connectivity, trust, cost, and task latency. A penalty-guided optimization (PGO) algorithm based on sequential quadratic programming (SQP) solves the resulting problem and achieves near-optimal connectivity relative to branch-and-bound at substantially lower computational cost.

We extend the framework to uncertain IoT device states, differentiated task requirements, and device mobility. Uncertainty in residual energy and device location enters the IoT-UAV association decision as an explicit input. Task priority accompanies trust within the same association decision and captures the deadline-sensitive requirements of individual tasks. IoT mobility follows a Gauss-Markov process within the extended problem formulation. UAV deployment and the associated decision-making proceed under centralized, hybrid, and fully distributed control schemes. Connectivity remains near the optimal benchmark under both device-state uncertainty and task heterogeneity. Mobile settings reveal a trade-off between accumulated reward and trust assurance across the three control strategies. Trust therefore functions as a decision criterion that complements resource availability, task requirements, and network dynamics in aerial edge computing.

Acknowledgements

I thank Dr. Waleed Ejaz for his supervision and support throughout this research.

I am grateful to my committee members Dr. Salama Ikki and Dr. Abdulsalam Yassine for the time they spent reading this work and for feedback that strengthened the thesis.

I also thank my colleagues in the Wireless Communication and Networks (WCN) Research Group for their suggestions, encouragement, and company over the course of this work.

To my family: your patience and support carried me through every stage of this work, and I could not have finished it without you.

This research was supported by Lakehead University through the Faculty of Engineering and the Faculty of Graduate Studies, and by the Natural Sciences and Engineering Research Council of Canada (NSERC) under a Discovery Grant.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Acronyms	Description
6G	Sixth-generation
UAVs	Uncrewed aerial vehicles
HAPs	High-altitude platforms
IoT	Internet of things
MEC	Mobile edge computing
PGO	Penalty-guided optimization
MINLP	Mixed-integer nonlinear program
BMK	Benchmark
BRE	Baseline relaxation
GDY	Greedy
ID	Intelligent deployment (Chapter 4); Intelligent association decision (Chapter 5)
RD	Random deployment (Chapter 4); Random association decision (Chapter 5)
TWK	Trust-weighted K-means deployment
RND	Random deployment (Chapter 6)
FXD	Fixed deployment
DQN	Deep Q-network
CTDE	Centralized training with decentralized execution
FDS	Fully distributed scheme
RL	Reinforcement learning
MADDPG	Multi-agent deep deterministic policy gradient
MDP	Markov decision process
SQP	Sequential quadratic programming
LP	Linear program
ADMM	Alternating direction method of multipliers
DBSCAN	Density-based spatial clustering of applications with noise
OFDMA	Orthogonal frequency division multiple access

LoS	Line-of-sight
NLoS	Non-line-of-sight
QoS	Quality of service
CPU	Central processing unit
ReLU	Rectified linear unit

List of Symbols

Symbol	Description
Indices and sets	
n	IoT device index
m	UAV index
k	HAP index (Chapters 3–5); time slot (Chapter 6)
$\mathcal{N}$	Set of IoT devices
$\mathcal{M}$	Set of UAVs
$\mathcal{K}$	Set of HAPs (Chapters 3–4)
K_u	Intra-cluster tracking interval (Chapter 6)
K_g	Inter-cluster repositioning interval (Chapter 6)
Positions and channel	
O_n	IoT device coordinates
O_m	UAV coordinates
O_k	HAP coordinates
x_n, y_n	IoT device horizontal coordinates
x_m, y_m, z_m	UAV horizontal coordinates and altitude
x_k, y_k, z_k	HAP horizontal coordinates and altitude
d_{nm}	IoT-UAV distance
d_{nm}^{W}	Worst-case IoT-UAV distance under spatial uncertainty
d_{mk}	UAV-HAP distance
d_m^{MAX}	Maximum UAV coverage radius
c	Speed of light
f_c	Carrier frequency
B_{nm}	Allocated bandwidth
γ_{nm}	Signal-to-noise ratio
$g_{nm}(k)$	Channel gain (Chapter 6)
p_n	IoT device transmit power
σ^2	Noise power
R_{nm}	IoT-UAV data rate

Symbol	Description
R_{mk}	UAV-HAP data rate
$\eta_{\text{LoS}}, \eta_{\text{NLoS}}$	Excessive path loss for LoS and NLoS links
θ	Minimum elevation angle
$\varepsilon_x, \varepsilon_y$	Spatial uncertainty bounds
Δ_n^{PER}	IoT device spatial perturbation
$v_n(k), \theta_n(k)$	IoT device velocity and direction (Chapter 6)
Tasks and computation	
F_{nm}	Task from IoT device n to UAV m
D_{nm}	Task data size
δ_{nm}	Computational workload
A_{nm}	Computational demand
A_m^{MAX}	Maximum UAV computational capacity
A_m^{RES}	Residual UAV computational capacity
S_{nm}	Computation result size
f_n, f_m	Processing frequency of IoT device and UAV
$a_n(k)$	Task arrival indicator (Chapter 6)
p^{GEN}	Task generation probability (Chapter 6)
$D_n(k), \delta_n(k)$	Task data size and workload (Chapter 6)
$f_{nm}(k)$	CPU frequency allocated to device n (Chapter 6)
F_m^{MAX}	Maximum UAV CPU frequency (Chapter 6)
Latency	
L_n^{LOC}	Local computation latency
L_{nm}^{EDGE}	Edge computation latency
L_{nm}^{TASK}	Total task latency
L_{mk}^{CC}	UAV-HAP coordination latency
L_n^{MAX}	Maximum tolerable latency
D_{mk}	Coordination overhead
S_{mk}^{CC}	Control data size
η_{mk}	Coordination decision
Energy	

Symbol	Description
E_{nm}	Energy required for task execution
E_m^{MAX}	Maximum UAV energy capacity
E_m^{RES}	Residual UAV energy
E_n^{NOM}	Nominal IoT residual energy
E_n^{PER}	IoT energy perturbation magnitude
E_n^{THR}	Minimum IoT energy for offloading
E_{nm}^{DEM}	Energy demanded to serve a task
E_{nm}^{HAR}	Energy harvested by the IoT device
χ, ζ_n	Bounded perturbation random variables
κ_m	Effective switched capacitance (Chapter 6)
Trust	
T_{nm}	Trust of UAV m for IoT device n
T_n^{THR}	IoT device trust threshold
T_{nm}^{REP}	Reputation trust
T_{nm}^{SUC}	Task completion trust
T_{nm}^{AVA}	Computational resource availability trust
T_{nm}^{STA}	Operational stability trust
T_{nm}^{COM}	Communication link trust
T_{nm}^{ENV}	Operational environment trust
T_{nm}^{ENE}	Energy sufficiency trust
T_{nm}^{SEC}	Security assurance trust
T_{nm}^{SD}	Threat detection trust
T_{nm}^{SP}	Undetected threat penalty
T_{nm}^{OPR}	Operational trust (Chapter 4)
T_{nm}^{TAS}	Task reliability trust (Chapters 4, 6)
T_{nm}^{COV}	Coverage trust (Chapter 5)
T_{nm}^{CMP}	Computational trust (Chapter 5)
T_{nm}^{SAT}	Satisfaction trust (Chapter 5)
T_{nm}^{SFR}	Service freshness trust (Chapter 6)
F_{nm}^{SUC}	Successfully completed tasks

Symbol	Description
λ_{nm}	Packet loss rate
ϱ_{nm}	Environmental difficulty
$\phi_{nm}^{\mathrm{DET}}, \phi_{nm}^{\mathrm{UNDET}}, \phi_{nm}^{\mathrm{TOT}}$	Detected, undetected, and total threats
r_m	Threat detection probability
ω_{nm}	Severity of undetected threats
$B_m^{\mathrm{BRH}}, B_m^{\mathrm{INT}}$	Security breaches and monitored interactions
$A_{nm}(k)$	Age of information (Chapter 6)
w_1, w_2, w_3	Trust metric weights
ε	Numerical stability constant
Priority (Chapter 5)	
ψ_{nm}	Task priority score
ψ_{nm}^{DLN}	Deadline urgency component
ψ_{nm}^{SIZ}	Data volume component
ψ_{nm}^{CRT}	Operational criticality component
κ_n	Task criticality score
κ^{MAX}	Maximum criticality level
Cost	
C_m^{DEP}	UAV deployment cost
C_m^{HW}	UAV hardware cost
C_m^{OP}	UAV operational cost
C_{nm}^{TR}	Trust-related cost
c_m^{COM}	UAV communication cost coefficient
C_{nm}^{HAR}	Harvesting cost coefficient
C_m^{FIX}	Capital and maintenance expense
t_m	UAV operational time
k_s	Steepness of cost increase
Optimization	
α_{nm}	IoT-UAV association variable
β_n	IoT device selection variable
ϑ	Maximum IoT devices per UAV

Symbol	Description
ν	Multi-criteria utility weights
$\mathcal{U}_{nm}, \mathcal{H}_{nm}$	Multi-criteria utility
$\mathbf{U}$	Utility matrix
$\mathbb{M}$	Big-M constant
$\mathcal{L}$	Penalty-augmented objective
Γ	Integrality gap penalty term
γ	Penalty coefficient (Chapters 3–5); discount factor (Chapter 6)
X	Relaxed decision vector
H_i	Hessian approximation
S_i	SQP search direction
μ_i	Line search step size
ϵ^{THR}	Convergence threshold
I	Number of SQP iterations
Learning and control (Chapter 6)	
$\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}_t$	MDP state, action, reward, and transition
$r_n(k)$	Per-device reward
$\mathcal{J}$	Expected return
$\mathcal{D}$	Experience replay buffer
$\boldsymbol{w}_m^{\mathrm{ACT}}$	Actor network weights
$\boldsymbol{w}_i^{\mathrm{CRT}}$	Critic network weights
τ^{UPD}	Soft update factor
$\epsilon^{\mathrm{INIT}}, \epsilon^{\mathrm{DEC}}, \epsilon^{\mathrm{MIN}}$	Exploration rate, decay, and floor
$\sigma^{\mathrm{INIT}}, \sigma^{\mathrm{DEC}}, \sigma^{\mathrm{MIN}}$	Exploration noise scale, decay, and floor
ω_n	Trajectory smoothness coefficient
Γ^{PNL}	Trust violation penalty coefficient
$c(n)$	Cluster assignment
$\mathcal{C}_m(k)$	UAV coverage set
$\mathcal{C}^{\mathrm{MAX}}$	Maximum devices observable by a UAV actor per slot
$\Delta_m(k)$	UAV trust mismatch indicator

Symbol	Description
$r_{nm}(k)$	Per-association reward
$i_n(k)$	Device action index
B_s	Minibatch size
ϵ^{CON}	Consensus step weight
$\mathcal{F}_n(k)$	Task profile

Chapter 1

Introduction

Sixth-generation (6G) networks aim to provide connectivity and computing support across urban, rural, industrial, and disaster-affected environments [1, 2]. Mobile edge computing (MEC) delivers this support through servers placed near the Internet of things (IoT) devices that generate data [3]. Tasks exceeding the processing and energy budgets of individual devices depend on this nearby capacity to complete within their latency requirements. Terrestrial infrastructure supports this model where population density justifies its deployment. In contrast, the rural monitoring sites and remote industrial sites require service in seasonal or event-driven bursts. The terrestrial infrastructure serving disaster-affected areas risks failure exactly when coordinated response requires continuous computing support. IoT devices in these regions therefore require computing resources that operate independently of terrestrial infrastructure. These resources must also meet the service expectations the IoT devices' tasks impose.

Aerial networks establish the computing layer above the terrestrial domain through platforms that deploy on demand to the areas requiring service. Uncrewed aerial vehicles (UAVs) constitute the lower tier and carry edge servers that operate close to the devices generating the workload [4]. High-altitude platforms (HAPs) [5] occupy the stratospheric tier, where they coordinate resources among the UAVs below and extend coverage across wider regions.

Combined operation of the two tiers provides MEC wherever demand emerges. This also accommodates seasonal and event-driven traffic patterns that exceed the economic reach of fixed deployments. Disaster response also benefits from this capability because aerial platforms reach affected areas before terrestrial infrastructure recovers. Satellite systems complement the same architecture across these regions, although propagation delays and power requirements limit their suitability for latency-sensitive workloads [6].

Aerial platforms deliver these benefits under constrained operating conditions. UAVs carry finite onboard energy and limited computing capacity, while link quality varies with position and surrounding environmental conditions. Task completion depends on these constraints, so they govern service delivery beyond resource availability. IoT devices add another dimension because the tasks they offload differ in urgency and in the level of assurance required from a serving UAV. Trust quantifies how consistently a UAV sustains service and compares that measure against the requirements each device declares.

1.1 Trust Evaluation for Aerial Edge Computing

Network conditions change continually for deployed aerial platforms. Task execution steadily depletes UAV energy, and the tasks currently in progress determine available computing capacity. UAV position further shapes link quality, and the operating context governs exposure to adversarial activity [7]. IoT-UAV associations formed without accounting for these conditions may therefore direct tasks to aerial platforms unable to complete them. Trust evaluation addresses this difficulty by scoring how consistently a UAV completes the tasks it accepts. An IoT-UAV association proceeds only once this score satisfies what the prospective IoT device requires. However, establishing such a measure in aerial edge computing raises several challenges:

- **Multi-Dimensional Characterization of Trustworthiness:** UAV trustworthiness spans historical task performance, operational stability, and resilience against security threats, each shaped by distinct factors. Historical performance varies independently of residual energy, and link stability varies independently of exposure to security threats. Single-dimension measures therefore overlook the failures that the remaining dimensions would reveal [8].

- **Resource Constraints:** UAV-mounted MEC servers carry finite energy and computing capacity, and every offloaded task they accept consumes both. A UAV serving many IoT devices early in a deployment therefore holds less capacity for the requests that arrive later. IoT-UAV association decisions consequently determine the service each UAV can offer for the remainder of its deployment.

- **Unbiased Trust Assessment:** Self-reported trust scores are vulnerable to bias, since a UAV has no external means to verify its own claims. IoT devices observe only the UAVs they have previously engaged [9]. Fair comparison across UAVs therefore requires an external observer. This observer must receive telemetry from every platform and apply consistent evaluation criteria. HAPs operate at altitudes that place the full UAV population within their coverage. Full-network visibility consequently suits HAPs to this assessment role.

Various approaches have incorporated trust into aerial network operation. Reinforcement learning frameworks for collaborative aerial computing used trust to stabilize policy updates [10]. Federated learning schemes assessed UAV reliability from behavioral consistency and contribution frequency [11]. Trust-driven offloading separated reliable regions from compromised ones through spatial trust assessment [12]. In [13], the authors investigated routing mechanisms for time-sensitive tasks and evaluated trust from transmission success probability. Collectively, these works treated trust as a measurable quantity in aerial networks. They applied trust within a specific mechanism and did not consider trust as a general input to network operation.

1.2 Multi-Criteria Optimization for Aerial Edge Networks

Trust scores determine whether a UAV satisfies the requirements of a given IoT device. The aerial network must still assign IoT devices across platforms and divide capacity among them. Allocation weighs trust alongside deployment cost, task latency, and connectivity (the number of devices that receive service) [14]. These four criteria can pull against one another when the network selects an association. A UAV may satisfy the trust requirement yet still exceed the latency bound the association imposes. Hence, solving the multi-criteria problem for a large IoT population raises several fundamental challenges:

- **Competing Optimization Objectives:** UAV computing capacity and energy limit the number of serviceable IoT devices. The admission of one IoT device therefore withholds capacity from another. Trust further restricts the available choices, since only UAVs that meet a device's requirement can serve it among those providing coverage. Cost and latency then determine which qualifying assignments the network should prefer. These criteria compete directly, so strengthening one often compromises another.

- **Scalability under Binary Decisions:** An IoT device either offloads a task to a single UAV or retains it locally. The IoT-UAV association decisions are therefore binary. Capacity and trust constraints couple these binary variables and place the problem among mixed-integer formulations [15]. The number of candidate assignments grows exponentially with the number of devices and platforms. Evaluating every candidate assignment thus becomes computationally expensive as deployments scale. Aerial edge computing therefore depends on near-optimal solutions that a system computes at sustainable cost.

- **Trust Assurance and Connectivity Trade-Off:** Stricter trust thresholds at IoT devices reduce the number of qualified UAVs, so the network admits fewer devices overall. Threshold selection therefore determines how much connectivity a deployment exchanges for greater trust assurance.

Solution strategies for trust-aware association include greedy assignment [16], relaxation of the binary variables followed by rounding [17], and exact optimization through branch-and-bound [18]. Greedy methods commit to each device sequentially and forgo improvements that joint assignment evaluation can provide. Rounding methods discard information recovered by the relaxation. Branch-and-bound attains the optimum, but its computational cost increases steeply with the number of devices and platforms. Aerial edge computing therefore requires methods that treat trust as a binding condition while remaining tractable as device populations grow. This multi-criteria formulation establishes the foundation for addressing uncertainty in IoT spatial positions and residual energy levels.

1.3 Uncertainty and Heterogeneity in IoT Deployments

Trust-aware IoT-UAV association assumes that IoT devices report their state accurately. It further requires that their tasks place comparable demands on a serving UAV. In practice, localization and reporting mechanisms introduce errors into the spatial positions and residual energy levels that the network receives. Devices admitted on a favorable report may exhaust their energy or move outside coverage before their tasks complete. Task demands vary as well, since urgent monitoring and routine sensing draw on the same UAV capacity but tolerate different delays. Realistic IoT populations therefore raise several fundamental challenges for trust-aware optimization:

- **Robustness to Uncertainty:** IoT devices report positions and residual energy levels to the network, and those values may deviate from actual device state. Localization mechanisms introduce uncertainty, while energy consumption follows irregular patterns [19, 20]. Reporting inaccuracies can cause UAVs to serve devices that fall outside coverage or lack sufficient energy to complete their offloaded tasks. The association must therefore remain valid across the range of possible device states to maintain service under these uncertainties.

- **Differentiated Handling of Task Priority:** IoT devices generate tasks that differ in deadline urgency, data volume, and operational criticality [21, 22]. Energy uncertainty compounds this heterogeneity, since urgency and energy sufficiency vary independently across the device population.

Prior studies in aerial edge computing address energy uncertainty and task prioritization in isolation from trust evaluation [22–24]. A framework that governs IoT-UAV association through UAV trustworthiness, IoT state uncertainty, and task prioritization together therefore remains absent from the literature. The following section considers networks in which IoT devices move and the control architecture assumes a decisive role in system performance.

1.4 Learning-Based Control for Mobile Aerial Networks

The framework described so far assumes a device population that holds its position throughout a deployment. In practice, IoT device positions continue to change over time. UAV placements optimized for an earlier distribution gradually fall behind this shift. Some IoT-UAV associations therefore become infeasible. UAVs must reposition as the IoT population redistributes. This repositioning proceeds on a slower timescale than the association and resource allocation decisions. Coordinating decisions across both timescales raises several fundamental challenges:

- **Decision-Making under Continuous Change:** The movement of IoT devices can invalidate an IoT-UAV association pattern shortly after the network computes it. This decision therefore requires continual revision across the operating period of the aerial network [25, 26]. Optimization methods that solve the problem from scratch operate too slowly for this rate. This motivates policies trained to respond as conditions evolve.

- **State Information for Coordinated Control:** Coordinating UAV repositioning and IoT-UAV association requires access to device positions, UAV resources, and trust scores across the coverage area. Central collection of these quantities scales with the device population. The resulting signaling overhead in turn delays the decisions it informs [27, 28].

Reinforcement learning has been applied extensively to UAV resource allocation and trajectory control [29]. The degree of centralization in these schemes follows from implementation convenience rather than analysis of what each architecture delivers [30]. Trust has likewise remained outside the learning objective. A policy trained across a spectrum of control architectures may or may not sustain the trust IoT devices require.

1.5 Problem Statement

We propose a trust-aware aerial edge computing framework in which UAVs deliver MEC services to IoT devices under HAP supervision. The framework scores the trustworthiness of each UAV on reputation, operational stability, and security assurance. The IoT devices declare their respective trust requirements before offloading a task. Trust evaluation proceeds at the HAP layer, which observes the full UAV population and therefore assesses each platform without the bias of self-reporting. We formulate the resulting IoT-UAV association and resource allocation task as a multi-criteria optimization problem over trust, connectivity, deployment cost, and task latency. The solution proceeds in two stages, where clustering over the IoT device distribution determines UAV deployment before resource optimization assigns IoT devices to UAVs.

IoT device reports of position and residual energy carry inherent error, arising from localization and reporting limitations. The framework extends to sustain service despite this imperfect knowledge. The formulation bounds both quantities by their maximum expected deviation and evaluates candidate IoT-UAV pairs under worst-case conditions. This conservative check builds robustness into the association decision rather than correcting for failures afterward. Serving UAVs supplement the energy of devices falling short of the transmission threshold. This support restores eligibility that the reported values alone would have denied.

IoT tasks range from routine monitoring to time-sensitive alerts, demanding different levels of scheduling urgency. The framework extends to capture this variation through a priority model that combines urgency, data volume, and operational criticality into a single score. Trust determines which UAVs qualify to serve a device. Priority ranks competing devices for the available capacity. The same energy support sustains these devices' continued operation.

The final setting considers networks where IoT devices move across the coverage area. Such movement changes both channel conditions and the distribution of service demand. As devices move, the trust a UAV provides can drift from the level required by its assigned devices. We propose a trust model that tracks this drift using energy sustainability, task reliability, and service freshness. Trust-weighted clustering assigns UAVs to device groups based on the trust required by each group. A mismatch between delivered and demanded trust then triggers repositioning across clusters. We compare reinforcement learning schemes with decreasing levels of centralization. The range extends from a single agent with complete network visibility to fully distributed UAVs that reach agreement through peer consensus.

1.6 Thesis Objectives

The objective of this thesis is to establish trust as an operational quantity within aerial edge computing. The network allocates UAV service based on what each platform demonstrably delivers rather than on availability alone. Objectives progress from a centrally supervised framework under static conditions to distributed control under IoT device mobility.

- **Objective 1**: Evaluate UAV trustworthiness through a set of behavioral and operational metrics. Couple the trust measure with IoT-UAV association in aerial edge computing. Decompose the problem into UAV deployment and resource optimization stages for tractability. Design a low-complexity algorithm with near-optimal performance, and compare it against optimal, heuristic, and relaxation-based approaches.

- **Objective 2**: Model uncertainties in the spatial positions and residual energy of IoT devices within the trustworthy aerial edge computing framework. Admit IoT devices only when service remains valid across the defined uncertainty range. Provide UAV energy harvesting support for devices with insufficient residual energy. Investigate the robustness of the proposed and comparative approaches to these uncertainties.

- **Objective 3**: Incorporate the heterogeneity of IoT tasks in trustworthy aerial edge computing through task prioritization. Treat trust as a service requirement and priority as a scheduling mechanism within one formulation. Sustain service for high-priority devices under uncertain IoT residual energy, and evaluate the number of supported IoT devices.

- **Objective 4**: Formulate a time-varying model for IoT device mobility and UAV positioning in trustworthy aerial edge computing. Devise a UAV deployment strategy that weights trust and repositions UAVs as trust demand shifts. Analyze the resulting performance across different levels of control centralization.

1.7 Thesis Contributions

This dissertation addresses these objectives with the following major contributions:

- **Contribution 1 - Trust-Driven Multi-Criteria Optimization Framework:** We propose a UAV trustworthiness model spanning reputation, operational stability, and security assurance, each weighted according to mission requirements. HAPs evaluate each UAV from telemetry and coordinate optimization across the network. UAV placement proceeds through k-means clustering, after which the framework resolves

IoT-UAV associations and resource shares. We propose the penalty-guided optimization (PGO) algorithm, which attains near-optimal connectivity at substantially lower computational cost than the branch-and-bound (BMK) benchmark. The algorithm further exceeds the greedy (GDY) and baseline relaxation (BRE) alternatives across the trust thresholds and network sizes examined.

- **Contribution 2 - Trustworthy Aerial Networks under IoT Uncertainties:** We propose a hybrid trustworthiness model that combines operationally coupled trust metrics multiplicatively. This reflects how a stable communication link cannot offset a UAV approaching energy exhaustion. The formulation admits IoT positions and residual energy as bounded uncertainties and evaluates candidate pairs under worst-case realizations. Serving UAVs deliver harvesting support to devices whose reserves fall below the offloading requirement. PGO tracks the optimal benchmark in connectivity and trust efficiency under both intelligent (ID) and random (RD) deployment as this uncertainty widens. GDY and BRE degrade markedly under the same conditions.

- **Contribution 3 - Trust- and Priority-Aware Resource Management:** We propose a task prioritization model that adds deadline urgency, data volume, and operational criticality into a quantized score. The accompanying trust model multiplies its constituent metrics, requiring every dimension to prove adequate before a UAV qualifies to serve a device. Priority-weighted clustering positions UAVs toward urgent-demand hotspots across the coverage area. The framework admits a device only when its residual energy combined with any harvesting support meets the task's requirement. Trust thereby determines which UAVs may serve a device. Priority orders the devices competing for capacity. PGO sustains near-optimal connectivity as multi-criteria weighting shifts toward trust, priority, cost, or latency. This flexibility allows a deployment to favor the criterion its mission demands, without sacrificing service.

- **Contribution 4 - Trust-Driven Control Architectures for Dynamic IoT:** We propose a multi-factor trust model spanning energy sustainability, task reliability, and service freshness. This model updates at every time slot as IoT devices move across the coverage area and network conditions evolve. We design trust-weighted K-means (TWK) deployment, which assigns UAVs to device groups by trust demand. TWK repositions these UAVs when delivered and demanded trust diverge. Random deployment (RND) and fixed deployment (FXD) form the two baselines in this comparison. RND repositions UAVs without regard to trust conditions, whereas FXD omits repositioning entirely and keeps each UAV at its initial position throughout an episode. A two-timescale decomposition separates this repositioning from the slot-level offloading decisions. Reinforcement learning architectures of decreasing centralization resolve these decisions. Comparison spans centralized deep Q-network (DQN) learning, centralized training with decentralized execution (CTDE), and a fully distributed scheme (FDS) under Laplacian consensus. Reward and trust performance diverge across the three architectures. Centralized control achieved higher reward, while the distributed scheme sustained higher trust.

1.8 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 – Background and State of the Art: This chapter provides essential background and a focused review of related research on trust-aware resource management in aerial edge computing. It first examines how prior work models UAV trustworthiness and formulates multi-criteria optimization for IoT-UAV association and resource allocation. The review then turns to prior treatments of uncertainty and task heterogeneity across IoT deployments, followed by learning-based control for networks operating under changing conditions. The chapter closes with open research challenges relevant to trustworthy aerial edge computing.

Chapter 3 – Trust-Driven Multi-Criteria Optimization Framework for Aerial MEC Enabling Resilient IoT Services: This chapter presents the trust-aware aerial edge computing framework. It presents the system model spanning IoT devices, UAVs, and HAPs, the multi-metric trust formulation, and the cost and latency models that complete the utility function. Trust enters the optimization as both an objective term and a binding constraint, which couples UAV evaluation to the IoT-UAV association decision. A two-stage methodology solves the problem, with clustering determining UAV deployment and the proposed PGO algorithm resolving the association and resource allocation decisions. The chapter closes with a performance analysis comparing PGO against the optimal BMK, GDY, and BRE algorithms across trust thresholds, weight configurations, IoT density, and UAV availability.

Chapter 4 – Toward Trustworthy Aerial Networks: Modeling IoT Spatial and Residual Energy Uncertainties: This chapter addresses IoT devices whose reported state differs from their actual condition. It presents the hybrid trustworthiness model, the spatial and residual energy uncertainty models, and the energy harvesting mechanism that supports devices below the offloading requirement. Uncertainty enters both the deployment stage through worst-case distance evaluation and the association stage through the admission of perturbed device state. The chapter closes with a performance analysis across trust thresholds, perturbation levels, and a comparison of intelligent against random UAV placement.

Chapter 5 – Trust- and Priority-Aware Resource Management for Reliable UAV-Assisted MEC in 6G IoT Networks: This chapter addresses IoT devices whose tasks differ in urgency and resource demand. It presents the task prioritization model, the capability-driven trust formulation, and the energy perturbation model applied to reported reserves. Priority enters both the placement decision through weighted clustering and the admission decision alongside trust and energy sufficiency. The chapter closes with a performance analysis across weight configurations, trust thresholds, and network scale.

Chapter 6 – Trust-Driven Aerial Edge Computing for Dynamic IoT Networks across Control Architectures: This chapter addresses networks where IoT devices move throughout the deployment. It presents the mobility and channel models, the multi-factor trust formulation, and the two-timescale decomposition separating UAV repositioning from task offloading. Trust-weighted clustering governs UAV repositioning, with random and fixed placement providing the comparison. Reinforcement learning architectures of decreasing centralization resolve the offloading decisions. The chapter closes with a performance analysis spanning the repositioning threshold, the trust threshold, and increasing IoT device density.

Chapter 7 – Conclusion: This chapter summarizes the contributions and examines the findings across the proposed frameworks. The chapter then identifies future directions for trustworthy aerial edge computing.

Table 1.1 draws together the preceding discussion, linking each objective to its research gap. It further specifies the methodology adopted to close that gap and the chapter above where the corresponding work appears. This mapping shows how the objectives and contributions fit within the overall thesis structure.

Table 1.1: Mapping each thesis objective to its research gap, methodology, and chapter.

Objective	Research Gap	Methodology	Chapter
1	Trust confined to a single mechanism rather than a general input to network optimization	Additive trust model with K-means placement and PGO algorithm	3
2	Energy and position uncertainty treated apart from trust	Hybrid trust with robust admission and harvesting under uncertainty	4
3	Task heterogeneity treated apart from trust-based association	Multiplicative trust with priority-weighted clustering and multi-criteria PGO algorithm	5
4	Centralization chosen by convenience, not analyzed for its effect on trust	Time-varying trust with TWK across DQN, CTDE, FDS	6

A complete list of cited references appears at the end of the thesis.

1.9 Summary

This chapter introduced the motivation and context for trustworthy aerial edge computing in 6G networks. Trust requires systematic evaluation where UAVs deliver MEC services on demand. Association and resource allocation decisions couple to that evaluation and determine which IoT devices receive service. Later sections examined uncertainty in reported device state, heterogeneity in task demand, and the control architectures that govern these decisions under device mobility. Four objectives and their contributions emerge from these considerations. The contributions couple trust with allocation, bound the effect of uncertain device state, admit devices by priority, and compare control architectures across levels of centralization. The following chapter develops this foundation through a detailed review of the background concepts and related research.

Chapter 2

Background and State of the Art

2.1 Aerial Networks for 6G IoT Services

6G networks set ambitious performance targets that span global coverage, high data rates, real-time services, and support for dense deployments [2]. Terrestrial infrastructure carries most of this load, while non-terrestrial elements extend service into remote and infrastructure-sparse areas. Aerial networks fill this role through UAVs and HAPs, which reach diverse service areas on demand at low cost. These platforms carry limited resources compared with terrestrial installations. A UAV works within whatever energy, computing capacity, and link quality it has, and all three degrade over a deployment. Trust reduces these conditions to one value that the network uses when selecting a UAV for a given IoT device.

UAVs read their own energy, computing load, and link quality, while the state of neighboring platforms lies outside their reach. The HAP layer holds the complementary view, receives reports from the UAVs within its coverage, and evaluates them against one another. Assessment from this layer ranks the UAV population on consistent terms rather than accumulating isolated self-descriptions. The network draws on this ranking to identify which UAVs meet the requirement a given IoT device declares.

A qualifying UAV still holds finite capacity, which the network divides among the IoT devices competing for it. Computing and energy limits make this division competitive, as capacity committed to one IoT device becomes unavailable to the others a UAV serves. Deployment cost and task latency bear on the same choice, and improvement along either constrains what the network achieves on the other. Formulations in the literature differ in which of these criteria they admit and in the weight each one receives.

The network solves this allocation for the conditions holding at the moment of computation. Aerial deployments move away from those conditions continuously, as IoT devices relocate and UAV state evolves through service. Recomputation produces a solution matched to the new conditions, though gathering network-wide state grows costly as the IoT population expands, and distributing the decision trades that cost against the coordination a complete view affords. The following sections review trust evaluation, resource allocation, and control under these conditions.

2.2 Trust Modeling in Edge and IoT Systems

Trust in terrestrial deployments has been studied well before its application to aerial networks, and this body of work established the vocabulary that the aerial literature inherits [31, 32]. The earliest approaches derived a score from accumulated interaction records, measuring how consistently a participant had honored past commitments. The authors of [8] extended this principle across several attributes in mobile crowdsensing, evaluating workers on the reliability of the data they contributed and matching them to tasks their record supported. The authors of [33] applied the same measure to incentive design in distributed information propagation, where the score determined what a participant received for forwarding information rather than what work it was assigned.

The outcomes of a participant's recent tasks describe its present state more accurately than records accumulated earlier, which shapes how the literature combines observations into a single score. The authors of [34] weighted each interaction by its recency and frequency, so that recent behavior carried more influence than earlier evidence. Their scheme applied the resulting values to cluster resource nodes before offloading, which placed trust ahead of the allocation decision rather than beside it. The authors of [35] paired the score with a confidence estimate, deriving both from traffic observations across connected micro-clouds so that a system distinguished a strong score from a well-supported one.

Conditions at an edge server shift throughout its service period, so a trust score computed at one instant loses accuracy as the server continues operating. The authors of [36] addressed this through concept drift adaptation, updating trust estimates as operating conditions evolved and sharing the resulting knowledge among edge nodes to reduce dependence on the core network. The authors of [37] separated belief, disbelief, and uncertainty as distinct quantities, which distinguished a server observed to perform poorly from one about which little evidence existed. Their allocation adjusted through reward and penalty functions derived from these values. Trust also entered server selection directly, governing which edge server received a task under competing performance and cost criteria [38–41].

In summary, the sustained participation and continuous observation form the basis of the trust models discussed above [8, 33–37]. UAVs complete a single mission rather than accumulating a record across repeated engagements, which leaves the network without the history these models depend on. The link to the ground varies with UAV position and terrain throughout that mission, so the observation channel they treat as steady fluctuates instead. Trust evaluation for aerial networks therefore requires measures that a UAV establishes within the period it serves, drawn from conditions the network observes directly rather than from a record accumulated over time.

2.3 Trust in UAV-Assisted Networks

Trust in aerial networks has been approached along three lines that differ in what the score describes and where it enters the decision. The subsections below examine each in turn, covering trust computed as an assessment separate from allocation, trust embedded within the offloading decision, and trust attributed to entities other than the serving UAV.

2.3.1 Trust as a Separate Assessment

Studies in this category treat trust computation as the contribution in itself, without linking the resulting measure to IoT-UAV association. The authors of [35] grounded their approach in measurement theory, deriving trustworthiness and confidence as paired metrics from real-time traffic observations across IoT-connected micro-clouds. The paired representation let operators reconfigure computing resources according to observed reliability, though the evaluation remained decoupled from task offloading and from the resource optimization it might have guided.

The survey literature treats trust the same way, identifying it as necessary without formalizing how it should be computed. The authors of [42] reviewed machine learning techniques for UAV communications and emphasized the need for trust in autonomous systems, where model-driven decisions introduce the possibility of error. The authors of [43] analyzed UAV-enabled MEC task offloading and identified computational reliability and secure communication links as the conditions on which trust depends. Surveys of HAP frameworks [5] and autonomous UAV swarms [44] extended this pattern to the architectural level, describing how platforms coordinate without quantifying the trust that governs their selection.

2.3.2 Trust Embedded in the Offloading Decision

Trust operates inside the offloading decision rather than as a separate assessment, so the score determines which candidates the allocation may select. Trust-weighted clustering was used in [34] to screen candidates ahead of offloading. Multiple feedback sources contributed to a credibility value computed for each resource node. Sequential least-squares programming solved the resulting nonlinear program, which minimized system cost under task completion and device capacity constraints. In [45], the problem was formulated as a joint minimization of energy consumption and task processing delay. Auction utility was maximized under bid range, coverage, and trust threshold constraints. The solution combined K-means clustering with a genetic algorithm and a Bayesian-Nash equilibrium. That formulation treated UAVs as relays and auctioneers rather than as computing entities.

Trust also governed the selection of serving nodes under explicit cost objectives. A mixed-integer nonlinear program was proposed in [40] to minimize the difference between cost and expected revenue under user association, sub-band allocation, and task delay constraints. Greedy random association with interference-based sorting and gradient descent solved the program. Trust drew on observed service consistency and peer recommendations to address the credibility of reported resource availability. The authors of [41] defined trust as a goal-achieving criterion in collaborative UAV and ground vehicle systems, minimizing value-of-service discrepancy as a mixed-integer program under communication range, contact duration, and UAV energy constraints. Adaptive K-means clustering with priority-weighted graph matching resolved the assignment, clustering subtasks by goal similarity before ordering them by priority. In [13], the authors evaluated trust through transmission success probabilities, which grounded the measure in link behavior rather than in the computing service a UAV delivers.

2.3.3 Trust Attributed Beyond the Serving UAV

Studies in this category attach the trust score to an entity other than the UAV executing the task, so a platform derives standing from its surroundings rather than from the service it delivers. The authors of [12] partitioned the coverage area into credible and suspicious regions and optimized UAV trajectories to favor secure paths through higher-trust areas, which improved task completion and latency. Trust in that formulation described geography, so a UAV operating within a credible region carried a score its own condition had not established.

Federated formulations place the measure on the data a device contributes. The authors of [11] quantified UAV reliability through behavioral consistency, data accuracy, and contribution frequency, which confined the assessment to data generation rather than to the edge computing service a UAV provides. The authors of [46] evaluated IoT device trust through direct interaction histories and indirect UAV recommendations within a blockchain-supported federated learning framework. The joint resource allocation balanced trust verification against learning latency, communication cost, and energy consumption. Their scheme suppressed malicious updates effectively, though the evaluation covered federated learning contributions rather than UAV capability.

Threat detection provides a third basis for attribution, where the score reflects exposure to adversarial activity rather than service capability. The authors of [47] secured UAV visual sensors through a belief-function evidence fusion model, combining direct analysis of recognized attacks with indirect estimation from pre-trained models for unknown threats, and resolving conflicting information through weighted fusion over basic belief assignments. Trust supported data collection in [48] and [49], where UAV-collected baselines verified vehicle-reported measurements. High-trust IoT devices served as aggregation points. Genetic and ant colony methods optimized the trajectories and balanced collection cost against verification coverage.

2.3.4 Lessons Learned

In the reviewed works, generally one aspect of UAV behavior was considered at a time. The authors of [13] derived the measure from transmission success. Threat detection served the same purpose in [47], and behavioral credibility across repeated interactions in [34]. The model in [11] came closest to a composite treatment and combined behavioral consistency, data accuracy, and contribution frequency. The resulting measure still characterized data generation rather than the computing service a UAV performs.

Beyond the measure itself, the works differ in what the trust threshold governs once it enters the allocation. In [45], a trust threshold constrained participation and established eligibility instead of assigning weight. The threshold applied to relay and auction roles rather than to computing capability. Trust as one criterion among several was studied in [40,41,46], where a low score lowered the objective value rather than removing the candidate.

HAPs enter the reviewed architectures as coordinating entities without carrying any trust computation. The authors of [10] and [50] placed HAPs within their formulations, where trust stabilized policy updates rather than describing the UAVs beneath them. The authors of [48] and [49] verified reported measurements against independently collected baselines, applying that verification to IoT devices rather than to the UAVs performing the collection. Assessment of the serving UAV therefore falls outside the mechanisms these studies provide. Table 2.1 summarizes the trust formulations across the reviewed studies.

Table 2.1: Summary of trust formulations in the reviewed UAV-assisted networks.

Ref.	**Network elements**	**Trust subject**	**Trust role**	**Solution approach**	**Remarks**
Trust as a separate assessment					
[5]	UAV, HAP	Not addressed	Not addressed	Survey	HAP frameworks for future wireless networks were reviewed without treating trustworthiness.
[35]	IoT, micro-clouds	Edge node	Assessment	Measurement-theory framework	Trustworthiness and confidence were paired as complementary metrics, and the evaluation remained decoupled from offloading and resource optimization.
[42]	UAV	UAV	Assessment	Survey	Machine learning techniques for UAV communications were surveyed, identifying trust as a requirement for autonomous operation.
Continued on the next page ⏵⏵⏵					

Table 2.1 continued from the previous page

Ref.	Network elements	Trust subject	Trust role	Solution approach	Remarks
[43]	IoT, UAV	UAV	Assessment	Mapping study	UAV-enabled MEC offloading was analyzed, naming computational reliability and link security as the conditions on which trust depends.
[44]	UAV, HAP	Not addressed	Not addressed	Survey	An architecture for autonomous UAV swarms was presented, addressing coordination rather than trust.
Trust embedded in the offloading decision					
[13]	IoT, UAV	Link	Criterion	Trust-based routing	Transmission success probabilities defined the measure, which described link behavior rather than computing service.
[34]	IoT, broker, edge servers	Edge node	Screening	Sequential least-squares programming	Multifeedback credibility guided trust-weighted clustering ahead of offloading, minimizing system cost under task completion and device capacity constraints.
[40]	Mobile users, edge servers, cloud	Edge server	Criterion	Greedy association, gradient descent	Trust was derived from service consistency and peer recommendations within a mixed-integer nonlinear cost formulation.
Continued on the next page ⏵⏵⏵					

Table 2.1 continued from the previous page

Ref.	Network elements	Trust subject	Trust role	Solution approach	Remarks
[41]	UAV, ground vehicles	Goal match	Criterion	Adaptive K-means, priority-weighted graph matching	Value-of-service discrepancy was minimized through trust-guided matching, without persistent UAV capability assessment.
[45]	IoT, UAV, edge servers	Relay, auctioneer	Constraint	K-means clustering, genetic algorithm, Bayesian-Nash equilibrium	A trust threshold constrained auction participation, though UAVs relayed tasks to edge servers rather than executing them.
Trust attributed beyond the serving UAV					
[11]	IoT, UAV	Data contribution	Criterion	Federated learning	Behavioral consistency, data accuracy, and contribution frequency were combined, confining the assessment to data generation.
[12]	IoT, UAV	Region	Trajectory	Online optimization	Credible and suspicious regions guided UAV paths, so trust described geography rather than the serving UAV.
[46]	IoT, UAV	IoT device	Criterion	Blockchain-assisted federated learning	Direct interaction histories and indirect UAV recommendations were balanced against learning latency and energy consumption.
Continued on the next page ▸▸▸					

Table 2.1 continued from the previous page

Ref.	Network elements	Trust subject	Trust role	Solution approach	Remarks
[47]	UAV sensors	Threat exposure	Assessment	Belief-function evidence fusion	Recognized attacks were analyzed directly and unknown threats estimated indirectly, without aerial edge computing context.
[48]	IoT, vehicles, UAV	IoT device	Verification	Genetic algorithm trajectory planning	Reported measurements were checked against UAV-collected baselines, directing verification toward IoT devices.
[49]	IoT, UAV, cloud	IoT device	Verification	Ant colony trajectory optimization	High-trust IoT devices were selected as aggregation points following UAV-verified inspection.

The reviewed measures typically characterize a single criterion, such as transmission success or contribution frequency, or attribute trust to an entity other than the serving UAV. This narrows what each measure can reveal about the UAV's overall service capability. The proposed framework separates these considerations explicitly. Trust assessment refers to computing this composite of reputation, stability, and security at the HAP, independent of the UAV it describes. Trust as a constraint concerns how this score is used once computed. The trust threshold the IoT device declares first determines whether a UAV qualifies to serve it. This score also enters the objective as a weighted term, alongside cost, latency, and connectivity.

2.4 Resource Optimization in UAV-Assisted MEC

Terrestrial deployments treat server positions as fixed parameters, whereas aerial networks include them among the decision variables. Joint optimization of UAVs placement and IoT association expands the feasible region combinatorially, so most studies decompose the resulting mixed-integer nonlinear program rather than solving it directly. The authors of [51] separated trajectory planning from resource allocation across two timescales, solving the slower subproblem through proximal policy optimization and the faster one through greedy selection with successive convex approximation. The authors of [52] applied block coordinate descent to offloading decisions, sub-band assignment, power control, and UAV deployment, addressing the four resulting subproblems through matching games and convex approximation. The authors of [53] derived the association variables in closed form through Lagrangian duality, which reduced the formulation to a deployment problem solved by federated multi-agent deep reinforcement learning.

The reviewed formulations optimize delay, energy, and connectivity as simultaneous objectives. The authors of [54] minimized task completion delay and UAV energy consumption while maximizing the number of offloaded tasks. Their solution separated offloading, resource allocation, and trajectory control, addressing each subproblem through threshold rounding, Karush-Kuhn-Tucker conditions, and successive convex approximation. The authors of [55] pursued energy consumption alone in a HAP-aided network under non-orthogonal multiple access, jointly designing beamforming, power allocation, and platform position through block coordinate descent. The authors of [56] minimized processing delay for divisible tasks under data dependency and UAV energy constraints, allowing partial offloading to distribute load across cooperating UAVs. A secrecy-aware offloading objective was proposed in [57] to maximize the minimum average secrecy capacity against eavesdroppers. The UAV in that work offloaded tasks to ground servers instead of serving IoT devices.

Branch-and-bound solves these programs exactly by enumerating the integer feasible set under successive bounding [15]. The number of candidate assignments grows exponentially with the count of IoT devices and UAVs. Exact methods therefore become computationally prohibitive at deployment scale. Relaxation-based methods remove the integrality restriction and recover binary values through rounding. A rounded solution may violate constraints that the relaxed one satisfied. Heuristics that minimize the integrality gap over a partitioned space of integer variables are proposed in [17] to address this loss. A penalty term added to the relaxed objective in [58] guided fractional values toward binary assignments and removed the separate rounding stage. Decomposition-based methods reduce the same complexity by partitioning the problem. Alternating direction approaches are surveyed in [59], where the resulting subproblems solve independently and iteratively.

2.4.1 Lessons Learned

The reviewed formulations decompose the problem before solving it, whether through timescale separation [51], block coordinate descent [52, 55, 57], or analytical elimination of a variable set [53]. Intractability at deployment scale makes this a requirement rather than a design preference, and the studies differ in which variables they separate and in what order. Their objectives combine delay, energy, connectivity, and cost, with the authors of [54] carrying three simultaneously while the authors of [55] and [56] minimized a single quantity and expressed the remainder as constraints. The relative weighting of these criteria remains a decision for the network operator.

The reviewed studies determine which IoT devices a UAV admits from coverage and available computing capacity. Their objectives cover delay, energy, connectivity, cost, and in one case secrecy, so the serving UAV enters each formulation through its present resources rather than through any record of how it has served. HAPs appear as optimization variables in [55] and [52], positioned for coverage or energy rather than assessing the UAVs they coordinate. Trust supplies the dimension these studies leave outside the formulation, and the aerial trust literature reviewed earlier has developed without the optimization structure described here. Table 2.2 summarizes the objectives, problem classes, and solution approaches across the reviewed studies.

Table 2.2: Resource optimization approaches in reviewed UAV-assisted MEC studies.

Ref.	**Objectives**	**Problem type**	**Solution approach**	**Remarks**
[51]	Minimize completion time and energy	Long-term optimization	Two-timescale decomposition, proximal policy optimization, greedy algorithm, successive convex approximation	Terminal mobility and stochastic arrivals were modeled without trust in the assignment.
[52]	Minimize device and UAV energy	Mixed-integer nonlinear	Block coordinate descent, matching game, concave-convex procedure, block successive upper-bound minimization	Offloading, sub-band assignment, power control, and deployment were solved in sequence.
[53]	Minimize average operation time	Non-convex	Lagrangian duality, federated multi-agent deep reinforcement learning	Heterogeneous UAV capabilities were represented, with association derived analytically from the delay objective.
[54]	Minimize delay and energy, maximize offloaded tasks	Mixed-integer nonlinear, NP-hard	Threshold rounding, Karush-Kuhn-Tucker conditions, successive convex approximation	Three objectives were carried jointly without any measure of UAV condition.
[55]	Minimize sum energy consumption	Non-convex	Block coordinate descent, successive convex approximation	HAP position entered as a placement variable rather than a supervisory function.
[56]	Minimize processing delay	Non-convex	Convex reformulation with rounding, two-layer game-theoretic approximation	Partial offloading balanced load under task dependency and energy constraints.
[57]	Maximize minimum secrecy capacity	Mixed-integer non-convex	Block coordinate descent, successive convex approximation	The UAV offloaded to ground servers, inverting the serving relationship.

The reviewed works perform resource optimization based on energy reserves, channel conditions, and processing capacity, each evaluated at the instant of decision. Their objective functions do not carry a term reflecting how reliably a UAV has served in the past. The proposed framework incorporates this history directly. The same trust score assessed at the HAP enters the objective alongside energy and latency. A UAV's demonstrated performance thus shapes the allocation together with its present resources.

2.5 Operating Conditions in IoT Deployments

IoT populations in aerial deployments vary in their reported state, in the urgency of the tasks they generate, and in their position over the service period. Optimization formulations differ in how far they represent each condition, and the subsections below review the treatments the literature has developed.

2.5.1 Uncertainty in IoT Device State

The spatial position and residual energy of IoT devices reach the serving UAV through localization systems and periodic reporting. This information remains prone to inaccuracy, since localization error and irregular IoT energy consumption separate the reported values from the actual ones. The authors of [60] represented vehicle locations as uncertain regions within an intelligent transportation framework, so that decisions computed from reported coordinates accounted for the deviation those coordinates carried. The authors of [20] applied comparable reasoning to harvested energy, predicting available power as a bounded quantity rather than a single value so that management decisions held across the predicted range.

IoT-UAV association decisions depend on the values each IoT device reports, so deviation in those values affects which pairs the aerial network admits. The authors of [19] modeled IoT spatial perturbations within a digital twin framework, maximizing the number of served IoT devices while minimizing latency and resource costs. The authors of [18] developed a robust counterpart formulation, relaxing the binary variables and appending an integrality gap penalty before solving the continuous problem through sequential quadratic programming. The authors of [61] extended this formulation to task priority, directing computational resources toward higher-priority tasks under intelligent and random UAV placement. These studies model uncertainty in the reported state of the IoT device, while the serving UAV is only characterized by its capacity limits.

Energy harvesting extends the role of the serving UAV beyond computation in several formulations [23, 62, 63], supplementing the power available at IoT devices. The authors of [23] maximized computational efficiency in disaster scenarios under a power-splitting architecture that divided source power between communication and harvesting. Their solution proceeded in three stages, determining UAV placement before resolving association and the remaining transmission variables. The authors of [63] and [62] minimized latency while maximizing the number of associated IoT devices, treating harvesting as one constraint alongside computing, caching, and coverage limits. The authors of [24] addressed unpredictable task arrivals and computational resource availability in maritime deployments, converting long-term constraints into short-term ones through Lyapunov optimization before reformulating the problem as a Markov game.

2.5.2 IoT Task Heterogeneity and Prioritization

IoT tasks differ in how urgently they require completion, and scheduling research has represented this difference through deadlines. The authors of [21] scheduled newly arriving tasks in edge computing through a greedy online algorithm, which displaced already-scheduled tasks whenever the substitution met more deadlines. The authors of [64] derived the task order that minimizes deadline violations for a given set in linearithmic time. They also developed a separate online approach for randomly arriving tasks in linear time. Deadlines in these formulations distinguish tasks by their timing requirement alone, without reference to the operational significance each task carries.

Aerial deployments carry priority into the offloading decision, where it competes with the other criteria the formulation weighs. The authors of [22] formulated task offloading across cooperating UAVs as a mixed-integer program maximizing long-term average system gain. Their objective combined energy consumption and task delay, with task priorities entering alongside trajectory design and resource allocation. A latent-space deep reinforcement learning algorithm addressed the resulting discrete-continuous hybrid action space. The authors of [65] incorporated task priority into user association across integrated satellite, aerial, ground, and sea networks, formulating a binary linear program that maximized energy efficiency, resource utilization, and connectivity. The authors of [61] directed computational resources toward higher-priority tasks within their robust offloading formulation.

2.5.3 IoT Device Mobility

IoT devices move during service, and mobility models represent this movement within the network formulation. The authors of [66] adopted a random walk in a post-disaster MEC network, drawing velocity and direction independently from uniform distributions at each time slot. The authors of [67] applied the Gauss-Markov model, where velocity and direction at each slot depend on their values at the previous one. The resulting trajectories persist in velocity and direction rather than resetting at every interval.

Network conditions in aerial deployments vary over different timescales, and the reviewed formulations separate decisions by the rate at which each must respond. The authors of [68] allocated computing resources and resolved offloading through short-timescale mechanisms while addressing UAV trajectory control over the longer one. Their formulation maximized system utility across a non-convex mixed-integer nonlinear program, with a price-incentive model governing resource allocation and a matching mechanism governing offloading.

2.5.4 Aerial Network Control Architectures

IoT-UAV association, resource allocation, and UAV placement decisions may be computed centrally or distributed across the UAVs themselves. Reinforcement learning has been applied across these decisions in UAV communications and networking [29], with schemes placing control between full centralization and full distribution [67, 69, 70]. The authors of [67] assigned control to heterogeneous agents across an air-ground network, where overloaded aerial vehicles relayed tasks to HAPs. The authors of [69] adopted a comparable multi-agent structure for multi-UAV MEC, optimizing trajectory, task partition, and computing resources under channel and task uncertainty. The authors of [70] trained a single deep Q-network on the pooled experience of multiple UAVs, reporting reduced training time relative to training each agent separately. Centralized control requires state from every UAV to reach the deciding entity, while distributed control acts on what each UAV observes locally. The authors of [30] found that centralizing the critic offers no assured advantage across the multi-agent benchmarks they examined.

2.5.5 Lessons Learned

Uncertainty, priority, and mobility all characterize the IoT device rather than the UAV that serves it. Robust counterpart methods bound the position and residual energy a device reports [18, 19, 61], and harvesting restores devices whose reserves fall below the offloading requirement [23, 62, 63]. Priority models score the tasks these devices generate [21, 22, 64, 65], while mobility models track their positions between service intervals [66, 67]. These representations together determine which IoT devices the network admits to service.

IoT device admission determines which devices receive service, though completion depends on the serving UAVs. These UAVs enter the formulation through coverage and capacity, so the network distinguishes them by the resources they hold. The record of what a UAV has delivered falls outside this comparison, and one that has failed repeatedly presents the same resource profile as one that has completed every assigned task. Distributing the decision across UAVs [69, 70] relocates the comparison without adding to the quantities it examines. Trust supplies the measure that separates these two cases, and the reviewed formulations leave that measure outside their scope.

2.6 Research Gaps and Positioning of Dissertation

Aerial networks admit trust as a measurable quantity and multi-criteria optimization as a framework for allocating UAV resources. Trust modeling and resource optimization have developed along separate tracks. Trust measures attach to regions [12], data contributions [11], and device populations [46], and each captures one aspect of UAV behavior. The optimization frameworks of [51, 52, 54] settle IoT-UAV association under computing and energy constraints, where a UAV qualifies on the resources it holds rather than on its demonstrated trustworthiness. This dissertation couples a composite trust measure with that association inside a single optimization. Trust then determines which UAVs qualify to serve a device. Priority orders the devices competing for capacity, and bounded uncertainty in reported state governs admission. Centralized, hybrid, and fully distributed architectures resolve the same decisions under device mobility.

- **Trust-Agnostic Resource Allocation:** Trust governs aerial service delivery through the eligibility of UAVs to serve a given device. Trust scores derived outside the optimization framework therefore remain disconnected from that eligibility.

- **IoT Admission Criteria without a UAV Trust Measure:** Coverage and capacity describe the resources a UAV holds rather than the service it has delivered. IoT admission on these criteria alone leaves task completion dependent on a UAV assessed only by its resources.

- **Separate Treatment of IoT Uncertainty and Task Priority:** IoT deployments contain devices whose reported state is unreliable and whose tasks differ in urgency.

A formulation serving such populations must weigh reported state and task urgency within the same decision admitting devices to UAV service.

- **Architectural Trade-Off between Learning Performance and Trust:** Distributing control reduces the state a single entity gathers, while the resulting policies may differ in the trust they sustain. Architecture selection therefore requires evidence on reward and on trust, since an architecture leading on one may trail on the other.

This thesis addresses the four gaps through a trust-aware framework for aerial edge computing. Trust enters the optimization as a binding condition on IoT-UAV association, computed from observation across the UAV population rather than from what each platform reports about itself. The framework then extends to IoT devices whose reported state carries uncertainty and whose tasks differ in urgency, weighing both within the same admission decision. Device mobility completes the treatment, where trust-weighted deployment responds to shifting demand and a comparison across reinforcement learning architectures relates centralization to the trust a policy sustains.

2.7 Summary

This chapter reviewed the research related to trust-aware aerial edge computing. It examined how trust has been modeled in edge and IoT systems, and how the aerial literature has applied the measure to regions, data contributions, and serving UAVs. Resource optimization for UAV-assisted MEC followed, covering the formulations that couple UAV placement with IoT-UAV association and the decomposition methods that render them tractable. Operating conditions completed the survey, spanning uncertainty in reported IoT device state, heterogeneity in task demand, IoT device mobility, and the control architectures that resolve decisions as conditions change.

Trust measurement has developed apart from the IoT-UAV association decision, and IoT admission criteria have formed without a measure of UAV trustworthiness. IoT state uncertainty and task priority have received separate treatment, while the relation between learning performance and trust has remained unexamined. The following chapters present the frameworks addressing each gap, beginning with the trust model and multi-criteria formulation that the remaining contributions extend.

Chapter 3

Trust-Driven Multi-Criteria Optimization Framework for Aerial MEC Enabling Resilient IoT Services

3.1 Introduction

Aerial networks combine HAPs and UAVs in a two-tier architecture [71], extending coverage and bringing computation closer to IoT devices in areas with disrupted connectivity [72]. The reliability of this architecture rests on the trustworthiness of its platforms. Trust captures the effectiveness of UAVs in completing assigned tasks and sustaining dependable operation throughout deployment. It builds on residual energy, computational capacity, communication reliability, and security performance as indicators [9].

The authors of [8] and [33] investigated trust modeling in mobile crowdsensing and distributed social networks. Aerial networks place trust directly in operational decisions under strict latency and connectivity requirements, where topology and node availability change frequently. Emergency scenarios impose strict limits on UAV energy, computation, and communication [73]. Reliable service under these limits depends on coordination among UAVs

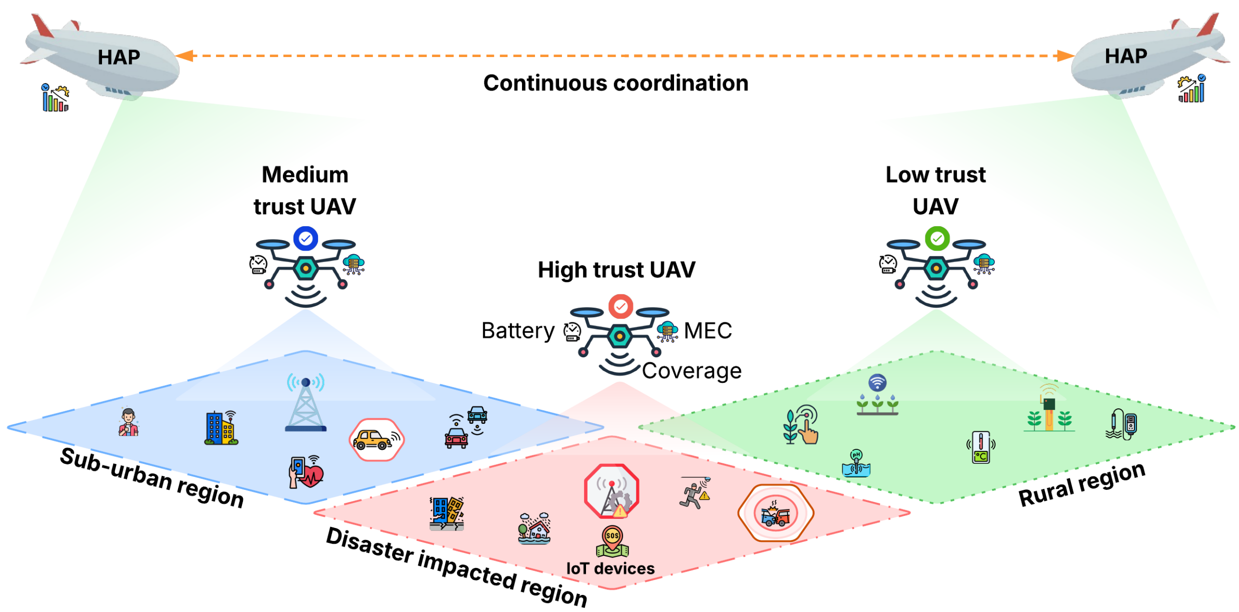


Figure 3.1: Trustworthy aerial edge computing framework where UAVs resource usage is optimized by HAPs to support diversified IoT service requirements.

that meets the trust requirements of heterogeneous IoT applications [65]. HAPs supply a centralized perspective for managing UAVs by allocating resources and prioritizing tasks. Trust-based coordination balances workloads and reduces service delays across the network.

Fig. 3.1 illustrates a trust-aware hierarchical aerial edge computing framework for 6G networks. Each UAV is assigned a trust level that reflects its ability to perform tasks, based on residual energy, computational capacity, communication reliability, and security performance. Trust is dynamic and evolves with these metrics, capturing the UAV's current operational reliability. Low-trust UAVs focus on non-critical applications and manage tasks such as rural smart-farming sensors, where occasional failures have minimal impact. Medium-trust UAVs support tasks like utility monitoring and infotainment, where service interruptions must be minimized. High-trust UAVs provide strong reliability and are assigned to critical, time-sensitive tasks, including disaster response, navigation, and personal tracking, where failures could have serious consequences. HAPs use these trust assessments to deploy UAVs effectively to coverage areas that match their trustworthiness requirements.

3.2 System Model and Problem Formulation

We consider a 6G-enabled aerial edge computing framework comprising IoT devices, UAVs, and HAPs. UAVs have been widely recognized as key enablers of 6G aerial networks due to their flexibility, mobility, and rapid deployment capability [74]. UAVs function as distributed MEC nodes for task execution, while HAPs enable coordinated trust evaluation and multi-criteria optimization for network-wide resource management. The IoT devices are represented by $\mathcal{N} = \{1, 2, \ldots, N\}$ and are distributed randomly across the ground area. UAVs are denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$ and operate at different altitudes. HAPs are represented by $\mathcal{K} = \{1, 2, \ldots, K\}$ and are fixed at high altitudes to provide centralized supervision and network optimization. The three-dimensional coordinates of the *n-th* IoT device, *m-th* UAV, and *k-th* HAP are denoted by $O_n = (x_n, y_n, 0)$, $O_m = (x_m, y_m, z_m)$, and $O_k = (x_k, y_k, z_k)$, where x and y indicate horizontal positions, and z represents the altitude.

IoT devices in the network are assumed to be resource-constrained in terms of bandwidth, computation, and power. They are assumed to execute computationally intensive tasks and cannot meet service requirements independently. To address this limitation, aerial MEC support is integrated to provide additional processing capabilities and support reliable task execution. UAVs exhibit heterogeneous processing capacities, energy availability, and trust metrics, which limit their ability to serve all devices uniformly. HAPs coordinate UAV deployment and oversee network-wide optimization, enabling autonomous operation in remote or infrastructure-limited regions. UAV placement is initially optimized to maximize coverage and is subsequently refined using sensing data collected by UAVs. Orthogonal frequency division multiple access (OFDMA) is adopted to achieve efficient spectrum allocation and mitigate interference during simultaneous and hierarchical communication between IoT devices, UAVs and HAPs [75].

In the proposed aerial MEC system, each IoT device can offload its computational tasks to its designated UAV. The task from the *n-th* IoT device to the *m-th* UAV is represented as: $F_{nm} = (D_{nm}, \delta_{nm}, C_m, L_{nm}^{\text{TASK}}, T_{nm})$, where D_{nm} denotes the data size in bits, δ_{nm} represents the computational workload in CPU cycles per bit, C_m represents the operational cost of the *m-th* UAV, including deployment and trust-related cost, L_{nm}^{TASK} indicates the computational latency, and T_{nm} corresponds to the trust of the *n-th* IoT device on the *m-th* UAV. Tasks and resource demands are assumed to be independent across IoT devices. The *n-th* IoT device transmits relevant task parameters, including data size, workload, and deadline, to the *m-th* UAV. A positive value of C_m and L_{nm}^{TASK} indicates that the task can be executed locally or offloaded to the *m-th* UAV via MEC. In addition, UAVs periodically relay their performance parameters such as residual energy, available computational resources, detected security threats, and local sensing information to the HAPs. This enables dynamic trust evaluations and allows HAPs to better coordinate UAVs for efficient service delivery to IoT devices.

We define binary decision variables to capture the task offloading and UAV assignment in the system. The variable α_{nm} indicates whether the *n-th* IoT device offloads its task to the *m-th* UAV.

$$\alpha_{nm} = \begin{cases} 1 & \text{if } n\text{-}th \text{ IoT device offloads to } m\text{-}th \text{ UAV,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

The binary variable β_n indicates whether the *n-th* IoT device is selected for offloading. Each UAV can be assigned to at most ϑ IoT devices simultaneously to respect its connection capacity.

$$\beta_n = \begin{cases} 1 & \text{if } n\text{-}th \text{ IoT device is selected for operation,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Fig. 3.2 illustrates the end-to-end operation of the proposed trust-driven aerial MEC system. The process begins with the initialization of the target coverage area and the deployment of IoT devices, characterized by predefined task requirements and trust thresholds. UAVs are then initialized with their operational states, including position, computational capacity, and energy resources, and deployed to provide baseline coverage of the service region.

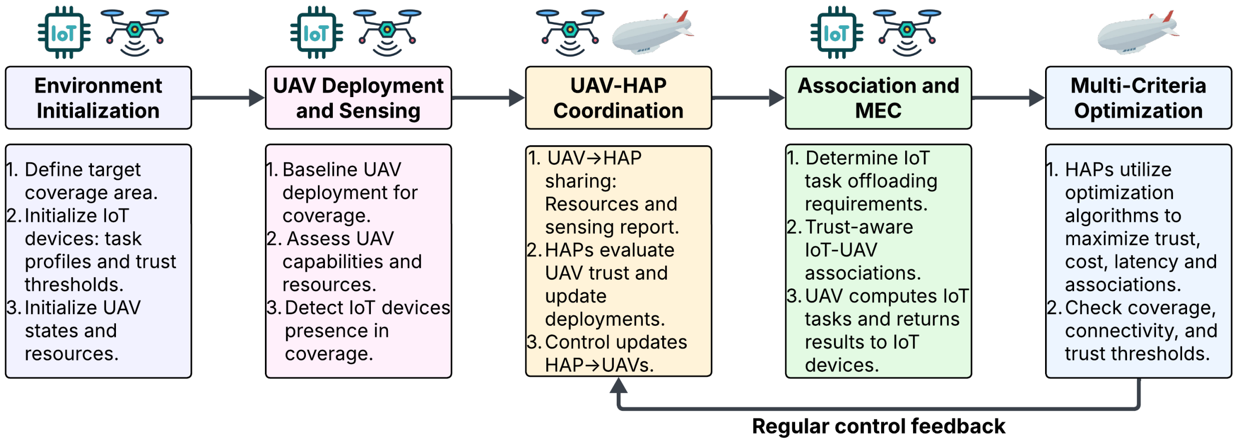


Figure 3.2: Trust-aware aerial MEC framework: UAV deployment, HAP coordination, IoT association, and multi-criteria optimization.

The sensing information collected by UAVs is forwarded to HAPs for trust evaluation and refinement of UAV deployment. This enables HAPs to adjust UAV positions to improve coverage and support trust-aware service delivery. IoT devices subsequently offload computational tasks to UAVs, which execute them under their resource constraints and return the processed results. Multi-criteria optimization continuously balances trust, latency, cost, and connectivity, while periodic UAV-HAP feedback enables adaptive and efficient service delivery across the network.

3.2.1 Communication Model

In the proposed framework, each IoT device establishes an uplink with its serving UAV to offload computational tasks. The wireless channel is assumed to be dominated by line-of-sight propagation and is determined by the relative placement of devices and UAVs as well as environmental factors [76]. The distance between the *n-th* IoT device and the *m-th* UAV is defined as $d_{nm} = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2 + z_m^2}$. The channel characteristics such as path loss, channel gain, bandwidth allocation, and noise power follow well-established models [76–79]. The achievable data rate is expressed as $R_{nm} = B_{nm} \log_2(1 + \gamma_{nm})$, where B_{nm} denotes the allocated bandwidth and γ_{nm} is the signal-to-noise ratio between the *n-th* IoT device and *m-th* UAV.

The communication model treats the IoT-UAV link as the primary access network that carries task data from distributed devices to their serving *m-th* UAV. The UAV-HAP link handles supervisory updates, UAV telemetry, and control messages and therefore forms a low-volume backhaul with modest power needs. Coordination between the *m-th* UAV and *k-th* HAP introduces a computational overhead (D_{mk}) and a coordination latency (L_{mk}^{CC}). These overheads recur periodically or when operational thresholds are exceeded. The IoT-UAV link therefore dominates end-to-end performance, and UAV-HAP synchronization sustains coordination across the two tiers.

The information reaching the *k-th* HAP reflects aggregate network conditions rather than the exact instantaneous state of each IoT device. UAV positioning is based on the spatial distribution of IoT devices established during initial deployment. This distribution changes gradually enough that refinement occurs at intervals rather than continuously. Such gradual change removes the need for constant or precise tracking of individual devices. The ongoing coordination instead relies on summaries the *m-th* UAV consolidates from the energy, computing, and security conditions of its associated IoT devices, observed through their regular interactions. The optimization works with these consolidated summaries rather than with each device's individual state. A small inaccuracy or a brief delay in any single update therefore has little effect on the resulting decisions.

3.2.2 UAV Trustworthiness Model

We define trust as a system-level metric that represents confidence in a UAV's ability to deliver reliable performance and support informed decision-making in aerial edge computing [11]. IoT devices may require different levels of trust depending on the criticality of the services they run. HAPs assess each UAV without bias along three dimensions: (i) reputation score, (ii) stability score, and (iii) security assurance score. Normalization maps every component metric to $[0, 1]$, where higher values denote greater trustworthiness, and each dimension score returns to the same range before entering the trust computation.

Reputation Score

This score tracks the consistency of a UAV in completing its assigned tasks over the operating history.

• *Task Completion (T_{nm}^{SUC})*: The *m-th* UAV's historical success rate on tasks from the *n-th* IoT device quantifies reliable execution across past assignments [12].

$$T_{nm}^{\mathrm{SUC}} = \frac{F_{nm}^{\mathrm{SUC}}}{F_{nm}}, \tag{3.3}$$

where F_{nm}^{SUC} and F_{nm} denote the number of successfully completed and total tasks assigned by the *n-th* IoT device to the *m-th* UAV, respectively.

• *Computational Resource Availability (T_{nm}^{AVA})*: The computational margin at the *m-th* UAV indicates its ability to absorb the demand from the *n-th* IoT device [51].

$$T_{nm}^{\mathrm{AVA}} = 1 - \frac{A_{nm}}{A_m^{\mathrm{MAX}}}, \tag{3.4}$$

where A_{nm} is the computational resource demand assigned from the *n-th* IoT device to the *m-th* UAV, and A_m^{MAX} denotes the maximum processing capacity of the *m-th* UAV.

The reputation score of the *m-th* UAV with respect to the *n-th* IoT device follows as:

$$T_{nm}^{\mathrm{REP}} = T_{nm}^{\mathrm{SUC}} + T_{nm}^{\mathrm{AVA}}. \tag{3.5}$$

Stability Score

This score reflects a UAV's capacity to serve its associated IoT devices without interruption.

• *Communication Link* (T_{nm}^{COM}): Packet loss on the link between the *n-th* IoT device and the

m-*th* UAV reflects communication reliability [80].

$$T_{nm}^{\mathrm{COM}} = 1 - \frac{\lambda_{nm}}{\lambda_m^{\mathrm{MAX}}}, \tag{3.6}$$

where λ_{nm} denotes the packet loss rate between the n-*th* IoT device and the m-*th* UAV, and λ_m^{MAX} represents the maximum tolerable packet loss rate for the m-*th* UAV.

• *Operational Environment* (T_{nm}^{ENV}): The environmental difficulty ϱ_{nm} measures conditions that reduce the m-*th* UAV's ability to sustain service to the n-*th* IoT device [81].

$$T_{nm}^{\mathrm{ENV}} = 1 - \frac{\varrho_{nm}}{\varrho^{\mathrm{MAX}} + \varepsilon}, \tag{3.7}$$

where ϱ_{nm} denotes the measured environmental difficulty encountered by the m-*th* UAV while serving the n-*th* IoT device, ϱ^{MAX} is the maximum observed difficulty across all IoT-UAV pairs in the region, and $\varepsilon > 0$ is a small constant included for numerical stability.

• *Energy Sufficiency (T_{nm}^{ENE})*: The energy demand of a task from the n-*th* IoT device sets how much of the m-*th* UAV's capacity that task consumes [41].

$$T_{nm}^{\mathrm{ENE}} = 1 - \frac{E_{nm}}{E_m^{\mathrm{MAX}}}, \tag{3.8}$$

where E_{nm} is the energy required by the m-*th* UAV to complete the task for the n-*th* IoT device, and E_m^{MAX} is the UAV's maximum energy capacity.

The stability score for the m-*th* UAV serving the n-*th* IoT device is given by:

$$T_{nm}^{\mathrm{STA}} = T_{nm}^{\mathrm{COM}} + T_{nm}^{\mathrm{ENV}} + T_{nm}^{\mathrm{ENE}}. \tag{3.9}$$

Security Score

This score captures a UAV's capability to detect and manage security threats.

• *Threat Detection (T_{nm}^{SD})*: The proportion of security threats detected by the m-*th* UAV while serving the *n-th* IoT device quantifies detection effectiveness. Security threats include adversarial actions such as spoofing, jamming, and malicious data injection. Threat detection metrics of this form have been studied in [13]. Total threats follow as $\phi_{nm}^{\mathrm{TOT}} = \phi_{nm}^{\mathrm{DET}}/r_m$, where $r_m \in (0,1]$ denotes the estimated threat detection probability of the *m-th* UAV. Historical performance or controlled validation tests supply r_m.

$$T_{nm}^{\mathrm{SD}} = \frac{\phi_{nm}^{\mathrm{DET}}}{\phi_{nm}^{\mathrm{TOT}} + \varepsilon}, \tag{3.10}$$

where ϕ_{nm}^{DET} and ϕ_{nm}^{TOT} denote the number of detected and total security threats associated with the *n-th* IoT device and *m-th* UAV, respectively, and ε is a positive constant introduced for numerical stability.

- *Undetected Threats Penalty (T_{nm}^{SP})*: Undetected security threats at the *m-th* UAV serving the *n-th* IoT device carry a quantified impact [7]. Their number follows as $\phi_{nm}^{\mathrm{UNDET}} = \phi_{nm}^{\mathrm{TOT}} - \phi_{nm}^{\mathrm{DET}}$. The penalty complements threat detection and captures the risk from missed threats.

$$T_{nm}^{\mathrm{SP}} = e^{-\omega_{nm}\left(\frac{\phi_{nm}^{\mathrm{UNDET}}}{\phi_{nm}^{\mathrm{TOT}} + \varepsilon}\right)}, \tag{3.11}$$

where $\phi_{nm}^{\mathrm{UNDET}}$ and ϕ_{nm}^{TOT} denote the undetected and total threats generated by the *n-th* IoT device and observed at the *m-th* UAV, ω_{nm} represents the severity of undetected threats, and ε maintains numerical stability.

Hence, the overall security score summarizes the *m-th* UAV's ability to detect threats and account for undetected risks for the *n-th* IoT device. The multiplicative formulation means that the score decreases whenever either detection performance or threat mitigation is insufficient, providing a comprehensive measure of UAV security trustworthiness.

$$T_{nm}^{\mathrm{SEC}} = T_{nm}^{\mathrm{SD}} T_{nm}^{\mathrm{SP}}. \tag{3.12}$$

The overall trust of the *m-th* UAV for the *n-th* IoT device is expressed as a weighted sum of its constituent scores.

$$T_{nm} = w_1 T_{nm}^{\text{REP}} + w_2 T_{nm}^{\text{STA}} + w_3 T_{nm}^{\text{SEC}}, \tag{3.13}$$

where w_1, w_2, and w_3 represent the relative importance of reputation, stability, and security, respectively, and satisfy $\sum_{i=1}^{3} w_i = 1$. The proposed trust model's flexibility allows different weight allocations based on mission requirements. Table 3.1 illustrates the influence of different trust weight configurations on UAV trust assessment in aerial MEC for 6G networks. Specific aspects of trust correspond to mission objectives such as coverage extension, precision computing, remote digital farming, and critical surveillance.

3.2.3 Cost Model

The cost model covers two components for the aerial networking framework. UAV deployment and trust-related expenses form these components, where the latter accounts for resource utilization and reliability requirements.

• *Cost of UAV Deployment* (C_m^{DEP}): Fixed and variable costs attach to the *m-th* UAV on deployment in the aerial network. Hardware characteristics, expected operational duration, and mission requirements determine the deployment cost [82]. We model the operational cost rate as constant over the mission duration, which integrates directly into the optimization formulation.

$$C_m^{\text{DEP}} = C_m^{\text{HW}} + C_m^{\text{OP}} \cdot t_m, \tag{3.14}$$

where C_m^{HW} is the fixed hardware cost, C_m^{OP} is the operational cost rate per unit time, and t_m is the expected operational time of the *m-th* UAV.

• *Trust-Related Cost* (C_{nm}^{TR}): This component captures the cost incurred to satisfy the trust threshold T_n^{THR} required by the *n-th* IoT device [83]. The relationship between this threshold and cost is modeled using a sigmoid function, which provides a smooth and bounded transformation from trust requirements to system cost. It exhibits a gradual variation at lower

Table 3.1: Illustration of mapping UAV trust parameters emphasis to different aerial edge computing scenarios in 6G networks.

Mode	$\boldsymbol{w_1}$	$\boldsymbol{w_2}$	$\boldsymbol{w_3}$	**Use Case**
i	0.33	0.33	0.34	Coverage extension
ii	**0.50**	0.25	0.25	Precision tasks computing
iii	0.25	**0.50**	0.25	Remote digital farming
iv	0.25	0.25	**0.50**	Critical surveillance

trust levels and an increasingly steep response as the requirement approaches the IoT trust threshold. The authors of [84] employed a similar sigmoid-based strategy for trust modeling, in which human-machine mutual trust was quantified using interaction conflict derived from behavioral differences. Their formulation produced smooth, bounded evolution of trust under varying interaction conditions. We extend this principle to model cost escalation under higher trust requirements in aerial networks and adopt it as a modeling assumption due to the absence of empirical measurements characterizing trust-cost relationships.

$$C_{nm}^{\mathrm{TR}} = \frac{C_m^{\mathrm{MAX}}}{1 + e^{-k_s(T_{nm} - T_n^{\mathrm{THR}})}}, \tag{3.15}$$

where C_m^{MAX} is the maximum allowable cost for the *m-th* UAV, representing the upper limit on the cost associated with achieving trust. T_{nm} is the trust level achieved by the *m-th* UAV for the *n-th* IoT device, and k_s controls the steepness of the cost increase near the threshold T_n^{THR}. This threshold may remain unsatisfied even when the cost reaches C_m^{MAX}.

3.2.4 Latency Model

The latency model characterizes the delays in task processing within the aerial edge computing framework. It captures all delays affecting task completion time and serves as a key factor in system optimization.

- *Local Computation Latency (L_n^{LOC})*: This component represents the time required for the *n-th* IoT device to process its computational task locally.

$$L_n^{\text{LOC}} = \frac{D_n \cdot \delta_n}{f_n}, \tag{3.16}$$

where D_n is the data size, δ_n denotes the computational workload (in CPU cycles per bit), and f_n is the processing frequency of the *n-th* IoT device. This component is considered for modeling purposes, while the effectiveness of the MEC framework is predominantly influenced by edge and centralized coordination latencies.

• *Edge Computation Latency (L_{nm}^{EDGE})*: This component accounts for the total delay when the *n-th* IoT device offloads a task to the *m-th* UAV. The subscript variation from n in local latency to nm in edge computation indicates that the latency depends on the specific pairing of the *n-th* IoT device and the *m-th* UAV. It consists of task transmission to the UAV, UAV-side computation, transmission of the processed result to the IoT device, and round-trip propagation delay. The channel is assumed symmetric, with identical uplink and downlink transmission rates (R_{nm}).

$$L_{nm}^{\text{EDGE}} = \frac{D_{nm}}{R_{nm}} + \frac{D_{nm}\delta_{nm}}{f_m} + \frac{S_{nm}}{R_{nm}} + \frac{2d_{nm}}{c}, \tag{3.17}$$

where D_{nm} represents the data size offloaded by the *n-th* IoT device to the *m-th* UAV, δ_{nm} represents the computational workload in CPU cycles per bit, f_m represents the UAV's processing frequency, and S_{nm} represents the size of the computation result. d_{nm} denotes the distance between the *n-th* IoT device and the *m-th* UAV, and c denotes the speed of light. The term $2d_{nm}/c$ indicates the round-trip propagation delay for signal transmission.

• *Centralized Coordination Latency (L_{mk}^{CC})*: This component represents the delay associated with UAV-HAP coordination, where the *m-th* UAV transmits status and sensing data to the *k-th* HAP for centralized optimization and receives updated control parameters in return. The uplink and downlink transmission rates (R_{mk}) are assumed identical under a symmetric channel condition.

$$L_{mk}^{\mathrm{CC}} = \frac{D_{mk}}{R_{mk}} + \frac{S_{mk}^{\mathrm{CC}}}{R_{mk}} + \frac{2d_{mk}}{c}, \tag{3.18}$$

where D_{mk} is the uplink coordination data size, S_{mk}^{CC} is the size of the control update sent back to the UAV, and d_{mk} is the distance between the *m-th* UAV and *k-th* HAP.

The overall task completion time combines local, edge, and occasional coordination delays, depending on the offloading decision. A binary indicator η_{mk} is defined, taking the value 1 when the *m-th* UAV undergoes a coordination update with the *k-th* HAP, and 0 otherwise.

$$L_{nm}^{\mathrm{TASK}} = (1 - \alpha_{nm})L_n^{\mathrm{LOC}} + \alpha_{nm}L_{nm}^{\mathrm{EDGE}} + \eta_{mk}L_{mk}^{\mathrm{CC}}. \tag{3.19}$$

3.2.5 Objective Function

We formulate a multi-criteria optimization problem that maximizes UAV trustworthiness and IoT device association while minimizing deployment cost, trust-related cost, and task completion latency. The objective is formulated as a weighted utility function defined as:

OP1:

$$\max_{\alpha_{nm},\beta_n,O_m} \sum_{n=1}^{N}\sum_{m=1}^{M} \alpha_{nm}\left(\nu^{\mathrm{TRS}}T_{nm} - \nu^{\mathrm{DEP}}C_m^{\mathrm{DEP}} - \nu^{\mathrm{TRC}}C_{nm}^{\mathrm{TR}} - \nu^{\mathrm{LAT}}L_{nm}^{\mathrm{TASK}} + \nu^{\mathrm{CON}}\beta_n\right) \tag{3.20}$$

Subject to:

$$C1 : \alpha_{nm},\ \beta_n \in \{0,1\}, \quad \forall\, n, m$$

$$C2 : \sum_{m=1}^{M} \alpha_{nm} = \beta_n, \quad \forall\, n$$

$$C3 : \sum_{n=1}^{N} \alpha_{nm} \leq \vartheta, \quad \forall\, m$$

$$C4 : \alpha_{nm}L_{nm}^{\mathrm{TASK}} \leq L_{nm}^{\mathrm{MAX}}, \quad \forall\, n, m$$

$$C5 : \sum_{n=1}^{N} \alpha_{nm}A_{nm} \leq A_m^{\mathrm{MAX}}, \forall\, m$$

$$C6: \sum_{n=1}^{N} \alpha_{nm} E_{nm} \leq E_m^{\text{MAX}}, \forall\ m$$

$$C7: T_{nm} \geq T_n^{\text{THR}} - \mathbb{M}\left(1-\alpha_{nm}\right), \forall\ n, m$$

where $\nu^{\text{TRS}}, \nu^{\text{DEP}}, \nu^{\text{TRC}}, \nu^{\text{LAT}}, \nu^{\text{CON}}$ are weights used to balance the importance of each objective. **C1** specifies the range of binary decision variables related to task offloading decision (α_{nm}) and IoT device selection (β_n). **C2** ensures that if the *n-th* IoT device is selected ($\beta_n = 1$), it is assigned to the *m-th* UAV. **C3** imposes an upper bound of ϑ on the number of IoT devices that can be connected to each UAV. **C4** ensures that the task latency L_{nm}^{TASK} for each IoT device does not exceed the maximum allowable value (L_{nm}^{MAX}). **C5** and **C6** collectively enforce the resource constraints of each UAV, limiting computational load and energy consumption to A_m^{MAX} and E_m^{MAX}, respectively. **C7** ensures that the trust threshold is enforced conditionally using the Big-M formulation: when offloading occurs ($\alpha_{nm} = 1$), the constraint $T_{nm} \geq T_n^{\text{THR}}$ is activated, and it is deactivated when offloading does not occur.

3.3 Proposed Solution

The optimization problem in (3.20) involves interdependent multi-criteria objectives in which UAV placement influences overall system performance. The proposed framework therefore adopts a two-stage decomposition under the coordination of a central HAP controller. A spatial clustering-based approach determines UAV deployment from the geographic distribution of IoT devices. Coverage extends beyond individual clusters through overlapping regions that permit service to neighboring clusters. The second stage performs IoT-UAV association and MEC resource allocation under trust-aware constraints. Heterogeneous trust levels across randomly distributed devices shape both association feasibility and resource allocation. The decomposition improves tractability while the two stages stay coupled, since UAV placement bounds both association feasibility and the trust-aware allocation constraints.

3.3.1 Stage 1 - UAV Placement Optimization

The coverage area partitions into M clusters according to the spatial distribution of the N IoT devices. Each cluster centroid fixes a UAV position in Stage 1 and captures the local device concentration. K-means clusters the IoT devices because it produces centroids that serve directly as UAV hovering points and minimize intra-cluster distances [85]. Computational efficiency and scalability suit dense IoT deployments. Hierarchical and density-based methods carry higher computational complexity and produce irregular cluster shapes, which suits them poorly to centroid determination [86]. Intra-cluster distance minimization formulates the clustering process, as given below:

$$\min_{\{O_m\}_{m=1}^{M}} \sum_{n=1}^{N} \min_{m} \|O_n - O_m\|^2 . \tag{3.21}$$

Algorithm 1 assigns devices to the nearest UAV and updates each UAV position to the centroid of its assigned devices. Iteration continues until convergence and produces the final UAV coordinates, $\{O_m^*\}_{m=1}^{M}$. These coordinates supply the spatial basis for the multi-criteria optimization in Stage 2.

3.3.2 Stage 2 - Trust-Aware IoT Association and Optimization

In this stage, the HAPs centrally perform a mixed-integer optimization that maximizes the weighted system utility, with UAV positions $\{O_m^*\}_{m=1}^{M}$ fixed from Stage 1. The framework jointly assigns IoT devices to UAVs and allocates MEC resources under trust-aware constraints, capturing the coupling between IoT-UAV association and resource allocation decisions. This results in a tightly interdependent optimization structure in which IoT-UAV association decisions directly influence the feasibility of resource allocation under trust constraints. The process is executed periodically or upon significant changes in device distribution, link quality, or UAV availability. The resulting formulation is solved using four complementary optimization algorithms, each offering different trade-offs between compu-

Algorithm 1 Optimized UAV Placements Using IoT Clusters

Require: Number of IoT devices N, Number of UAVs M,
IoT coordinates $\{O_n = (x_n, y_n, z_n)\}_{n=1}^{N}$
Ensure: Optimized UAVs 3D coordinates
$\{O_m^* = (x_m, y_m, z_m)\}_{m=1}^{M}$

Initialize: $\{O_m\}_{m=1}^{M} \subset \{O_n\}_{n=1}^{N}$ selected randomly
repeat
  Assignment Step:
  for $n = 1$ to N **do**
    Assign *n-th* IoT device to the closest UAV: $C(n) \leftarrow \arg\min_m \|O_n - O_m\|^2$
  end for
  Update Step:
  for $m = 1$ to M **do**
    $S_m \leftarrow \{n \mid C(n) = m\}$
    $O_m \leftarrow \frac{1}{|S_m|} \sum_{n \in S_m} O_n$
  end for
until UAV coordinates converge
Set $O_m^* \leftarrow O_m$ for all $m = 1, \ldots, M$
return $\{O_m^*\}_{m=1}^{M}$

tational complexity and solution quality.

OP2:

$$\max_{\alpha_{nm},\beta_n} \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_{nm} \big(\nu^{\mathrm{TRS}} T_{nm}(O_m^*) - \nu^{\mathrm{DEP}} C_m^{\mathrm{DEP}}(O_m^*) \ldots$$
$$-\nu^{\mathrm{TRC}} C_{nm}^{\mathrm{TR}}(O_m^*) - \nu^{\mathrm{LAT}} L_{nm}^{\mathrm{TASK}}(O_m^*) + \nu^{\mathrm{CON}} \beta_n\big) \tag{3.22}$$

Subject to: C2, C3, ..., C7

The multi-criteria optimization in (3.22) jointly determines the association variables α_{nm} and the device selection variables β_n under constraints C2–C7. OP1 in (3.20) is a binary optimization problem with $\alpha_{nm}, \beta_n \in \{0, 1\}$. OP2 relaxes these variables to $\alpha_{nm}, \beta_n \in [0, 1]$, resulting in a continuous formulation that remains non-convex due to variable coupling. The proposed PGO algorithm reformulates OP2 into OP2* by incorporating a penalty term into the objective function to promote binary-like solutions.

Benchmark Algorithm

This approach provides an optimal solution to OP2 under the coordination of the HAP using a branch-and-bound method. Its objective is to determine the binary variables α_{nm} and β_n that maximize system utility. The algorithm begins by relaxing the integer constraints and solving the resulting linear program (LP) to estimate the maximum utility [18]. It then explores IoT-UAV assignments and resource allocations, and discards those that violate constraints. The fully determined assignments are evaluated and used to update the best-known solution if the utility improves. Furthermore, the partially determined assignments are branched by fixing one undecided variable to 0 or 1 and generating new candidate solutions. The candidates that cannot exceed the current best utility are pruned to avoid unnecessary computation. This iterative process builds on the Stage 1 UAV positions O_m^* and explores all promising assignments to obtain a globally optimal solution. The detailed procedure is presented in Algorithm 2.

Greedy Algorithm

This approach provides a greedy solution to OP2 under the HAP's coordination. IoT devices are sequentially assigned to the UAV that maximizes the system utility defined in (3.22), based on the optimized UAV positions from (3.21). The algorithm evaluates all available UAVs for each IoT device to determine their utility values and selects the UAV that provides the highest utility [16]. The algorithm precomputes these values and stores them in the matrix $\mathbf{U} = [U_{nm}]$. A tentative assignment places the IoT device with the selected UAV, and confirmation follows only when constraints C2-C7 hold. The algorithm discards assignments that violate any constraint and leaves the IoT device unassigned. Confirmed assignments update the UAV resources before the algorithm proceeds to the next device. Each association decision depends only on the current device and the available UAV resources. The greedy strategy yields feasible IoT-UAV associations at low computational cost. Algorithm 3 presents the detailed procedure.

Algorithm 2 Benchmark Algorithm for OP2

Require: Number of IoT devices N, Number of UAVs M,
UAV coordinates $\{O_m^*\}_{m=1}^M$, Trust T_{nm}, Cost C_{nm}^{DEP} and C_{nm}^{TR}, Latency L_{nm}^{TASK}, Weights $\nu^{\text{TRS}}, \ldots, \nu^{\text{CON}}$,
Constraints ϑ, A_m^{MAX}, E_m^{MAX}, L_{nm}^{MAX}, T_n^{THR}

Ensure: Optimal α_{nm}^*, β_n^* for maximum weighted utility

Define: Objective function (OP2):

$$f(\alpha, \beta) = \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_{nm} \Big(\nu^{\text{TRS}} T_{nm}(O_m^*) - \nu^{\text{DEP}} C_m^{\text{DEP}}(O_m^*) - \nu^{\text{TRC}} C_{nm}^{\text{TR}}(O_m^*) - \nu^{\text{LAT}} L_{nm}^{\text{TASK}}(O_m^*) + \nu^{\text{CON}} \beta_n \Big)$$

Initialize: Create root node with LP-relaxation of OP2 (relax $\alpha_{nm}, \beta_n \in [0, 1]$), UAV resources
Set upper bound $UB \leftarrow \infty$, best solution $f^* \leftarrow -\infty$
Initialize queue $\mathcal{Q}$ with root node
while $\mathcal{Q}$ is not empty **do**
  Pop node N_{curr} from $\mathcal{Q}$
  Solve LP relaxation of OP2 at N_{curr}
  if solution violates any constraint (C2-C7) **then**
    Prune node
  end if
  if solution is integer ($\alpha_{nm}, \beta_n \in \{0, 1\}$) **then**
    Compute objective f_{curr}
    if $f_{curr} > f^*$ **then**
      Update best solution: $f^* \leftarrow f_{curr}$, save α^*, β^*
    end if
  else if objective $f_{relax} > f^*$ **then**
    Select a fractional variable α_{ij} or β_ℓ to branch on
    for value in $\{0, 1\}$ **do**
      Create child node with variable fixed
      Add child to $\mathcal{Q}$
    end for
  else
    Prune node
  end if
end while
return Optimal integer solution: α_{nm}^*, β_n^*

Baseline Relaxation Algorithm

This algorithm solves OP2 in its continuous form defined in (3.22). The solution builds on the interior-point optimization method [62]. The relaxed objective carries no penalty term,

Algorithm 3 Greedy Algorithm for OP2

Require: Num. of IoT devices N, num. of UAVs M, UAV coord. $\{O_m^*\}_{m=1}^M$, trust T_{nm}, cost C_{nm}^{DEP} and C_{nm}^{TR}, latency L_{nm}^{TASK}, weights $\nu^{\mathrm{TRS}}, \ldots, \nu^{\mathrm{CON}}$, utility matrix $\mathbf{U} = [U_{nm}]$, constraints ϑ, A_m^{MAX}, E_m^{MAX}, L_{nm}^{MAX}, T_n^{THR}

Ensure: Greedy association matrix α_{nm}^G

Initialize: UAV resources, $\alpha_{nm}^G \leftarrow 0$ for all n, m
for $n = 1$ TO N **do**
  MAX_UTIL $\leftarrow -\infty$
  BEST_UAV $\leftarrow -1$
  for $m = 1$ TO M **do**
    if $U_{nm} >$ MAX_UTIL **then**
      MAX_UTIL $\leftarrow U_{nm}$
      BEST_UAV $\leftarrow m$
    end if
  end for
  Tentatively assign: $\alpha_{n,\mathrm{BEST\_UAV}}^G \leftarrow 1$
  if All Constraints C2 TO C7 are satisfied **then**
    Update UAV resource usage counters
  else
    Revert Assignment: $\alpha_{n,\mathrm{BEST\_UAV}}^G \leftarrow 0$
  end if
end for
return α_{nm}^G

and the integrality gap therefore persists through optimization. Deterministic rounding maps the continuous solution to a feasible binary decision space. A repair step then restores satisfaction of constraints C2-C7. The fractional information lost in rounding limits the solution quality of the BRE algorithm.

Penalty-Guided Optimization Algorithm

The proposed PGO algorithm augments the continuous relaxation in (3.22) with a penalty term $\Gamma(\gamma, \alpha_{nm}, \beta_n)$. The penalty grows as α_{nm} and β_n move away from binary values, which steers the solver toward integral assignments [58]. We employ sequential quadratic programming (SQP), which handles the nonlinear penalized objective through a sequence of tractable subproblems. Quadratic subproblems with linearized constraints update the decision variables at each iteration. Feasibility tightens as the solution refines within the continuous domain. The penalty term is defined as:

$$\Gamma(\gamma, \alpha_{nm}, \beta_n) = -\gamma \sum_{n=1}^{N} \left(\beta_n (1-\beta_n) + \sum_{m=1}^{M} \alpha_{nm} (1-\alpha_{nm}) \right), \tag{3.23}$$

where $\gamma > 0$ is a coefficient that controls the influence of the penalty term on the objective function. The penalty $\Gamma(\gamma, \alpha_{nm}, \beta_n)$ vanishes at binary values of α_{nm} and β_n. Fractional values drive it negative and reduce the overall objective.

OP2*:

$$\begin{aligned} \mathcal{L} = \max_{\alpha_{nm}, \beta_n} & \sum_{n=1}^{N} \sum_{m=1}^{M} \alpha_{nm} \left(\nu^{\text{TRS}} T_{nm}(O_m^*) - \nu^{\text{DEP}} C_m^{\text{DEP}}(O_m^*) \ldots \right. \\ & \left. -\nu^{\text{TRC}} C_{nm}^{\text{TR}}(O_m^*) - \nu^{\text{LAT}} L_{nm}^{\text{TASK}}(O_m^*) + \nu^{\text{CON}} \beta_n \right) + \Gamma(\gamma, \beta_n, \alpha_{nm}) \end{aligned} \tag{3.24}$$

Subject to: C2, C3, ..., C7

OP2* in (3.24) recasts the original mixed-integer problem as a penalized continuous optimization. Relaxation-based approaches instead solve OP2 in continuous form and map the result to feasible discrete values through post-processing [62]. Alternating direction method of multipliers (ADMM) methods decompose the problem and solve the subproblems iteratively [59]. The proposed PGO algorithm is well-suited to the HAP-controlled environment, as it preserves a centralized optimization structure while promoting binary-like solutions within the optimization process under coupled system constraints. This reformulation enables efficient centralized gradient-based optimization.

In this context, OP2* is solved using SQP, starting from initial estimates of α_{nm} and β_n, with penalty parameter γ and convergence threshold ϵ [87]. SQP iteratively constructs a quadratic approximation of the penalized objective function $\mathcal{L}$ and solves the resulting subproblem using a linearized approximation of the system constraints. The decision variables are updated at each iteration and enforced within the interval $[0, 1]$ to satisfy the variable bounds of OP2*. The iterations proceed until the norm of the gradient of $\mathcal{L}$ falls below ϵ.

Finally, the obtained near-binary values of α_{nm} and β_n are converted into feasible binary assignments through a post-processing step that enforces all system constraints. The detailed procedure is presented in Algorithm 4.

Algorithm 4 PGO Algorithm for OP2

Require: Number of IoT devices N, Number of UAVs M,
UAV coordinates $\{O_m^*\}_{m=1}^{M}$, Trust T_{nm}, Cost C_{nm}^{DEP} and C_{nm}^{TR}, Latency L_{nm}^{TASK}, Weights $\nu^{\mathrm{TRS}}, \ldots, \nu^{\mathrm{CON}}$,
Constraints ϑ, A_m^{MAX}, E_m^{MAX}, L_{nm}^{MAX}, T_n^{THR}

Ensure: Near-optimal α_{nm}^*, β_n^* for maximum weighted utility

Define: Objective function (OP2*)

Input: Initial guess $\alpha_{nm}^{(0)}$, $\beta_n^{(0)}$, Penalty parameter γ, Convergence threshold ϵ

$j \leftarrow 0$

repeat

Compute gradient $\nabla\mathcal{L}^{(j)}$ and Hessian approximation $H^{(j)}$ of:

$$\mathcal{L} = \sum_{n=1}^{N}\sum_{m=1}^{M}\alpha_{nm}\Big(\nu^{\mathrm{TRS}}T_{nm}(O_m^*) - \nu^{\mathrm{DEP}}C_m^{\mathrm{DEP}}(O_m^*)$$

$$-\nu^{\mathrm{TRC}}C_{nm}^{\mathrm{TR}}(O_m^*) - \nu^{\mathrm{LAT}}L_{nm}^{\mathrm{TASK}}(O_m^*) + \nu^{\mathrm{CON}}\beta_n\Big) + \Gamma(\gamma, \beta_n, \alpha_{nm})$$

Solve QP subproblem:

$$\min_{\Delta\mathbf{x}} \quad \frac{1}{2}\Delta\mathbf{x}^{\mathbf{T}}H^{(j)}\Delta\mathbf{x} + \nabla\mathcal{L}^{(j)\mathbf{T}}\Delta\mathbf{x}$$

Subject to: Linearized constraints C2-C7 at j^{th} iteration

Update variables:

$$\alpha_{nm}^{(j+1)} = \alpha_{nm}^{(j)} + \Delta\alpha_{nm}$$
$$\beta_n^{(j+1)} = \beta_n^{(j)} + \Delta\beta_n$$

Project $\alpha_{nm}^{(j+1)}, \beta_n^{(j+1)}$ to $[0, 1]$ if needed

$j \leftarrow j + 1$

until $\|\nabla\mathcal{L}^{(j)}\| < \epsilon$

return Near-optimal decision variables α_{nm}^*, β_n^*

3.3.3 Computational Complexity

The computational complexity of the proposed framework is analyzed in two stages. In Stage 1, UAV placements are optimized using Algorithm 1 with a complexity of $O(UNM)$, where U denotes the number of iterations required for convergence. In Stage 2, four algorithms are applied to the OP2 problem with varying computational demands. The benchmark algorithm evaluates all possible binary assignments, resulting in a worst-case complexity of $O(2^{NM})$, which makes it infeasible for large-scale scenarios [18]. The greedy algorithm assigns each IoT device to the UAV that maximizes system utility, and it has worst-case complexity $O(NM)$ [16]. Linear scaling suits resource-constrained environments. The BRE algorithm runs at $O((N+M)^3)$ [62] and serves as the baseline for large-scale scenarios. The PGO algorithm carries an approximate complexity of $O((N+M)^3 + I(N+M)^2)$, where I denotes the number of SQP iterations [87]. The polynomial costs of BRE and PGO contrast with the exponential growth of the exact benchmark.

3.4 Performance Analysis

Simulations across a range of network scenarios assess the proposed framework. HAPs supply global supervision and centralized optimization of UAVs and IoT devices under varying trustworthiness thresholds.

3.4.1 Simulation Setup

The simulation settings for IoT-UAV interactions follow [18], while HAP placement and configuration follow the approach in [55]. We assume a HAP computational capacity of 50 GHz, which supports centralized optimization for the underlying IoT-UAV network. The aerial edge computing framework considers up to $N=300$ IoT devices and $M=8$ UAVs under the coordination of a single HAP. UAVs operate at altitudes between $z_m=30$ and 50 m and cover a radius of approximately 250 m each, while the HAP is positioned at $z_k=20$ km. IoT device data sizes are set within $D_{nm} = 10$ to 20 Mb, and the UAV-HAP coordination overhead is $D_{mk} = 0.5$ Mb. UAV trust is evaluated through reputation (T_{nm}^{REP}), stability (T_{nm}^{STA}), and security (T_{nm}^{SEC}) metrics, each ranging within $[0, 1]$ and weighted by $w_1, w_2, w_3 \in [0, 0.5]$. These are further integrated with utility weights $\nu^{\mathrm{TRS}}, \ldots, \nu^{\mathrm{CON}} \in [0.1, 0.6]$.

Additional simulation parameters include IoT-UAV trust $T_{nm} \in [0, 1]$, UAV deployment cost $C_m^{\mathrm{DEP}} = 0.5$, and initial trust-related cost $C_{nm}^{\mathrm{TR}} = 0.25$. The logic for cost modeling is conceptually based on [82], which informed the formulation of deployment and trust-related costs in this work. Task latency is considered within $L_{nm}^{\mathrm{TASK}} = 1$ to 5 s, while the coordination latency between HAP and UAV, $L_{mk}^{\mathrm{CC}} = 1$ s, is obtained by neglecting the control data size (S_{mk}^{CC}) in (3.18). Each UAV supports a maximum of $\vartheta = 50$ IoT devices, with computation powers $A_n = 1$ GHz for IoT devices and $A_m = 7$ to 15 GHz for UAVs. The normalized IoT energy requirement is $E_{nm} = 0.1$, and the trust threshold is set as $T_n^{\mathrm{THR}} = 0.5$ to 0.95. The Big-M parameter is defined as $\mathbf{M} = 50$. The penalty coefficient is set to $\gamma = 2$, which controls the strength of the penalty term, enforcing integrality constraints. We observed degraded performance at $\gamma = 0$, where integrality enforcement weakens, and stable behavior across network configurations at $\gamma \geq 1$. Table 3.2 summarizes the simulation parameters.

3.4.2 Simulation Methodology

The simulation framework considers N IoT devices and M UAVs within the defined coverage region, with a HAP supervising execution. IoT devices are randomly positioned to approximate a uniform spatial distribution. UAVs operate within their designated altitude limits, and task sizes are sampled from the specified ranges. Trust components are drawn from a uniform distribution over $[0, 1]$ and aggregated through weighting into the overall trust scores. IoT-UAV associations are established by K-means clustering with one cluster per UAV, subject to trust thresholds and capacity limits. The optimization framework finalizes the assignments for task allocation and resource management. The performance of the proposed and comparative algorithms is evaluated in terms of the number of connected

Table 3.2: Simulation parameters details.

Variable Description	**Symbol**	**Value**	**Unit**
IoT devices	N	[20,300]	–
UAVs	M	[2,8]	–
UAVs altitude	z_m	[30,50]	m
UAVs coverage radius	d_m^{MAX}	250	m
HAPs	K	1	–
HAPs altitude	z_k	20	km
IoT data size	D_{nm}	[10,20]	Mb
UAV-HAPs overhead	D_{mk}	0.5	Mb
UAV Reputation trust	T_{nm}^{REP}	[0,1]	–
UAV Stability trust	T_{nm}^{STA}	[0,1]	–
UAV Security trust	T_{nm}^{SEC}	[0,1]	–
Trust weights	w_1, w_2, w_3	[0.25,0.5]	–
Utility weights	$\nu^{\mathrm{TRS}}, \ldots, \nu^{\mathrm{CON}}$	[0.1,0.6]	–
IoT-UAV trust	T_{nm}	[0,1]	–
UAV deployment cost	C_m^{DEP}	0.5	–
Initial trust-related cost	C_{nm}^{TR}	0.25	–
Task latency	L_{nm}^{TASK}	[1,5]	s
Coordination latency	L_{mk}^{CC}	1	s
Max. IoT per UAV	ϑ	50	–
IoT compute power	A_n	1	GHz
UAV compute power	A_m	[7,15]	GHz
IoT energy threshold	E_{nm}	0.1	–
IoT trust threshold	T_n^{THR}	[0.5,0.95]	–
Big-M value	$\mathbf{M}$	50	–
Iterations	$\mathcal{I}$	≤ 400	–
Penalty Coefficient	γ	2	–

IoT devices, utility under varying trust thresholds, and normalized latency, averaged over multiple runs. We fix the random seed to keep IoT device positions constant across runs, and perform all simulations in MATLAB (R2024b).

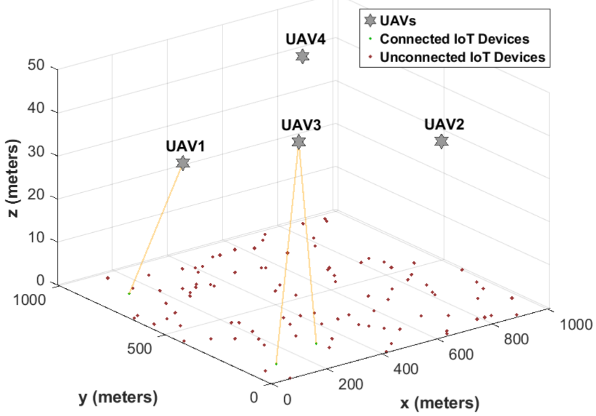
3.4.3 Performance Metrics

We evaluate the framework across connectivity, utility, robustness, and latency. The association ratio gives the proportion of IoT devices that obtain a serving UAV. Multi-criteria utility aggregates trust, deployment cost, and latency under the objective weights. The count of admitted devices under increasing trust thresholds measures robustness to stricter requirements. Normalized latency reports task execution and coordination latency per connected device. The number of IoT devices and UAVs governs cluster formation and workload distribution, which in turn affects connectivity and latency. Task data sizes, UAV computation capacities, and trust thresholds influence energy consumption and task completion times. These quantities jointly determine the efficiency of the proposed and comparative algorithms.

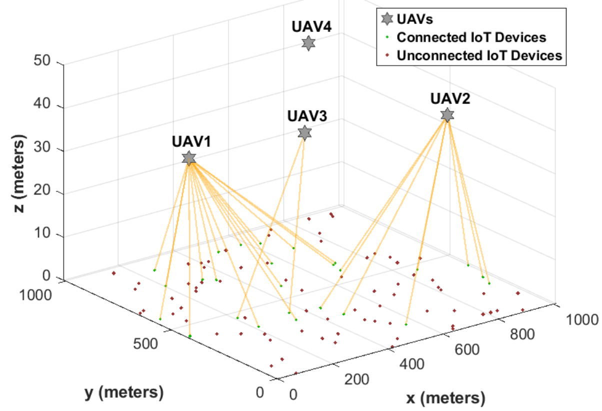
3.4.4 Simulation Results

Simulation outcomes assess the performance of the proposed framework. Fig. 3.3 presents the IoT-UAV association results for intelligently placed UAVs under the BRE, greedy, PGO, and benchmark algorithms. Equal weights in the objective function (3.20) (all $\nu = 0.2$) apply to each criterion for a balanced evaluation. Severe constraint violations limit the BRE algorithm to 3% association, while localized decision-making brings the greedy algorithm to 26%. The PGO algorithm reaches 75% association against the 79% of the benchmark. A penalty-guided heuristic balances multiple criteria in PGO and accounts for the difference. Branch-and-bound determines the optimal solution in the benchmark algorithm at substantially higher computational effort.

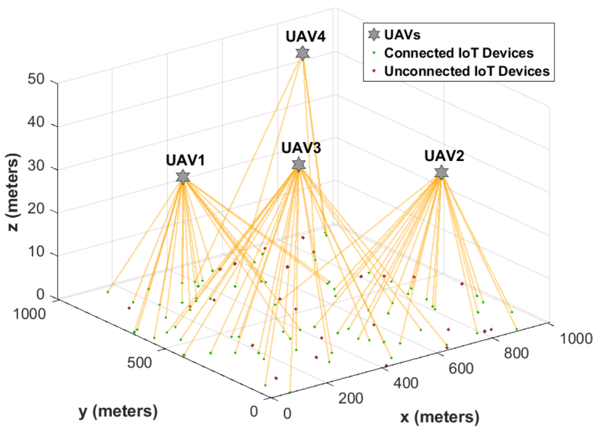
Beyond the equal-weight setting, unequal weight configurations in (3.20) allow the framework to emphasize specific criteria according to mission needs. Table 3.3 summarizes IoT-UAV association performance for two illustrative operational settings defined by distinct weight configurations. The trust emphasis scenario prioritizes trust-aware IoT-UAV interactions by assigning $\nu^{\mathrm{TRS}} = 0.6$ with all other weights $= 0.1$, while the association emphasis scenario prioritizes connectivity performance by assigning $\nu^{\mathrm{CON}} = 0.6$ with all other weights $= 0.1$. These configurations reflect different mission requirements and illustrate the flexibility available to system designers in adapting the proposed multi-criteria framework. The considered settings serve as representative cases to evaluate the robustness of the proposed



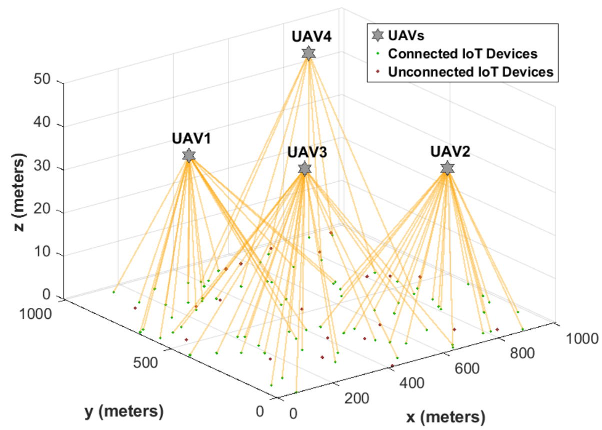
(a) BRE algorithm: 3%



(b) Greedy algorithm: 26%



(c) PGO algorithm: 75%



(d) Benchmark algorithm: 79%

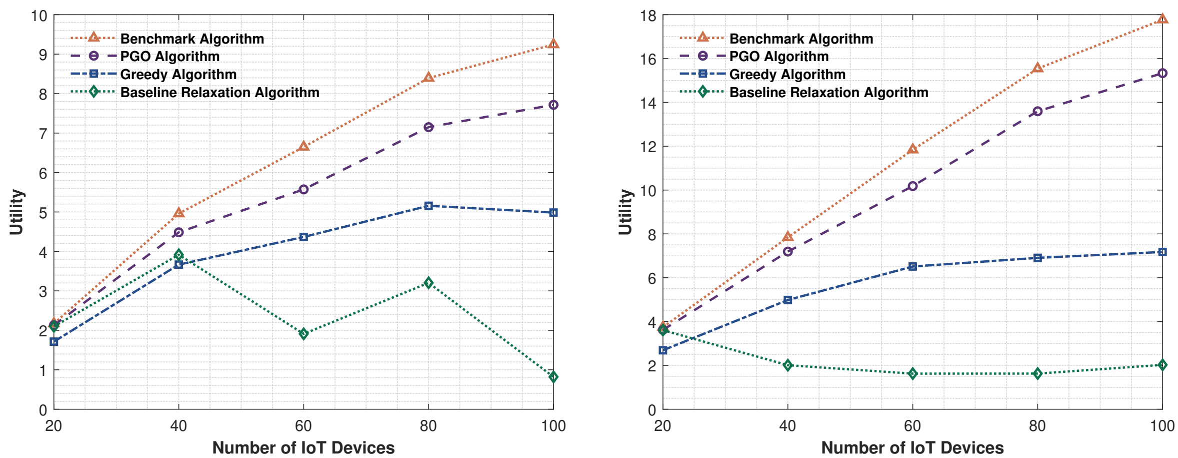
Figure 3.3: IoT-UAV associations for $N = 100$ IoT devices requiring task offloading are evaluated under the intelligent deployment of $M = 4$ UAVs, using different optimization algorithms and equally weighted multi-criteria utility parameters (all $\nu = 0.2$).

PGO algorithm under contrasting operational objectives. In the trust-emphasis case, the BRE and greedy algorithms achieve 14% and 46% IoT-UAV association, whereas the PGO algorithm achieves 77%, matching the benchmark performance. In the association-emphasis case, BRE and greedy algorithms achieve 14% and 26%, respectively, while the PGO algorithm achieves 61%, compared to 81% for the benchmark algorithm, demonstrating stable performance across varying mission configurations. These results confirm that PGO algorithm effectively balances multiple design objectives.

Fig. 3.4 illustrates the multi-criteria utility variation with the number of IoT devices

Table 3.3: Performance of optimization algorithms for N=100 IoT devices and M=4 UAVs, under trust and association emphasis.

Optimization	BRE	Greedy	PGO	Benchmark
Trust emphasis (ν^{TRS}=0.6, all other $\nu = 0.1$)	14%	46%	77%	77%
Association emphasis (ν^{CON}=0.6, all other $\nu = 0.1$)	14%	26%	61%	81%



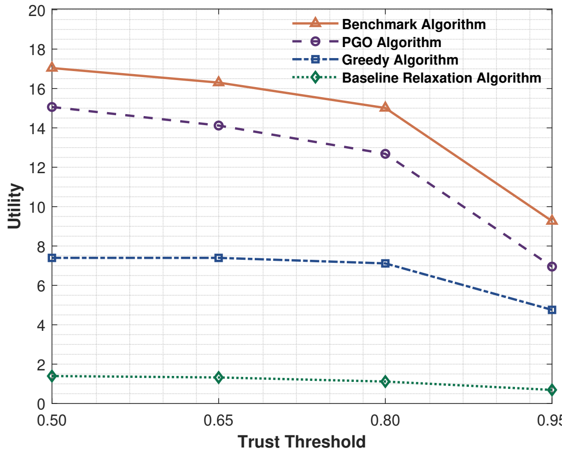
(a) M = 2, $T_n^{\mathrm{THR}} = 0.65$ (b) M = 5, $T_n^{\mathrm{THR}} = 0.65$

Figure 3.4: Utility versus IoT devices (N), under equally weighted parameters (all $\nu = 0.2$) for different number of serving UAVs.

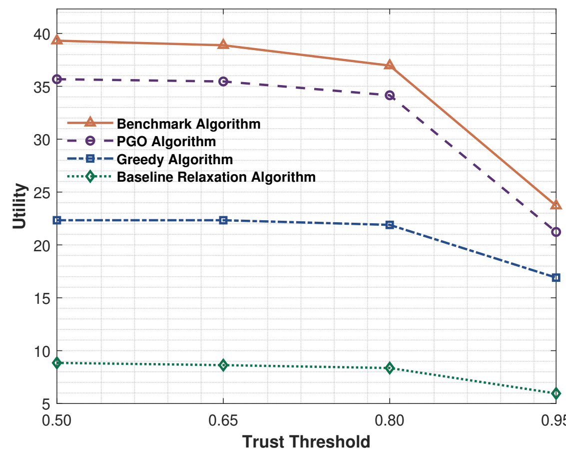
under two levels of resource availability, (a) $M = 2$ and (b) $M = 5$ UAVs, with a trust threshold $T_n^{\mathrm{THR}} = 0.65$. The PGO algorithm approximates the benchmark algorithm across the full range of network sizes and available resources. Fig. 3.4(a) and Fig. 3.4(b) show the greedy algorithm saturating at 40 IoT devices and beyond, where localized and sequential decisions reach their limit. In Fig. 3.4(a), the BRE algorithm outperforms greedy at $N = 20$ and 40, then fluctuates and falls below the other algorithms. Fig. 3.4(b) shows BRE ahead of greedy only at $N = 20$, with steadier but lowest performance as number of IoT devices increase.

Fig. 3.5 presents the multi-criteria utility outcomes for three weighting strategies with equal weighting in (a), trust emphasis in (b), and connectivity emphasis in (c), evaluated with respect to IoT device trust threshold variations. The benchmark algorithm consistently achieves the highest utility, while the proposed PGO algorithm maintains a close approximation with significantly reduced computational complexity. The BRE and greedy algorithms show inferior performance due to constraint violations and localized decision-making, respectively. The results of Fig. 3.5 (a) with equal weighting of multi-criteria parameters produce the lowest utilities, which shows that uniform treatment of parameters dilutes the effectiveness of optimization. Similarly, the results of Fig. 3.5 (c) with connectivity emphasis lead to the highest utilities across all algorithms, which highlights its decisive role in maximizing system-wide performance. Furthermore, the Fig. 3.5 (b) with trust emphasis configuration also achieves competitive utility while maintaining reliable associations. These results indicate that trust is a necessary condition for robust aerial edge computing to support diversified IoT services. These insights reinforce the need for effective designs to balance connectivity-driven utility gains with trust-centric assurances to achieve both high performance and trustworthiness in aerial edge networks.

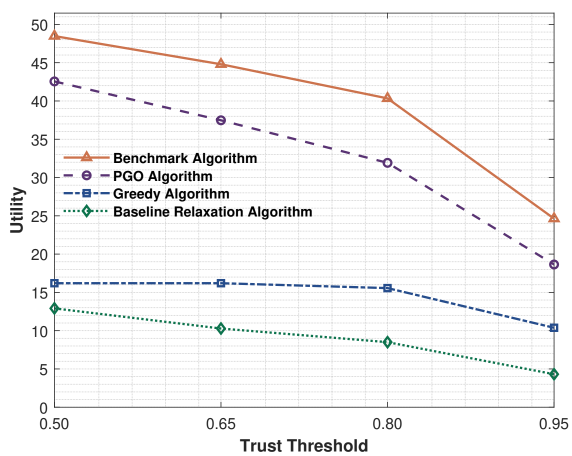
Fig. 3.6 presents the number of connected IoT devices under varying trust thresholds for the three previously considered weighting strategies: equal weighting (Fig. 3.3), trust and association emphasis (Table 3.3). The bar plots highlight the impact of stricter trust thresholds on overall connectivity. The benchmark algorithm consistently provides the optimal trend, with connectivity starting at 79%, 77%, and 80% for equal, trust, and connectivity emphasis weighting at $T_n^{\mathrm{THR}} = 0.5$, respectively. The connectivity offered by benchmark algorithm declines to 40% at $T_n^{\mathrm{THR}} = 0.95$ for equal, trust, and connectivity emphasis weighting. The proposed PGO algorithm closely follows the benchmark algorithm and achieves 64%, 66%, and 70% at $T_n^{\mathrm{THR}} = 0.5$, and 27%, 36%, and 30% at $T_n^{\mathrm{THR}} = 0.95$. Specifically, the PGO algorithm shows only a 4% degradation at $T_n^{\mathrm{THR}} = 0.95$ in the trust-emphasis case, as compared to the benchmark algorithm. Furthermore, the BRE and greedy algorithms consistently exhibit inferior performance in supporting IoT device connectivity under varying trust thresholds. Overall, the PGO algorithm delivers near-optimal connectivity with



Equally weighted parameters
(a) (all $\nu = 0.2$)



Trust emphasized
(b) ($\nu^{\mathrm{TRS}} = 0.6$, others $= 0.1$)

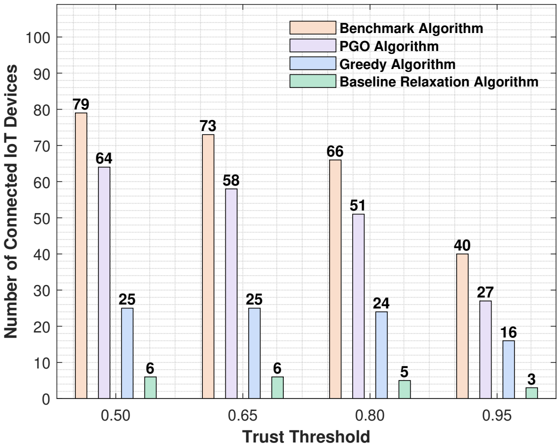


Connectivity emphasized
(c) ($\nu^{\mathrm{CON}} = 0.6$, others $= 0.1$)

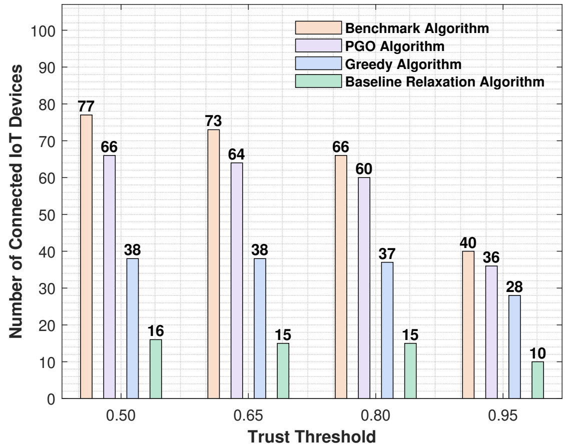
Figure 3.5: Utility performance under varying trust thresholds for $N = 100$ IoT devices requiring MEC services from $M = 4$ UAVs.

lower computational overhead, which highlights its robustness and suitability for trustworthy aerial edge computing.

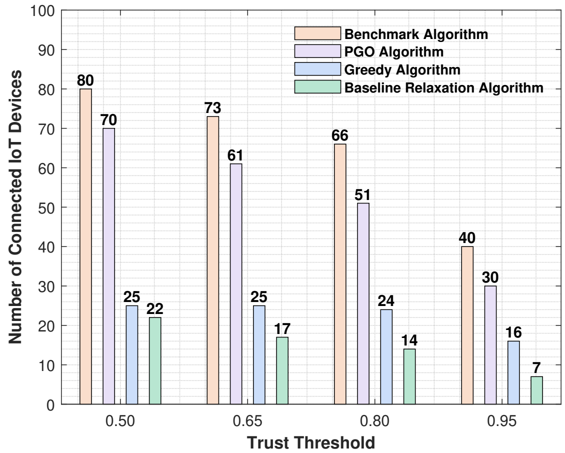
Fig. 3.7 presents the number of successfully connected IoT devices under varying network scales with equally weighted utility parameters. In Fig. 3.7 (a), connectivity is evaluated as the number of IoT devices increases from $N = 100$ to 300 under fixed UAV availability $M = 6$ and trust threshold ($T_n^{\mathrm{THR}} = 0.65$). The proposed PGO algorithm closely follows the benchmark and achieves approximately 90% of its performance at higher IoT densities. In contrast, both the greedy and BRE algorithms remain significantly inferior and exhibit



Equally weighted parameters
(a) (all $\nu = 0.2$)



Trust emphasized
(b) ($\nu^{\mathrm{TRS}} = 0.6$, others $= 0.1$)

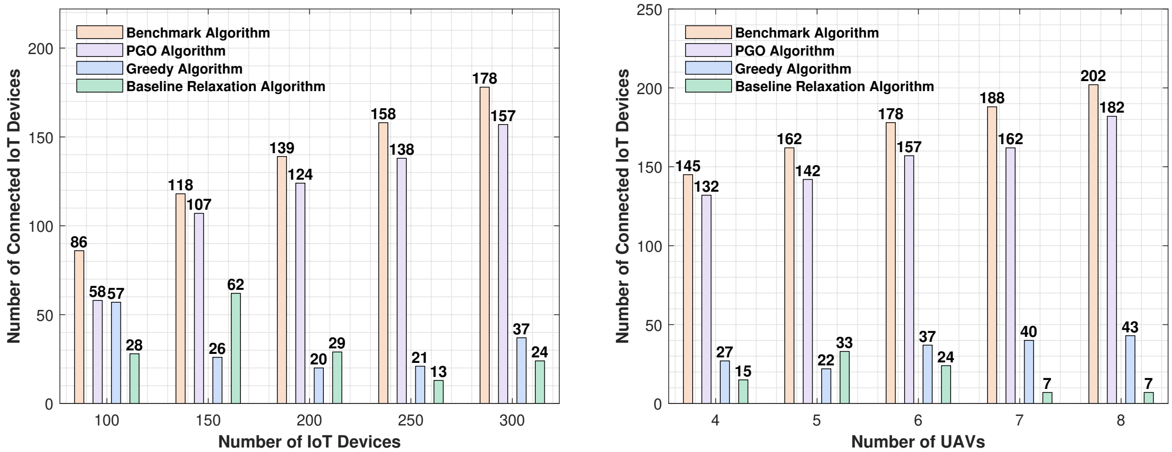


Connectivity emphasized
(c) ($\nu^{\mathrm{CON}} = 0.6$, others $= 0.1$)

Figure 3.6: Number of successfully connected IoT devices under varying trust thresholds for $N = 100$ IoT devices and $M = 4$ UAVs.

unstable behavior as the network scale increases. Furthermore, Fig. 3.7 (b) shows the impact of increasing UAV availability from $M = 4$ to 8 under fixed IoT load ($N = 300$), where the proposed PGO algorithm maintains approximately 86%–91% of the benchmark algorithm's performance across all configurations. The greedy algorithm improves only marginally, while the BRE algorithm remains inconsistent, exhibiting non-monotonic behavior. Overall, the results demonstrate that the proposed PGO algorithm achieves scalable and robust connectivity under both demand and resource-driven network scales.

Fig. 3.8 presents normalized latency as defined in the latency term of the multi-criteria

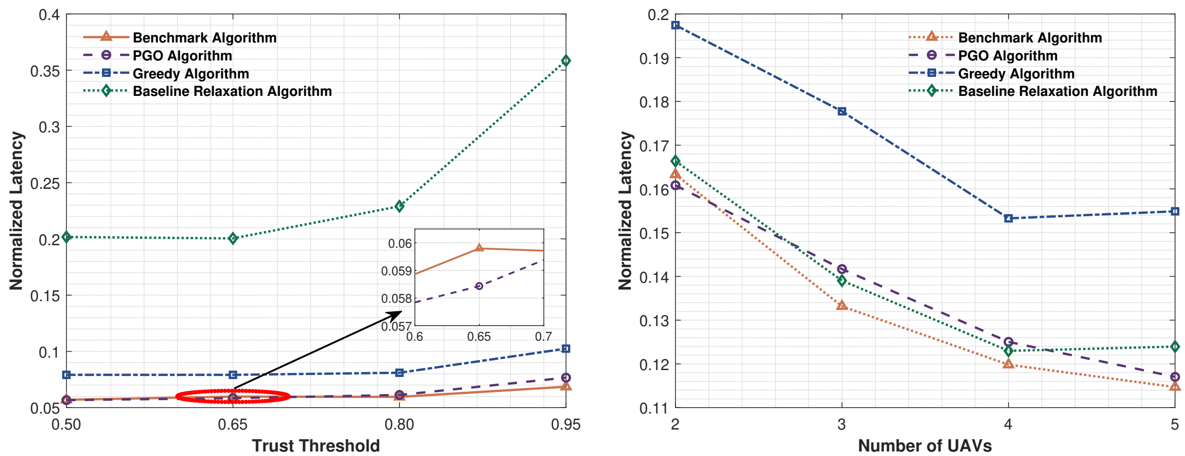


(a) M = 6, $T_n^{\mathrm{THR}} = 0.65$ (b) N = 300, $T_n^{\mathrm{THR}} = 0.65$

Figure 3.7: Number of successfully connected IoT devices, under equally weighted parameters (all $\nu = 0.2$) for IoT-UAV offloading.

utility function in (3.20). In Fig. 3.8 (a) with $N = 100$ IoT devices, the PGO algorithm maintains low latency for moderate thresholds ($T_n^{\mathrm{THR}} = 0.5$-0.75), while the benchmark algorithm achieves the minimum latency at $T_n^{\mathrm{THR}} = 0.95$. The BRE algorithm records the highest latency due to repeated constraint violations. The greedy algorithm improves on the BRE algorithm and trails the PGO and benchmark algorithms, since localized decisions address the multi-criteria objectives only partially. Fig. 3.8 (b) shows near-optimal PGO results across all UAV counts at $N = 20$ IoT devices. The BRE algorithm follows PGO closely and slightly exceeds it for $M = 3$ and 4, whereas the greedy algorithm records the highest latency in this scenario. PGO therefore balances latency across IoT densities and UAV deployments, while BRE and greedy performance varies with device count and resource availability.

The simulations sweep IoT device density, UAV availability, and trust thresholds. PGO sustains high utility and connectivity at low latency across the tested configurations. IoT-UAV clustering balances task allocation, and HAP coordination manages resources. Prior work supports the two-stage approach, with k-means applied to partitioning large UAV-assisted deployments [88], large-scale IoT scenarios examined in [89], and joint optimization frameworks employed in [53].



(a) N = 100, M = 4

(b) N = 20, $T_n^{\text{THR}} = 0.65$

Figure 3.8: Normalized latency versus trust threshold and UAVs, under equally weighted parameters (all $\nu = 0.2$) for IoT-UAV offloading.

3.5 Summary

This chapter presents a multi-criteria optimization framework for aerial edge computing in 6G networks. A trust metric spanning UAV reputation, reliability, and security enters the association decision directly. Evaluation across varying trust thresholds, IoT densities, and UAV-MEC configurations exposes how trust shapes association, utility, connectivity, and latency. The PGO algorithm achieves near-optimal results across all trust levels at low computational complexity. A multi-dimensional trust metric therefore governs which IoT-UAV associations form under tightening requirements, and PGO suits deployments where computational resources and trust requirements are jointly considered.

Chapter 4

Toward Trustworthy Aerial Networks: Modeling IoT Spatial and Residual Energy Uncertainties

4.1 Introduction

6G networks aim to extend reliable, low-latency service into remote and emergency settings beyond the reach of terrestrial deployment [1]. Aerial networks composed of UAVs and HAPs provide rapidly deployable coverage that complements terrestrial and satellite infrastructure [71]. Service from these platforms primarily depends on UAV energy, computing capacity, and link quality, each of which varies throughout a deployment. IoT devices contribute a further variation, since the spatial positions and residual energy they report may deviate from actual values [19]. Deviations of this kind arise from localization inaccuracy and from IoT device energy consumption that follows irregular patterns. An IoT-UAV association computed from reported values may therefore fail once service begins. Robust formulations address this by admitting associations that remain feasible across the range each reported value may take, alongside the trust and latency requirements every device declares.

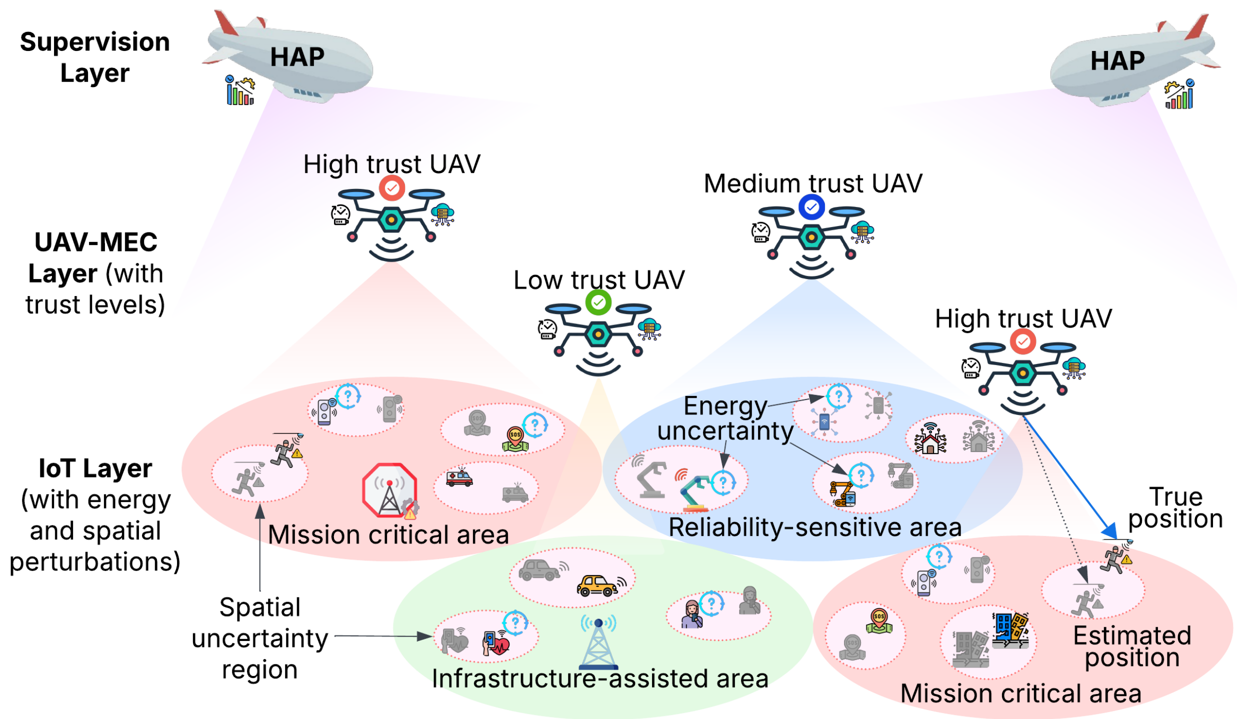


Figure 4.1: Trustworthy aerial edge computing architecture with IoT spatial and residual-energy uncertainties, UAV trust levels, and a supervisory HAPs layer.

Fig. 4.1 illustrates the considered trustworthy aerial edge computing architecture across three interacting layers. The IoT layer consists of distributed devices with uncertainty in both spatial position and residual energy. A bounded region distinguishes the estimated and true device locations, while residual energy perturbations reflect mismatches between the estimated and actual energy levels. The UAV layer provides aerial coverage and MEC support across the service area. UAVs carry different levels of trustworthiness and deploy according to regional criticality across mission-critical, reliability-sensitive, and infrastructure-assisted areas. The HAP layer monitors and coordinates multiple UAVs under dynamic conditions.

4.2 System Model and Problem Formulation

We propose an integrated three-layer architecture to support trustworthy MEC for IoT devices through adaptive UAV deployments and supervision by HAPs. The IoT layer consists of $\mathcal{N} = \{1, 2, \ldots, N\}$ devices distributed across the ground plane, with their spatial positions and residual energy subject to uncertainty, reflecting realistic operational conditions. The IoT devices specify trust thresholds T_n^{THR} that define the minimum reliability required for UAV-assisted MEC. Task offloading from *n-th* IoT device to the *m-th* UAV is modeled as $F_{nm} = (D_{nm}, \delta_{nm}, L_{nm}, T_n^{\text{THR}})$, where D_{nm} represents the data size, δ_{nm} the computational workload, L_{nm} the latency requirement, and T_n^{THR} the trust demand of *n-th* device from the *m-th* UAV. These characteristics establish the offloading requirements that guide UAV operations and supervisory decisions in the upper layers.

The aerial layer comprises $\mathcal{M} = \{1, 2, \ldots, M\}$ UAVs operating at mission-specific altitudes and executing MEC tasks for their associated IoT devices. The trust value T_{nm} expresses the reliability of the *m-th* UAV in serving the *n-th* IoT device and determines its suitability for that device's task under varying network conditions. Computational load, energy demand, and link quality vary with the specific device served, so trust holds per device-UAV pair rather than as a fixed attribute of the UAV alone. Prior work on trust management in IoT and edge networks follows the same principle and derives trust from measurable indicators of past and current behavior [31]. The threshold T_n^{THR} rests on the same value and represents the minimum reliability an IoT device requires from its serving UAV, so offloading proceeds only when $T_{nm} \geq T_n^{\text{THR}}$. Safety-critical monitoring and routine sensing enforce distinct minimum expectations through one consistent measure. UAVs also forward operational parameters such as residual energy, computing resources, and IoT sensing data to the supervisory layer comprising $\mathcal{K} = \{1, 2, \ldots, K\}$ HAPs positioned at stratospheric altitudes. The HAPs optimize UAV deployments and IoT-UAV associations, coordinate task scheduling, and regulate network behavior against uncertainties originating from the IoT layer. Supervision at this level sustains dependable MEC service delivery and stable network operation.

The IoT layer relies on binary decision variables relating to task offloading, IoT-UAV association, and IoT energy harvesting. The devices generate tasks individually, which can be executed locally or offloaded to a UAV for edge computing. The task offloading decision follows α_{nm}, where $\alpha_{nm} = 1$ denotes that the *n-th* IoT device offloads to the *m-th* UAV, and $\alpha_{nm} = 0$ otherwise. Task offloading must also satisfy the UAV's maximum coverage distance (d_m^{MAX}), which bounds the communication range. Edge computing support for the *n-th* IoT device requires a serving UAV that satisfies its operational requirements. The IoT device selection status is represented by β_n, where $\beta_n = 1$ indicates a selected device and $\beta_n = 0$ otherwise. The device must harvest energy before offloading whenever $E_n^{\text{RES}} < E_n^{\text{THR}}$, with $\gamma_n = 1$ marking this condition and $\gamma_n = 0$ otherwise.

4.2.1 Spatial Configuration and Uncertainty Model

The coordinates of the *n-th* IoT device, *m-th* UAV, and *k-th* HAP are denoted by $O_n = (x_n, y_n, 0)$, $O_m = (x_m, y_m, z_m)$, and $O_k = (x_k, y_k, z_k)$, where (x, y) indicate horizontal positions and z represents altitude. The IoT-UAV and UAV-HAP distances are expressed as:

$$d_{nm} = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2 + z_m^2}. \tag{4.1}$$

$$d_{mk} = \sqrt{(x_m - x_k)^2 + (y_m - y_k)^2 + (z_m - z_k)^2}. \tag{4.2}$$

The UAVs and HAPs are assumed to maintain precise positions through controlled trajectories and stable deployment. IoT devices experience location variations and modeling their spatial uncertainty is critical for reliable task offloading and efficient network planning [60].

In the proposed system, the serving UAV maintains an estimated location (O_n) for each IoT device. The actual IoT device position is subject to horizontal spatial uncertainty $\Delta O_n = (\Delta x_n, \Delta y_n, 0)$, bounded by $|\Delta x_n| \leq \varepsilon_x$ and $|\Delta y_n| \leq \varepsilon_y$, where ε_x and ε_y denote the maximum localization error bounds. Hence, the actual IoT location is given by:

$$\tilde{O}_n = O_n + \Delta O_n. \tag{4.3}$$

The offloading decision remains feasible ($\alpha_{nm} = 1$) only when the three-dimensional distance between $\tilde{O}_n$ and O_m does not exceed a maximum coverage distance d_m^{MAX}. The worst-case scenario is defined by the boundary perturbations $\Delta x_n^{\text{W}} = \text{sign}(x_n - x_m) \cdot \varepsilon_x$ and $\Delta y_n^{\text{W}} = \text{sign}(y_n - y_m) \cdot \varepsilon_y$, which maximize the IoT device's distance to the serving UAV. The resulting worst-case IoT-UAV distance is expressed as:

$$d_{nm}^{\text{W}} = \sqrt{(x_n - x_m + \Delta x_n^{\text{W}})^2 + (y_n - y_m + \Delta y_n^{\text{W}})^2 + z_m^2}. \tag{4.4}$$

The IoT spatial uncertainty modeling allows UAVs to account for potential deviations and enables robust IoT device associations, trust evaluation, and reliable MEC task offloading.

4.2.2 Communication Model

In the proposed system, IoT-UAV and UAV-HAP links are assumed to be dominated by line-of-sight (LoS) propagation, which is typical for aerial communication environments [76]. The IoT-UAV path loss is expressed as $\text{PL}_{nm} = \epsilon \left(\frac{4\pi d_{nm} f}{c}\right)^2$, where d_{nm} denotes the IoT-UAV distance, f the carrier frequency, c the speed of light, and ϵ captures additional propagation losses. The corresponding channel gain is expressed as $g_{nm} = \text{PL}_{nm}^{-1}$, and the achievable IoT-UAV data rate is given by R_{nm} [77]. These parameters provide a baseline representation of wireless link conditions following commonly used aerial communication models [78]. OFDMA is assumed as the underlying multiple-access mechanism for the considered aerial network [75]. UAV-HAP links follow the same channel characterization principles as IoT-UAV links. They carry supervisory coordination information between UAVs and the HAP, introducing signaling overhead D_{mk} and latency L_{mk} during network coordination. The adopted channel representation provides a realistic characterization of wireless links in the considered aerial network.

4.2.3 IoT Residual Energy-Uncertainty Model

The residual energy of IoT devices is inherently uncertain due to variations in operating conditions and energy consumption patterns [18]. The residual energy of *n-th* IoT device is modeled as:

$$E_n^{\mathrm{RES}} = E_n^{\mathrm{NOM}} + \chi E_n^{\mathrm{PER}}, \tag{4.5}$$

where E_n^{NOM} denotes the nominal residual energy, E_n^{PER} represents the magnitude of perturbations, and $\chi \in [-1, 1]$ is a bounded random variable capturing deviations from the nominal energy. The total available energy at *n-th* IoT device, including energy harvested from *m-th* UAV (E_{nm}^{HAR}), must satisfy the feasibility condition:

$$E_n^{\mathrm{RES}} + E_{nm}^{\mathrm{HAR}} \geq \beta_n E_n^{\mathrm{THR}}, \tag{4.6}$$

where E_n^{THR} denotes the minimum energy required for task offloading. The harvested energy E_{nm}^{HAR} supplements the residual energy of the *n-th* IoT device toward this requirement. Harvesting enters the model as a provisioned resource rather than a physical process, and carries only a distance-dependent cost defined later in this section.

4.2.4 UAV Hybrid Trustworthiness Model

Trustworthiness reflects the long-term behavioral credibility of a UAV in delivering MEC services and captures its consistency under dynamic operating conditions. Reliability and quality-of-service (QoS) metrics differ in scope, since they quantify instantaneous performance such as latency or throughput. The distinction matters in UAV-assisted MEC systems, where mobility, workload fluctuations, and security risks jointly influence UAV behavior. HAPs therefore employ trustworthiness as a high-level decision signal alongside conventional performance metrics, informing UAV selection and task orchestration beyond short-term link or computational quality. We propose a hybrid trustworthiness model that integrates dependent and independent UAV performance metrics. The formulation retains interactions among operational factors while preserving UAV-specific characteristics. Chapter 3 modeled UAV trustworthiness through independent metrics alone, using an additive formulation. A key limitation of this formulation was its omission of the operational dependencies among these metrics.

The dependent metrics comprise UAV computational resources (COM), operating environment (OPR), residual energy (ENE), and task completion (TAS), grouped as $\psi^{\mathrm{DPN}} = \{\mathrm{COM}, \mathrm{OPR}, \mathrm{ENE}, \mathrm{TAS}\}$. These factors are operationally interdependent, as energy availability directly influences both task completion and computational capacity, while environmental conditions affect energy consumption and task execution. The independent metrics comprise IoT-UAV communication stability (STA) and security assurance (SEC), grouped as $\psi^{\mathrm{IND}} = \{\mathrm{STA}, \mathrm{SEC}\}$. Communication stability varies across IoT devices, which justifies its treatment as an independent metric. Security evaluations likewise depend on the IoT service under consideration. The grouping keeps the dependent metrics sensitive to their interactions while the independent metrics reflect performance across diverse MEC conditions.

• UAV Computational Resources: Available computational resources at the UAV determine its capacity to process offloaded tasks [51]. The corresponding metric is:

$$T_{nm}^{\mathrm{COM}} = 1 - \frac{A_{nm}}{A_m^{\mathrm{MAX}}}, \tag{4.7}$$

where A_{nm} and A_m^{MAX} denote the requested computational load and maximum processing capacity, respectively.

• UAV Operating Environment: Terrain and obstacles impede UAV operation while serving IoT devices [81]. The resulting difficulty is captured as:

$$T_{nm}^{\mathrm{OPR}} = 1 - \frac{\varrho_{nm}}{\varrho^{\mathrm{MAX}} + \varepsilon}, \tag{4.8}$$

where ϱ_{nm} denotes the observed environmental difficulty for *m-th* UAV to serve *n-th* IoT device, ϱ^{MAX} is the maximum difficulty across all IoT-UAV links, and ε is a small constant for numerical stability.

• UAV Residual Energy: Task execution draws on the serving UAV's available energy [41]. Energy sufficiency follows as:

$$T_{nm}^{\mathrm{ENE}} = 1 - \frac{E_{nm}}{E_m^{\mathrm{MAX}}}, \tag{4.9}$$

where E_{nm} denotes the energy required by the *m-th* UAV to execute the *n-th* IoT device's task and E_m^{MAX} represents the UAV's maximum available energy capacity. A UAV's total energy budget supports two functions: sustaining flight through propulsion, and delivering MEC services through computation and communication. Propulsion energy depends on trajectory and aircraft parameters rather than on task execution decisions, and remains constant in this model [90]. The quantities E_{nm} and E_m^{MAX} isolate the service-related component of the budget.

• UAV Task Completion: This metric reflects the historical task execution capability of the UAV when serving the IoT device. It is expressed as:

$$T_{nm}^{\mathrm{TAS}} = \frac{F_{nm}^{\mathrm{SUC}}}{F_{nm}}, \tag{4.10}$$

where F_{nm}^{SUC} denotes the number of successfully completed tasks and F_{nm} represents the total number of tasks assigned to the *m-th* UAV by the *n-th* IoT device.

• IoT-UAV Communication Stability: This metric quantifies the reliability of the communication link between a UAV and an IoT device [80]. It is expressed as:

$$T_{nm}^{\mathrm{STA}} = 1 - \frac{\lambda_{nm}}{\lambda_m^{\mathrm{MAX}}}, \tag{4.11}$$

where λ_{nm} denotes the observed packet loss rate between the *n-th* IoT device and the *m-th* UAV, and λ_m^{MAX} represents the maximum tolerable packet loss for that UAV.

• UAV Security Assurance: This metric evaluates a UAV's effectiveness in detecting security threats from associated IoT devices. The detected threat count ϕ_{nm}^{DET} reflects the number of anomalous or malicious events flagged by the UAV's onboard threat detection mechanism for the *n-th* device's traffic over an observation window. The estimated total threats ϕ_{m}^{TOT} at the *m-th* UAV represent the true number of threats present across all its currently associated devices. This total includes threats that evade detection and therefore cannot be observed directly. This quantity is estimated from the UAV's aggregate detected count $\phi_{m}^{\mathrm{DET}} = \sum_{n\in\mathcal{N}_m} \phi_{nm}^{\mathrm{DET}}$. The recall $\mathcal{R}_m$, the fraction of true threats correctly identified, then gives $\phi_{m}^{\mathrm{TOT}} \approx \phi_{m}^{\mathrm{DET}}/\mathcal{R}_m$. This value is typically obtained through offline validation on labeled data [91]. This estimate of ϕ_{m}^{TOT} is statistically grounded, rather than an unconstrained model parameter. The resulting security assurance metric is expressed as:

$$T_{nm}^{\mathrm{SEC}} = \frac{\phi_{nm}^{\mathrm{DET}}}{\phi_{m}^{\mathrm{TOT}} + \varepsilon}, \tag{4.12}$$

where ϕ_{nm}^{DET} and ϕ_{m}^{TOT} denote the detected and estimated total threat counts, and ε is a small positive constant ensuring numerical stability.

The trust metrics in (4.7) – (4.12) share a common range of [0, 1] and permit direct comparison across UAVs [12]. Dependent and independent metrics relate to overall reliability according to the degree of correlation among their underlying operational factors. A UAV that consistently fails to complete assigned tasks remains unreliable despite substantial computing capacity and stable environmental conditions, since one weak factor offsets several favorable ones. Reliability engineering formalizes this principle for series-connected systems, whose overall reliability equals the product of individual component reliabilities rather than their average [92]. The dependent metrics in ψ^{DPN} follow the same principle and combine as a product. The constant $\varepsilon > 0$ guards against a single degenerate measurement dominating the result.

Communication stability and security assurance remain independent of the UAV's internal resource state. A UAV low on energy or computational capacity can still maintain a stable link and detect threats reliably. The two metrics are also mutually independent, since communication link quality and threat detection vary independently across devices. The metrics in ψ^{IND} therefore combine additively, with each contributing its measured value directly to the total. The resulting trust value T_{nm} exceeds unity in its raw form and undergoes min-max normalization to restore the $[0, 1]$ range.

Normalization corrects the scale of T_{nm} without linking the two metric groups, so a UAV with severely diminished dependent metrics could still satisfy T_n^{THR} on the strength of its independent metrics alone. Each dependent metric must therefore satisfy $T_{nm}^{(i)} \geq \tau^{\text{MIN}}$ for every $i \in \psi^{\text{DPN}}$ before an IoT-UAV pair is considered for association. A UAV with near-depleted energy fails this condition regardless of how favorably its communication stability or security assurance scores. The condition covers the dependent metrics alone, since those reflect basic operational capacity. The independent metrics instead capture the quality of specific service dimensions. The overall trust value is expressed as:

$$T_{nm} = \prod_{i \in \psi^{\text{DPN}}} \left(T_{nm}^{(i)} + \varepsilon\right) + \sum_{j \in \psi^{\text{IND}}} T_{nm}^{(j)}, \tag{4.13}$$

where i and j index metrics in the dependent set ψ^{DPN} and independent set ψ^{IND}, respectively, and ε maintains numerical stability. This formulation combines the multiplicative influence of dependent metrics with the additive effect of independent factors, enabling a comprehensive measure of trustworthiness between UAVs and IoT devices in dynamic MEC environments.

4.2.5 UAV Deployment Cost Model

This model captures the deployment cost of the *m-th* UAV in serving the *n-th* IoT device, comprising offloading, computation, and energy-harvesting components. This decomposition is consistent with cost-aware UAV deployment and mission planning objectives established in multi-UAV IoT networks [82]. Offloading cost scales with both the task data size D_{nm} and the distance $\tilde{d}_{nm}$ to the realized IoT device position $\tilde{O}_n$, introduced in (4.3) to capture spatial uncertainty. Computation cost similarly scales with D_{nm} by applying a UAV-specific cost coefficient c_m^{COM} that is drawn from a representative range to reflect that different UAVs may incur different computation costs in practice. Harvesting cost scales only with $\tilde{d}_{nm}$, applying whenever the *n-th* device requires energy harvesting, indicated by γ_n. The deployment cost is expressed as:

$$C_{nm}^{\mathrm{DEP}} = D_{nm} c^{\mathrm{OFF}} \tilde{d}_{nm} + D_{nm} c_m^{\mathrm{COM}} + \gamma_n c^{\mathrm{HAR}} \tilde{d}_{nm}, \tag{4.14}$$

where c^{OFF} and c^{HAR} denote fixed offloading and harvesting cost coefficients, respectively. The total deployment cost is obtained by summing C_{nm}^{DEP} over all active IoT-UAV associations, directly supporting cost-aware task assignment within the normalized optimization framework.

4.2.6 Latency Model

This model quantifies the time required to complete computational tasks generated by the *n-th* IoT device within the proposed aerial edge computing framework. It accounts for processing at the device, computation at the *m-th* UAV, and control signaling with the *k-th* HAP.

• Local Latency: It is the time required for the *n-th* IoT device to execute its computational task locally. This latency depends on the task data size D_n, the computational workload per bit δ_n, and the device's processing frequency f_n, and is expressed as $L_n^{\mathrm{LOC}} = \frac{D_n \delta_n}{f_n}$.

• Edge Latency: It is the time required for the *n-th* IoT device to offload its task to the *m-th* UAV and receive the computed result. This latency depends on the task size D_{nm}, the computational workload per bit δ_{nm}, and the UAV processing frequency f_m, and is expressed as $L_{nm}^{\mathrm{EDGE}} = \frac{2D_{nm}}{R_{nm}} + \frac{D_{nm}\delta_{nm}}{f_m}$. The transmission rate R_{nm} is assumed identical for uplink and downlink, while propagation delays are considered negligible due to the relatively small distance between the *n-th* IoT device and *m-th* UAV.

Edge latency is computed as a fixed value per task, without queuing dynamics from task arrival timing or service ordering. Related robust UAV-MEC offloading models adopt the same treatment and derive task completion delay deterministically from allocated resources [69]. Resource contention nonetheless stays bounded, since the device capacity and computational load constraints limit the tasks assigned to each UAV.

• Control Latency: It is the time required for the *m-th* UAV to exchange control and status information with the *k-th* HAP. This latency component reflects the coordination and supervisory information exchanged to support efficient system operation while avoiding energy-intensive transfer of raw IoT data over long-range links. It is expressed as $L_{mk}^{\text{CC}} = \frac{D_{mk}}{R_{mk}} + \frac{2d_{mk}}{c}$, where R_{mk} is the transmission rate, D_{mk} is the UAV's control data size, d_{mk} is the distance between the UAV and HAP, and c is the speed of light.

The task execution latency of the *n-th* IoT task depends on whether it is processed locally or offloaded to the *m-th* UAV. The *m-th* UAV may exchange control information with the *k-th* HAP, introducing a delay associated with system-level control. The binary decision variable $\eta_{mk} \in \{0, 1\}$ specifies the state of a control update between the *m-th* UAV and the *k-th* HAP, $\eta_{mk} = 1$ when a control update occurs and $\eta_{mk} = 0$ otherwise. The total latency is expressed as:

$$L_{nm}^{\text{TOT}} = (1 - \alpha_{nm})L_n^{\text{LOC}} + \alpha_{nm} L_{nm}^{\text{EDGE}} + \eta_{mk} L_{mk}^{\text{CC}}. \tag{4.15}$$

4.2.7 Multi-criteria Objective Function

This work defines a multi-criteria utility function that jointly balances the key operational trade-offs of the proposed framework. Trust, deployment cost, and total latency are each mapped to the interval $[0, 1]$ through min-max scaling, given by $\overline{\mathcal{X}} = \frac{(\mathcal{X} - \mathcal{X}^{\text{MIN}})}{(\mathcal{X}^{\text{MAX}} - \mathcal{X}^{\text{MIN}})}$. The resulting utility function is expressed as $\mathcal{U}_{nm} = \alpha_{nm}(\overline{T_{nm}} + \beta_n - \overline{C_{nm}^{\text{DEP}}} - \overline{L_{nm}^{\text{TOT}}})$. Here $\overline{T_{nm}}$, $\overline{C_{nm}^{\text{DEP}}}$, and $\overline{L_{nm}^{\text{TOT}}}$ denote the normalized trust, deployment cost, and total latency. The device selection variable β_n is binary and requires no normalization. The variable α_{nm} determines whether a given IoT-UAV pair contributes to the objective. An active pair earns a fixed reward through β_n, and the achieved trust, cost, and latency values leave that reward unchanged. The formulation favors serving more IoT devices over optimizing trust, cost, and latency across a fixed subset.

$$\max_{\alpha_{nm},\beta_n,O_m} \sum_{n=1}^{N}\sum_{m=1}^{M} \mathcal{U}_{nm} \tag{4.16}$$

Subject to:

Assignment and Association:

$$C1 : \alpha_{nm},\ \beta_n \in \{0,1\}, \quad \forall\ n, m \tag{4.16a}$$

$$C2 : \sum_{m=1}^{M} \alpha_{nm} = \beta_n, \quad \forall\ n \tag{4.16b}$$

$$C3 : \sum_{n=1}^{N} \alpha_{nm} \leq \vartheta, \quad \forall\ m \tag{4.16c}$$

Latency and Trust:

$$C4 : \alpha_{nm} L_{nm}^{\mathrm{TOT}} \leq L_{nm}^{\mathrm{MAX}}, \quad \forall\ n, m \tag{4.16d}$$

$$C5 : \overline{T_{nm}} \geq T_n^{\mathrm{THR}} - \mathbb{M}\left(1 - \alpha_{nm}\right), \forall\ n, m \tag{4.16e}$$

UAV Resources:

$$C6 : \sum_{n=1}^{N} \alpha_{nm} A_{nm} \leq A_m^{\mathrm{MAX}}, \forall\ m \tag{4.16f}$$

$$C7 : \sum_{n=1}^{N} \alpha_{nm} E_{nm} \leq E_m^{\mathrm{MAX}}, \forall\ m \tag{4.16g}$$

IoT Energy and Spatial Uncertainty:

$$C8 : E_n^{\mathrm{RES}} + E_{nm}^{\mathrm{HAR}} \geq \beta_n E_n^{\mathrm{THR}},\ \forall\ n \tag{4.16h}$$

$$C9 : |\Delta x_n| \leq \varepsilon_x,\ |\Delta y_n| \leq \varepsilon_y, \forall\ n \tag{4.16i}$$

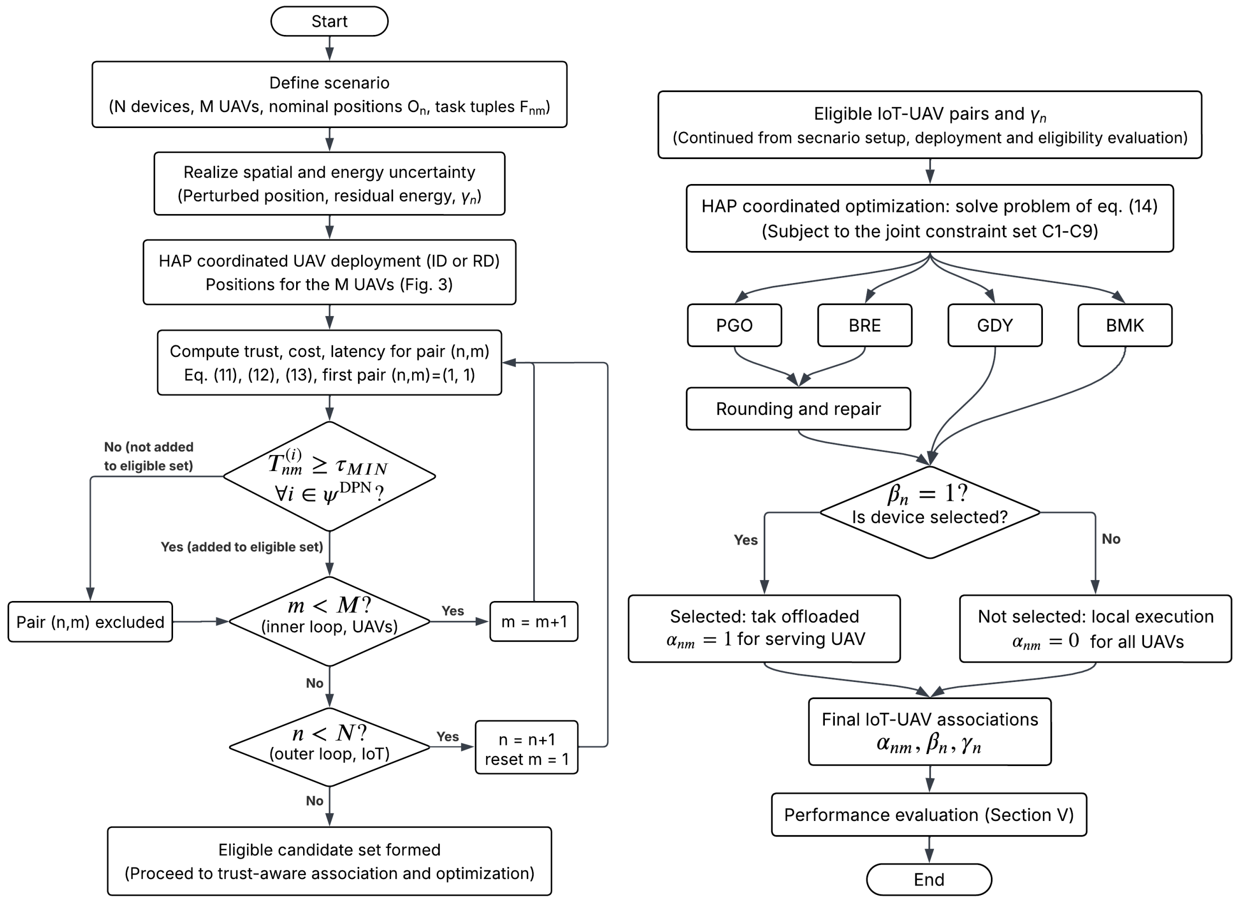
where constraints C1, C2 and C3 ensure consistent task assignment and association. Each selected IoT device connects to a single UAV, and the number of associated IoT devices per UAV stays within ϑ. Constraint C4 bounds the total task execution latency L_{nm}^{TOT} by the maximum allowable value L_{nm}^{MAX}. Constraint C5 enforces the trust threshold T_n^{THR} through a Big-M formulation, which activates only when offloading occurs. Constraints C6 and C7 impose computational and energy capacity limits at each UAV, ensuring that the aggregate workload A_{nm} and energy consumption E_{nm} do not exceed A_m^{MAX} and E_m^{MAX}, respectively. Constraint C8 enforces that the residual energy E_n^{RES} together with the harvested energy E_{nm}^{HAR} satisfies the minimum energy requirement E_n^{THR} for task execution. C9 bounds the spatial perturbations $(\Delta x_n, \Delta y_n)$ by $(\varepsilon_x, \varepsilon_y)$ to account for localization uncertainty.

4.3 Proposed Solution

The optimization problem in (4.16) maximizes system utility by jointly assigning IoT devices to UAVs and allocating MEC resources. Binary variables model offloading and selection decisions, while continuous variables represent UAV positions and resource allocations. The combinatorial nature of the binary variables renders the problem non-convex and NP-hard. We propose a HAP-supervised hierarchical solution that balances computational efficiency with coordinated system-level decision-making. Fig. 4.2 illustrates the overall execution flow, from the initial positioning of UAVs through the final IoT-UAV associations. The first step deploys UAVs across the service region to serve a heterogeneous IoT distribution, filtering candidate IoT-UAV pairs against the trust operability floor (Fig. 4.2(a)). The second step performs trust-aware optimization to assign IoT devices and allocate MEC resources under the established UAV deployment, comparing multiple candidate algorithms (Fig. 4.2(b)). The following subsections detail the deployment strategies and optimization algorithms underlying each stage [93].

4.3.1 UAVs Deployment Strategy

This stage determines UAV positions across the service region to provide efficient coverage and facilitate IoT device assignment and resource allocation.



(a) UAV deployment and eligibility evaluation of candidate IoT-UAV pairs.

(b) Trust-aware optimization and final IoT-UAV association.

Figure 4.2: The proposed HAP-supervised, trust-aware execution framework, spanning UAV deployment to final IoT-UAV association and resource optimization.

• Random Deployments (RD): UAV positions are set independently of the spatial distribution of IoT devices. The horizontal coordinates (x_m, y_m) of the *m-th* UAV are drawn uniformly at random within the service region, and its altitude (z_m) comes from a predefined range. A three-dimensional distance metric (4.4) evaluates IoT-UAV connectivity under bounded uncertainty in the horizontal coordinates of IoT devices, and returns a conservative estimate of the effective separation. RD serves as a baseline configuration, with placement that disregards IoT device density and spatial clustering.

• Intelligent Deployments (ID): The spatial distribution of IoT devices within the service region determines UAV positions. Clustering partitions the area into M groups, and the *m-th*

UAV occupies the centroid of its assigned cluster. We employ the K-means algorithm because it yields centroid locations directly and minimizes intra-cluster distances [85]. Hierarchical and density-based methods instead require higher computational effort and produce irregular cluster shapes, which limits their effectiveness for centroid-based UAV positioning [86]. The optimized UAV coordinates minimize the sum of squared worst-case IoT-UAV distances:

$$\min_{\{O_m\}_{m=1}^{M}} \sum_{n=1}^{N} \min_{m} \left(d_{nm}^{\mathrm{W}}\right)^2, \tag{4.17}$$

where d_{nm}^{W} is defined in (4.4) and incorporates the horizontal spatial uncertainty of the IoT devices.

Fig. 4.3 depicts the overall UAV deployment procedure, where the workflow branches into random and intelligent strategies after incorporating IoT spatial uncertainty. In both cases, worst-case IoT-UAV distances are evaluated to establish robust IoT-UAV associations. Furthermore, the intelligent branch iteratively updates UAV positions through clustering-driven refinement. The resulting UAV coordinates serve as the spatial foundation for subsequent system-level optimization.

4.3.2 Optimization Algorithms

The optimization stage applies the algorithms established in Section 3.3. The proposed PGO algorithm relaxes the binary decision variables and augments the utility with a penalty term. SQP resolves the resulting formulation, and post-processing restores feasibility. The BRE, GDY, and BMK algorithms serve as comparisons. Two aspects distinguish the present formulation. The constraint set extends to C2–C9, accommodating the spatial and residual energy uncertainty of the IoT devices. The SQP iterations therefore enforce the linearized forms of the extended constraint set. The utility $\mathcal{U}_{nm}$ in (4.16) incorporates harvested energy and the worst-case IoT-UAV separation. The computational complexity of each algorithm follows the analysis in Section 3.3.3.

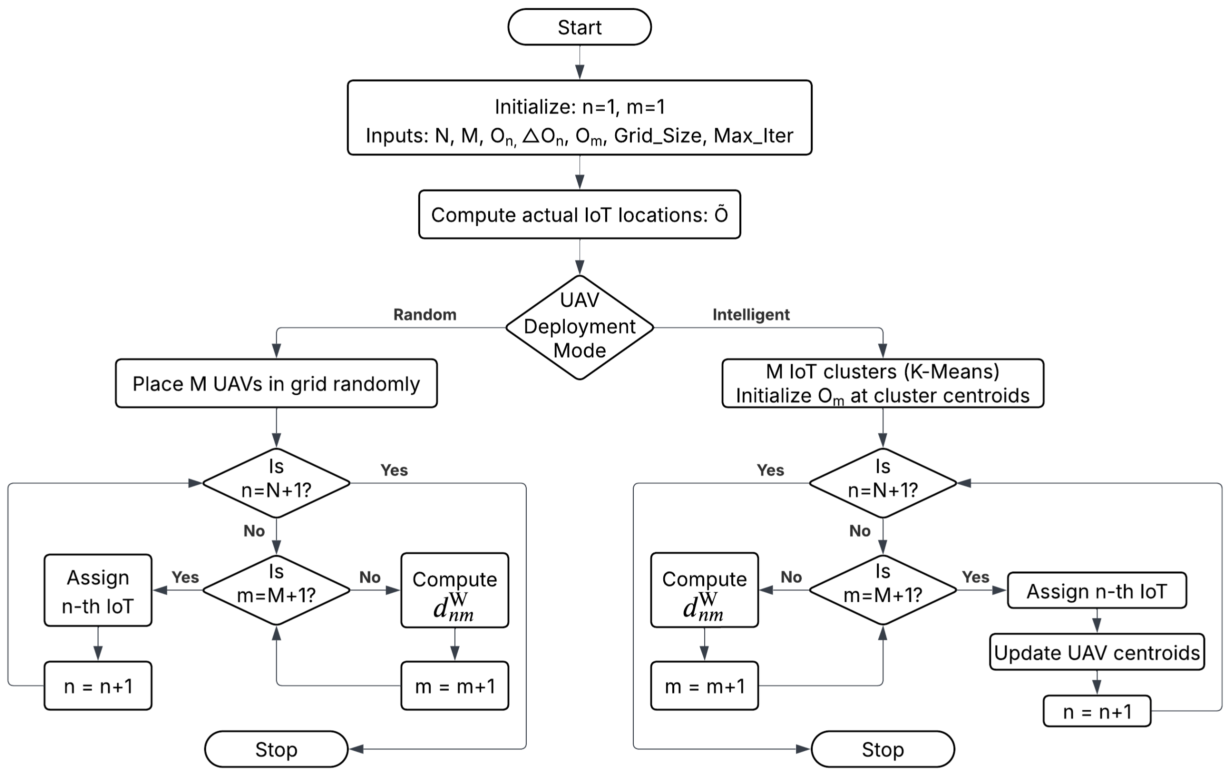


Figure 4.3: Flowchart of the UAV deployment framework, illustrating random and intelligent deployments under IoT spatial uncertainty.

4.4 Performance Analysis

The proposed framework is evaluated through metrics that characterize association capability, robustness, and trust-aware efficiency under IoT energy and spatial perturbations. The number of connected IoT devices measures how effectively IoT-UAV links are sustained across operating conditions. The multi-criteria utility (4.16) reflects the joint influence of trustworthiness, deployment cost, and delay-related tradeoffs within the optimization objective. The price of robustness counts the IoT devices disconnected relative to the optimal solution (BMK-ID) and quantifies the degradation in association performance under perturbations. Trust efficiency divides the total trustworthiness attained by the normalized overall latency and expresses trusted operation per unit delay.

4.4.1 Simulation Setup

We evaluate the proposed framework through simulations based on realistic IoT-UAV and HAP characteristics [18, 55]. The study considers varying numbers of IoT devices and UAVs deployed over a 1000 m × 1000 m service region, with each UAV covering users within a radius of $d_m^{\text{MAX}} = 500$ m. The compact size of this region relative to the number of UAVs considered allows many IoT devices to remain simultaneously reachable by more than one UAV under the latency, capacity, and trust constraints of the proposed formulation. The relevant optimization decision is therefore not whether a device can reach a UAV, but which reachable UAV offers the best trade-off in trust, cost, and latency under spatial and energy uncertainty. HAPs are capable of providing coverage across extensive geographic areas. Service delivery within a given remote area depends instead on the cluster of UAVs deployed within it, consistent with the scenario simulated in this work. This trust-aware association framework can be applied independently across several remote areas supervised by a single HAP.

IoT devices experience spatial perturbations within a 250 m bound along both the x and y-axes, with deviations from their nominal positions scaled between $0 - 100\%$ of this baseline, and residual energy fluctuations of up to 40% to reflect realistic variations affecting task scheduling and resource allocation. IoT devices require a minimum trust threshold ranging from 50% to 90%, with a minimum operability floor ($\tau^{\text{MIN}} = 0.1$) additionally required of each dependent trust indicator, ensuring UAV eligibility reflects basic operational capacity rather than trust score alone. The simulations further incorporate total latency ($L_{nm}^{TOT} = 1 - 5$ s), deployment cost coefficients $\left(c^{OFF}=0.4,\ c^{HAR}=5,\ c_m^{COM}\in[0.75, 2]\right)$, UAV and IoT device computational capacities ($A_m = 7 - 15$ GHz, $A_n = 1$ GHz), and the Big-M parameter ($\mathbf{M} = 10$, enforcing constraint C5), providing a comprehensive performance assessment. Table 4.1 summarizes the simulation parameters with their corresponding units.

Table 4.1: Summary of the simulation parameters.

Parameter	**Variable**	**Value**
IoT devices	N	20, 60, 100
UAVs	M	2, 3, 4
Service region	–	1000×1000 m^2
UAV altitude	z_m	30–50 m
UAV coverage radius	d_m^{MAX}	500 m
IoT spatial perturbation bound	$\varepsilon_x, \varepsilon_y$	250 m
IoT spatial perturbation	$\Delta_n^{\mathrm{PER}}\%$	0–100
IoT residual energy perturbation	$E_n^{\mathrm{PER}}\%$	0–40
IoT energy threshold	$E_n^{\mathrm{THR}}\%$	20
IoT-UAV trust	$\overline{T_{nm}}$	0–1
IoT trust threshold	$T_n^{\mathrm{THR}}\%$	50–90
Trust operability floor	τ^{MIN}	0.1
Total latency	L_{nm}^{TOT}	1–5 s
Offloading cost coefficient	c^{OFF}	0.4
Harvesting cost coefficient	c^{HAR}	5
Computation cost coefficient	c_m^{COM}	0.75–2
Max. IoT per UAV	ϑ	50
IoT computation power	A_n	1 GHz
UAV computation power	A_m	7–15 GHz
Big-M value	$\mathbf{M}$	10
PGO penalty coefficient	γ	1
PGO convergence threshold	ϵ^{THR}	1×10^{-6}

4.4.2 Simulation Results

Table 4.2 summarizes the connectivity between IoT devices and UAVs for different optimization algorithms under RD and ID UAV deployments. Trust thresholds range from 50% to 90%, with a residual energy threshold of 20% and fixed energy and spatial perturbations. The proposed PGO-ID algorithm consistently delivers near-optimal results across all network sizes ($N = \{20, 60, 100\}$) and UAV counts ($M = \{2, 3, 4\}$), demonstrating stable operation under stringent trust requirements. IoT-UAV connectivity generally decreases as trust thresholds rise. In this context, the PGO-ID algorithm preserves a high level of

Table 4.2: IoT-UAV connectivity achieved by optimization algorithms under random and intelligent UAV deployments (RD and ID), for varying IoT trust thresholds (T_n^{THR}), with 20% IoT residual energy threshold and fixed energy and spatial perturbations.

IoT devices (N)	UAVs (M)	2					3					4				
	T_n^{THR} %	50	60	70	80	90	50	60	70	80	90	50	60	70	80	90
	Algorithms	Number of connected IoT devices versus M, T_n^{THR}% ($E_n^{\mathrm{PER}} = 20\%$ and $\Delta_n^{\mathrm{PER}} = 25\%$)														
20	BMK-RD	10	7	7	6	5	11	8	8	7	6	12	10	10	9	7
	BRE-RD	9	6	6	5	5	11	8	8	7	6	12	10	8	6	6
	GDY-RD	6	6	6	5	5	8	8	8	7	6	9	10	9	7	6
	PGO-RD	9	6	6	5	5	11	8	8	7	6	12	9	10	8	6
	BMK-ID	12	9	7	6	5	13	13	12	9	7	14	14	14	12	10
	BRE-ID	12	9	7	6	5	13	13	12	9	7	14	14	13	12	10
	GDY-ID	7	7	5	5	5	11	11	10	8	7	13	13	12	11	10
	PGO-ID	11	8	6	5	4	13	13	12	9	7	14	14	13	12	10
60	BMK-RD	27	24	21	15	14	31	26	24	20	19	36	31	29	24	21
	BRE-RD	17	15	15	12	12	3	2	2	2	2	18	18	17	16	13
	GDY-RD	23	21	19	14	14	25	24	24	20	19	27	25	25	21	20
	PGO-RD	23	20	18	14	13	30	25	24	20	19	35	30	29	23	20
	BMK-ID	30	25	23	19	19	39	27	24	18	17	42	34	34	25	24
	BRE-ID	22	19	19	16	16	18	23	8	8	8	5	5	5	5	5
	GDY-ID	19	19	19	19	19	23	21	20	18	17	28	28	28	25	24
	PGO-ID	25	19	18	15	15	37	26	24	18	17	42	33	32	24	23
100	BMK-RD	35	32	30	27	23	44	42	40	35	31	57	50	48	41	36
	BRE-RD	13	12	11	10	9	1	1	1	1	1	19	13	13	10	6
	GDY-RD	19	19	19	19	19	25	25	25	25	25	30	30	30	30	30
	PGO-RD	30	26	23	19	17	37	34	33	28	26	47	42	42	35	32
	BMK-ID	42	37	34	30	27	54	50	48	44	41	60	58	56	53	46
	BRE-ID	21	18	23	19	18	4	4	3	3	3	28	26	25	25	21
	GDY-ID	21	21	21	21	19	26	26	26	26	26	32	32	32	32	32
	PGO-ID	37	30	27	22	18	49	42	41	38	35	54	51	49	47	41

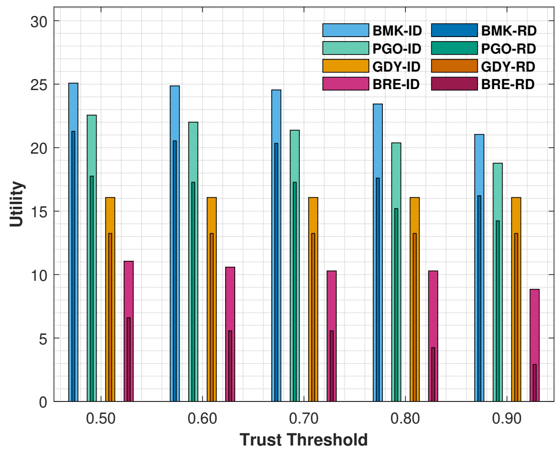
connectivity, whereas BRE-ID and GDY-ID exhibit reduced associations, which are further reduced under random UAV deployments. The addition of UAVs improves coverage across all algorithms, and PGO-ID efficiently leverages these additional resources to maintain reliable links. Intelligent UAV deployments further strengthen coverage and reduce the occurrence of disconnected devices, highlighting the impact of UAV deployment strategy on network reliability.

Fig. 4.4 presents the system utility performance of optimization algorithms under varying trust thresholds, residual energy and spatial perturbations for 100 IoT devices and 4 UAVs.

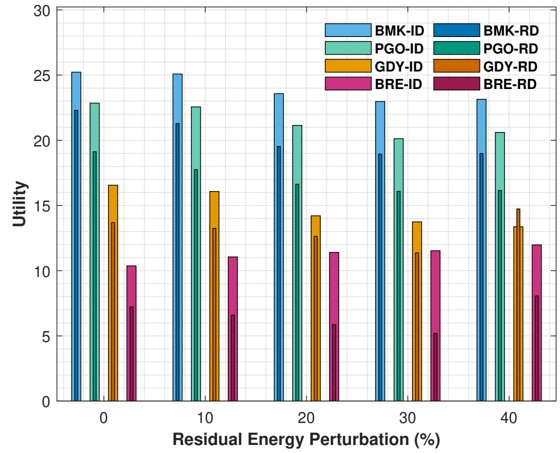
The utility declines for all algorithms as trust or perturbation levels increase. The BMK-ID algorithm provides an upper-bound, and the proposed PGO-ID algorithm consistently maintains close performance across all conditions. In Fig. 4.4(a) and (b), the GDY-ID and BRE-ID algorithms show lower utility, which reflects their limited ability to balance trust, energy reliability, and deployment efficiency. In Fig. 4.4(b), when residual energy perturbation reaches 40%, the GDY-RD algorithm slightly outperforms the GDY-ID algorithm because its lower sensitivity to individual device trust offsets the stricter constraints of GDY-ID. Furthermore, in Fig. 4.4(c), only at the maximum spatial perturbation (100%), the GDY-ID algorithm approaches the performance of the proposed PGO-ID algorithm. This indicates that extreme positional uncertainty reduces the advantage of intelligent deployments.

Fig. 4.5 presents the price of robustness, quantifying IoT disconnectivity under varying trust thresholds, residual energy perturbations, and spatial uncertainties for 100 IoT devices and 4 UAVs. The BMK-ID algorithm achieves the lowest disconnectivity across all scenarios. The proposed PGO-ID algorithm closely matches this performance and shows strong resilience under challenging conditions. The BRE-ID and BRE-RD algorithms exhibit the highest disconnectivity, indicating limited ability to sustain reliable IoT-UAV associations. In Fig. 4.5(a), the GDY-ID and GDY-RD algorithms maintain a stable but higher level of disconnectivity as trust thresholds increase. Fig. 4.5(b) shows that GDY-RD and BRE-RD achieve an approximate 9% reduction in IoT disconnectivity under residual energy perturbations between 30% to 40%. This improvement occurs because random UAV deployments may coincide with devices that retain sufficient energy. In Fig. 4.5(c), GDY-RD and BRE-RD display fluctuating disconnectivity as spatial perturbations grow, reflecting the impact of positional uncertainty on random deployments.

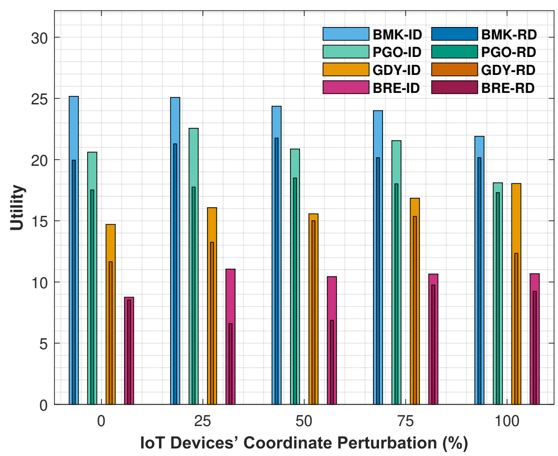
Fig. 4.6 illustrates trust efficiency, defined as the total network trustworthiness relative to normalized operational latency. In contrast to the IoT trust threshold (T_n^{THR}), which sets the minimum requirement for IoT devices, trust efficiency reflects how effectively the network maintains high trust levels under latency and perturbation constraints. The BMK-ID



(a) $E_n^{\text{PER}} = 10\%$ and $\Delta_n^{\text{PER}} = 25\%$



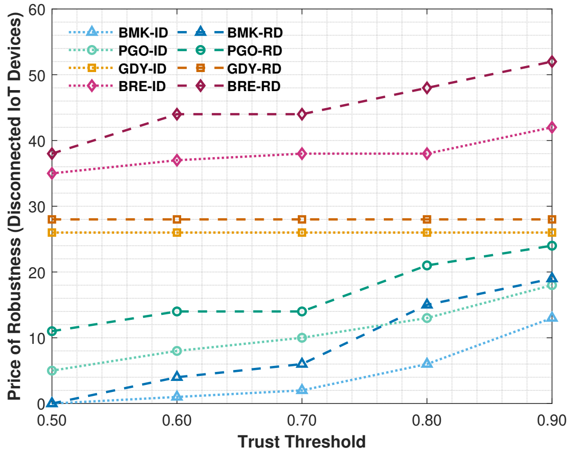
(b) $T_n^{\text{THR}} = 50\%$ and $\Delta_n^{\text{PER}} = 25\%$



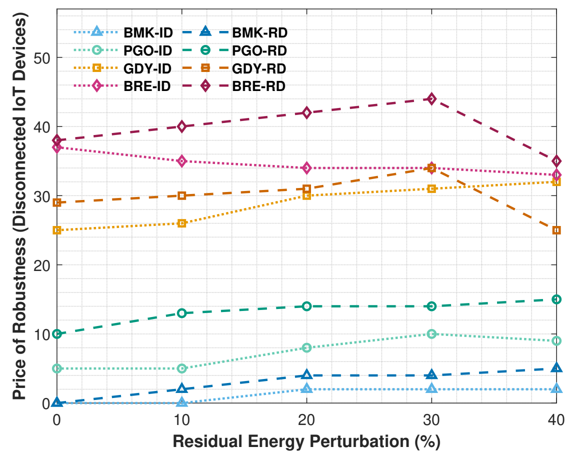
(c) $T_n^{\text{THR}} = 50\%$ and $E_n^{\text{PER}} = 10\%$

Figure 4.4: System utility sensitivity to trust-aware IoT perturbations in an IoT-UAV network with $N = 100$ IoT devices and $M = 4$ UAVs.

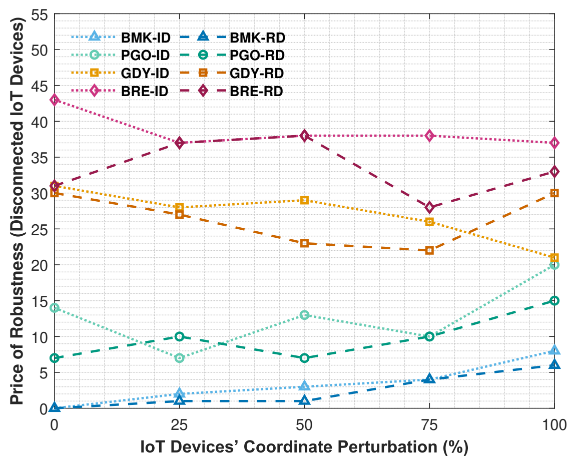
algorithm consistently achieves the highest efficiency, with the proposed PGO-ID algorithm closely approximating this performance across all scenarios. The BRE-ID and BRE-RD algorithms perform the worst, indicating a limited ability to deliver trustworthy edge computing services. In Fig. 4.6(a), trust efficiency declines as the trust threshold increases, while GDY-ID matches PGO-ID between 0.8 and 0.9, temporarily sustaining competitive associations under stringent requirements. Fig. 4.6(b) shows efficiency decreasing with residual energy perturbations, whereas BRE-RD shows slight increase for higher levels of residual energy perturbations. In Fig. 4.6(c), spatial perturbations produce more variable efficiency,



(a) $E_n^{\mathrm{PER}} = 10\%$ and $\Delta_n^{\mathrm{PER}} = 25\%$



(b) $T_n^{\mathrm{THR}} = 50\%$ and $\Delta_n^{\mathrm{PER}} = 25\%$



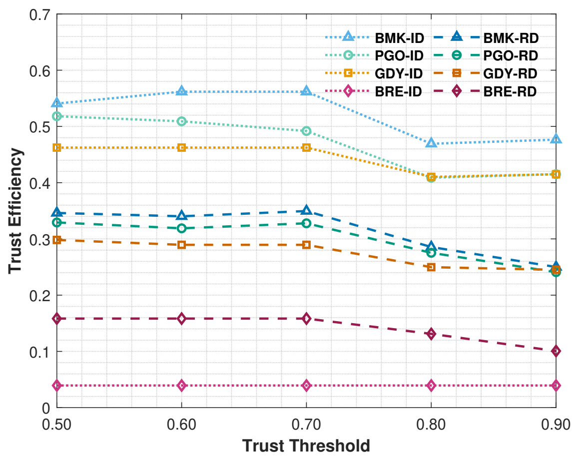
(c) $T_n^{\mathrm{THR}} = 50\%$ and $E_n^{\mathrm{PER}} = 10\%$

Figure 4.5: Price of robustness sensitivity to IoT trust and perturbation conditions in an IoT-UAV network with $N = 100$ IoT devices and $M = 4$ UAVs.

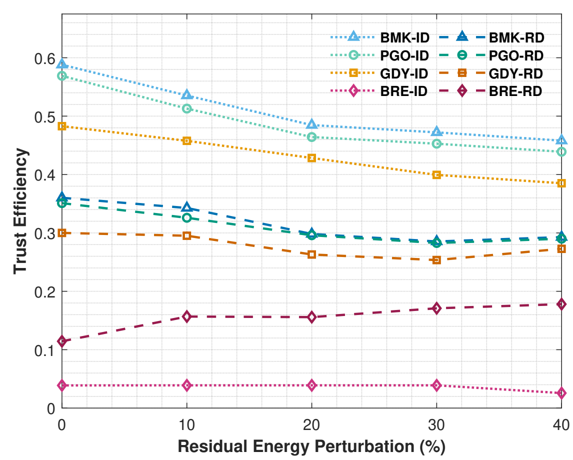
with GDY-ID slightly outperforming PGO-ID between 67% and 100% due to occasional favorable alignment of UAV coverage and IoT positions. Overall, the proposed PGO-ID algorithm maintains robust trust efficiency under diverse network conditions.

4.5 Summary

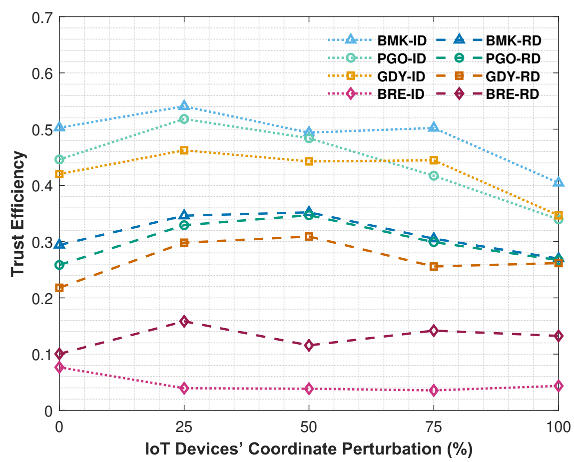
This chapter presents a trust-aware aerial edge computing framework for 6G networks in rural, remote, and emergency scenarios. The framework integrates HAPs, UAVs, and energy-



(a) $E_n^{\mathrm{PER}} = 10\%$ and $\Delta_n^{\mathrm{PER}} = 25\%$



(b) $T_n^{\mathrm{THR}} = 50\%$ and $\Delta_n^{\mathrm{PER}} = 25\%$



(c) $T_n^{\mathrm{THR}} = 50\%$ and $E_n^{\mathrm{PER}} = 10\%$

Figure 4.6: Trust efficiency sensitivity to IoT trust thresholds and perturbation conditions in an IoT-UAV network with $N = 60$ IoT devices and $M = 4$ UAVs.

harvesting IoT devices under operational constraints. A hybrid model quantifies UAV trustworthiness through dependent and independent metrics. The model also accounts for perturbed residual energy, harvested energy, and spatial uncertainty at the IoT devices, and enforces their heterogeneous trust thresholds. A multi-criteria utility function then optimizes UAV-IoT associations subject to computational, energy, latency, and trustworthiness constraints. PGO-ID solves this formulation at low complexity and reaches near-optimal network utility with high trust efficiency. The algorithm outperforms baseline relaxation and greedy optimization across the tested configurations.

Chapter 5

Trust- and Priority-Aware Resource Management for Reliable UAV-Assisted MEC in 6G IoT Networks

5.1 Introduction

IoT devices deployed in disaster-affected and emergency-driven environments generate tasks that differ in urgency, from routine monitoring to mission-critical response [94]. UAV-mounted MEC servers reach these devices where terrestrial infrastructure is unavailable, though each UAV holds finite energy and computing capacity [95]. Capacity may therefore fall short of demand, and the aerial network must select which IoT devices receive service and which UAV serves each admitted device.

Trust expresses whether a UAV meets the service credibility an IoT device demands, examined in prior work through secure transmission, node authentication, and service reliability [32, 96, 97]. Task priority ranks device urgency so that time-critical demands reach computational resources ahead of routine traffic [22, 98]. Selecting IoT devices by urgency without verifying UAV capability admits tasks that may fail, while verifying capability without ranking urgency serves whichever devices the formulation reaches first. The two criteria have been pursued in separate formulations, leaving their joint evaluation within a single admission decision unaddressed.

5.2 System Model and Problem Formulation

Fig. 5.1 illustrates the proposed aerial network that comprises $\mathcal{M} = \{1, 2, \ldots, M\}$ UAVs acting as MEC nodes and serving $\mathcal{N} = \{1, 2, \ldots, N\}$ IoT devices. The IoT devices follow a uniformly random distribution over the coverage area, and generate computationally intensive tasks whose local execution may be infeasible or inefficient within the required delay bounds, motivating MEC-assisted offloading. The IoT devices exhibit heterogeneous trust requirements and task priorities that further differentiate the service demands. Similarly, the UAVs differ in computational capacity, residual energy, and trust levels. The UAV deployments follow an initial coverage-driven placement that undergoes continuous refinement as network conditions evolve. The joint optimization of resource allocation, UAV positioning, and task scheduling entails a computational burden that the UAV layer cannot absorb without compromising its serving function. We assume an intelligent controller that gathers periodic reports from the underlying UAVs on their trust indicators, resource availability, and task statistics to evaluate trustworthiness, verify resource feasibility, and determine IoT device association and selection decisions. The controller initiates a new optimization cycle when reported metrics deviate beyond acceptable thresholds, ensuring operational consistency across the network.

The three-dimensional positions of the *m-th* UAV and the *n-th* IoT device are given by $O_m = (x_m, y_m, z_m)$ and $O_n = (x_n, y_n, 0)$, respectively. The channel between the *n-th* IoT device and its serving *m-th* UAV is modeled under the assumption of LoS dominated

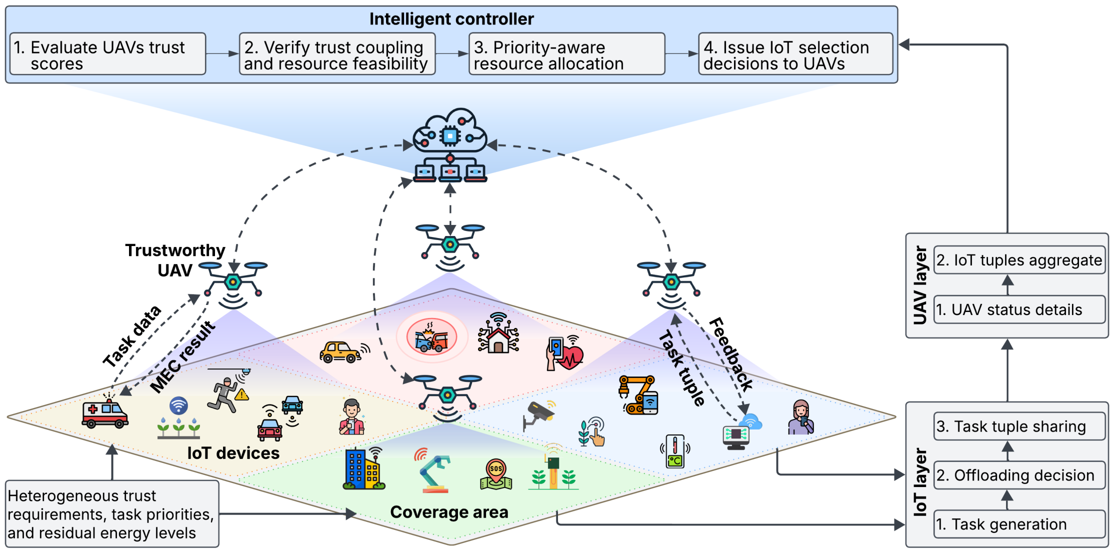


Figure 5.1: Trust-aware aerial network serving heterogeneous IoT devices with varying priorities, trust thresholds, and energy levels.

propagation with symmetric uplink and downlink characteristics. The IoT-UAV distance is given by $d_{nm} = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2 + z_m^2}$, and the associated path loss, channel gain, and noise characteristics are modeled in accordance with [76, 77]. The achievable rate is expressed as $R_{nm} = B_{nm} \log_2(1 + \gamma_{nm})$, where B_{nm} is the allocated bandwidth and γ_{nm} denotes the signal-to-noise ratio experienced over the link. OFDMA is adopted to enable interference-free simultaneous uplink transmissions from IoT devices to their serving UAVs [75].

The task of the *n-th* IoT device potentially assigned to the *m-th* UAV is represented as $F_{nm} = (D_{nm}, \delta_{nm}, L_n^{\text{MAX}}, T_n^{\text{THR}}, \psi_{nm})$, where D_{nm} is the data size in bits, δ_{nm} is the computational workload in CPU cycles per bit, L_n^{MAX} is the maximum tolerable latency, T_n^{THR} is the minimum trust level demanded by the device from its serving UAV, and ψ_{nm} is the task priority. The trust compatibility between an IoT device and its candidate UAV forms the primary eligibility condition for IoT-UAV associations, ensuring that only UAVs meeting T_n^{THR} are considered as valid association candidates in the optimization. Furthermore, task priority ψ_{nm} and the UAV constraints on coverage, energy, and computational capacity shape the resource scheduling decisions across the network.

5.2.1 IoT Task Prioritization Model

The prioritization of IoT tasks governs scheduling decisions at edge computing nodes, ensuring that tasks of greater urgency and operational significance receive preferential access to available computational resources. The authors of [21] and [64] addressed deadline-aware scheduling and joint task offloading under resource constraints in ground-based MEC settings, where prioritization was driven by computational demand and task urgency. The deployment of UAVs as MEC servers introduces additional service heterogeneity that demands a more expressive task priority characterization. In this work, we characterize task priority through three attributes, namely deadline urgency, data volume, and criticality level, denoted by $\mathcal{I} = \{\text{DLN, SIZ, CRT}\}$.

• Task Deadline: This attribute quantifies the urgency of a task relative to its remaining latency budget. The remaining latency is expressed as $L_{nm}^{\text{REM}} = L_n^{\text{MAX}} - L_{nm}$. It captures the latency budget remaining after task execution, where L_n^{MAX} is the maximum allowable latency, and L_{nm} is the expected task execution latency between the *n-th* IoT device and the *m-th* UAV. The deadline priority score is accordingly expressed as $\psi_{nm}^{\text{DLN}} = 1 - \frac{L_{nm}^{\text{REM}}}{L_n^{\text{MAX}}}$. It is bounded within [0, 1], where a higher score reflects a task approaching its latency limit and naturally elevating its scheduling urgency.

• Task Size: This attribute reflects the data volume of the task and its consequent demand on the communication and computational resources of the serving UAV. The size priority score is expressed as $\psi_{nm}^{\text{SIZ}} = 1 - \frac{D_{nm}}{D_n^{\text{MAX}}}$, where D_{nm} is the task data size and D_n^{MAX} is the maximum task data size of the *n-th* IoT device. It is bounded within [0, 1], where a lower score reflects a more resource-intensive task, moderating its scheduling priority relative to less demanding tasks competing for the resources of the *m-th* UAV. This prevents a single large task from consuming capacity that could otherwise serve several smaller tasks. Hence, raising the priority with size instead would reward exactly the tasks most likely to displace others. The task's operational importance informs a separate attribute of the priority score.

• Task Criticality: This attribute reflects the degree of operational significance associated with a task, as determined by the nature and priority class of its originating device. The criticality priority score is expressed as $\psi_{nm}^{\mathrm{CRT}} = \frac{\kappa_n}{\kappa^{\mathrm{MAX}}}$, where κ_n is the criticality score assigned to the task of the *n-th* IoT device and κ^{MAX} is the maximum criticality level observed across the network. It is bounded within [0, 1], where a higher score reflects greater operational significance and elevates the task precedence accordingly.

The overall priority score of the task associated with the *n-th* IoT device and the *m-th* UAV consolidates the individual attribute scores into a unified measure as follows:

$$\psi_{nm} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \psi_{nm}^{(i)}. \tag{5.1}$$

Consolidating the attributes as a plain average in (5.1) treats deadline urgency, size, and criticality identically, reflecting the absence of an operational basis for favoring one over the others. Their relative influence could instead be adjusted through explicit weights, such as increasing the emphasis on deadline urgency for latency-sensitive deployments.

We quantize the resulting continuous IoT prioritization score (ψ_{nm}) into five discrete priority levels as presented in Table 5.1, each reflecting a distinct degree of task urgency. This structured mapping translates the fine-grained priority information into a basis suitable for differentiated scheduling across the UAV-MEC network.

5.2.2 UAV Trustworthiness Model

In the proposed framework, the trustworthiness of the *m-th* UAV quantifies its capacity to fulfill the computational service demands of the *n-th* IoT device under prevailing network conditions [99]. Chapter 3 combined UAV trust metrics additively, without distinguishing dependent factors from independent ones. Chapter 4 instead combined dependent metrics multiplicatively, retaining an additive treatment for independent metrics. This design still required a separate per-metric threshold to prevent independent metrics from compensating for a weak dependent one. The present work removes this reliance on an external safeguard.

Table 5.1: Quantization of IoT task priority scores.

Priority score range	**Priority level**	**Task interpretation**
[0,0.2)	1	Delay tolerant
[0.2,0.4)	2	Low urgency
[0.4,0.6)	3	Moderate urgency
[0.6,0.8)	4	High urgency
[0.8,1]	5	Mission critical

Capability-driven metrics instead combine multiplicatively throughout, so the trust score grows sensitive to deficiency in any single dimension. The trust score T_{nm} consolidates these dimensions relative to the trust threshold (T_n^{THR}) of the *n-th* IoT device, with constituent metrics bounded within $[\tau^{\mathrm{MIN}}, 1]$, where τ^{MIN} reflects the minimum acceptable operational capability. The intelligent controller relies on T_{nm} to determine IoT-UAV association decisions, with each metric described as follows:

• Coverage Suitability: This metric reflects the degree to which the *n-th* IoT device lies within the operational coverage footprint of the *m-th* UAV [22, 23]. It is expressed as $T_{nm}^{\mathrm{COV}} = \frac{d_m^{\mathrm{MAX}} - d_{nm}}{d_m^{\mathrm{MAX}}}, \;\; d_{nm} \leq d_m^{\mathrm{MAX}}$, where d_m^{MAX} is the maximum coverage radius of the *m-th* UAV. It is bounded within [0, 1], where higher values reflect a device positioned well within the serving range. The metric evaluates to zero for IoT devices lying beyond the coverage boundary of the *m-th* UAV.

• Energy Sufficiency: This metric reflects the energy available at the *m-th* UAV beyond the demand imposed by the *n-th* IoT device's task [56]. It is expressed as $T_{nm}^{\mathrm{ENE}} = \frac{E_m^{\mathrm{RES}} - E_{nm}^{\mathrm{DEM}}}{E_m^{\mathrm{MAX}}}, \quad E_m^{\mathrm{RES}} \geq E_{nm}^{\mathrm{DEM}}$, where E_m^{RES} denotes residual energy of the UAV, E_{nm}^{DEM} denotes the energy demanded to serve the task of the IoT device, and E_m^{MAX} denotes the UAV's maximum energy capacity. It is bounded within [0, 1], where higher values indicate a UAV with sufficient energy reserves to sustain task execution.

• Computational Capacity: This metric reflects the computational resources available at the *m-th* UAV beyond the demand imposed by the *n-th* IoT device's task [40]. It is expressed as $T_{nm}^{\mathrm{CMP}} = \frac{A_m^{\mathrm{RES}} - A_{nm}^{\mathrm{DEM}}}{A_m^{\mathrm{MAX}}}, \quad A_m^{\mathrm{RES}} \geq A_{nm}^{\mathrm{DEM}}$, where A_m^{RES} is the residual computational capacity,

$A_{nm}^{\text{DEM}} = \delta_{nm} D_{nm}$ is the computational demand of the task, and A_m^{MAX} is the maximum computational capacity. It is bounded within [0, 1], where higher values indicate a UAV with sufficient computational reserves to process the assigned IoT tasks.

• IoT Service Satisfaction: This metric captures the latency performance of the *m-th* UAV in serving the computational tasks of the *n-th* IoT device relative to its maximum allowable bound (L_n^{MAX}) [34]. It is expressed as $T_{nm}^{\text{SAT}} = 1 - \frac{L_{nm}}{L_n^{\text{MAX}}}, \quad L_{nm} \leq L_n^{\text{MAX}}$, where L_{nm} is the expected task completion latency of the *n-th* IoT device. It is bounded within [0, 1], where higher values indicate greater latency satisfaction of the associated IoT device.

• Security Resilience: This metric captures the overall security posture of the *m-th* UAV through the empirically estimated probability of a security breach [35]. It is defined as $P_m^{\text{SEC}} = \frac{B_m^{\text{BRH}}}{B_m^{\text{INT}}}$, where B_m^{BRH} denotes the number of security breaches recorded, and B_m^{INT} is the total number of monitored interactions. The security resilience metric is accordingly expressed as $T_{nm}^{\text{SEC}} = 1 - P_m^{\text{SEC}}, \quad B_m^{\text{BRH}} \leq B_m^{\text{INT}}$. It is bounded within [0, 1], where higher values reflect a UAV with a stronger security posture.

The overall trust score is computed through a multiplicative aggregation of the constituent metrics in $\mathcal{J} = \{\text{COV, ENE, CMP, SAT, SEC}\}$, normalized by $|\mathcal{J}|$. This ensures sensitivity to deficiency across any individual dimension while maintaining scores within $[\tau^{\text{MIN}}, 1]$.

$$T_{nm} = \left(\prod_{j \in \mathcal{J}} T_{nm}^{(j)} \right)^{\frac{1}{|\mathcal{J}|}}. \tag{5.2}$$

5.2.3 IoT Residual Energy Perturbation Model

Energy uncertainty is an inherent characteristic of resource-constrained IoT devices. Dynamic usage patterns and sensing inaccuracies cause the reported residual energy to deviate from its nominal value (E_n^{NOM}) [18]. This deviation is strictly unidirectional, as energy can only degrade from its nominal state. The perturbed residual energy of the *n-th* IoT device is expressed as $E_n^{\text{RES}} = \max(E_n^{\text{NOM}} - \zeta_n E_n^{\text{PER}}, 0)$, where E_n^{PER} denotes the maximum perturbation level, $\zeta_n \sim \mathcal{U}[0, 1]$ is a uniformly distributed random variable sampled independently across IoT devices, and the max operator enforces non-negativity of the residual energy. The case $\zeta_n = 1$ represents the worst-case perturbation yielding $E_n^{\text{RES}} = E_n^{\text{NOM}} - E_n^{\text{PER}}$, while $\zeta_n = 0$ corresponds to no deviation. The residual energy takes values within $[\underline{E}_n, \overline{E}_n]$, where $\overline{E}_n = E_n^{\text{NOM}}$ and $\underline{E}_n = E_n^{\text{NOM}} - E_n^{\text{PER}}$. The normalized residual energy is expressed as:

$$\tilde{E}_n^{\text{RES}} = \begin{cases} \dfrac{E_n^{\text{RES}} - \underline{E}_n}{\overline{E}_n - \underline{E}_n} & \text{if } E_n^{\text{PER}} > 0 \\ 1 & \text{if } E_n^{\text{PER}} = 0 \end{cases} \tag{5.3}$$

where $\tilde{E}_n^{\text{RES}} = 1$ indicates no deviation from the nominal energy state and $\tilde{E}_n^{\text{RES}} = 0$ reflects full depletion to the lower perturbation bound. The energy feasibility condition for the *n-th* IoT device incorporates both residual energy and UAV-assisted harvesting [23]. It is expressed as $\tilde{E}_n^{\text{RES}} + \alpha_{nm}\tilde{E}_{nm}^{\text{HAR}} \geq \tilde{E}_n^{\text{THR}}$, where $\tilde{E}_{nm}^{\text{HAR}} \in [0, 0.5]$ denotes the normalized energy harvested from the *m-th* UAV. It is a derived parameter determined by the association decision α_{nm} and the residual energy state of the *n-th* IoT device, bounded at 0.5 to preserve harvesting as a supplementary mechanism. The IoT device retains primary responsibility for its own energy sufficiency while benefiting from UAV assistance when necessary. The energy harvesting incurs an operational cost C_{nm}^{HAR} that contributes a latency overhead to the task completion process. This cost interacts with the UAV-side energy sufficiency T_{nm}^{ENE} to jointly govern energy feasibility across the IoT-UAV association.

5.2.4 Cost Model

Cost captures the cumulative expenditure imposed across IoT-UAV associations in facilitating reliable MEC service delivery to heterogeneous IoT devices [39]. Quantifying this expenditure across UAV operational overhead, service demand, and energy harvesting dimensions is essential for governing resource utilization within the network.

• UAV Operational Cost: The operational cost of the *m-th* UAV reflects two inherently distinct expenses arising from its participation in the network. The fixed hardware cost is expressed as $\frac{C_m^{\text{FIX}}}{C^{\text{MAX}}}$, where C_m^{FIX} denotes the UAV-specific capital and maintenance expense and C^{MAX} is the maximum hardware cost across all UAVs. The service load cost is expressed as $\frac{|\mathcal{S}_m|}{\vartheta}$, where $\mathcal{S}_m = \{n \in \mathcal{N} : \alpha_{nm} = 1\}$ is the subset of IoT devices actively associated with the *m-th* UAV and ϑ is the maximum IoT device connectivity capacity per UAV. The UAV operational cost is accordingly expressed as $C_m^{\text{UAV}} = \frac{C_m^{\text{FIX}}}{C^{\text{MAX}}} + \frac{|\mathcal{S}_m|}{\vartheta}$.

• Service Demand Cost: The service demand cost reflects the interaction-specific overhead of MEC service delivery, jointly driven by the task priority ψ_{nm} and the declared trust threshold T_n^{THR} of the *n-th* IoT device. It is expressed as $C_{nm}^{\text{TRU}} = \psi_{nm} \cdot T_n^{\text{THR}}$, where higher values of either dimension elevate the cost. This coupling introduces a trade-off between priority-driven demand and trust-governed eligibility. Higher priority devices impose greater service overhead, partially offsetting their utility reward within the multi-criteria objective. This cost applies to active IoT-UAV device associations.

• IoT Energy Harvesting Cost: The energy harvesting cost captures the fraction of *m-th* UAV's maximum energy capacity (E_m^{MAX}) consumed in providing energy assistance to the associated IoT device. It is expressed as $C_{nm}^{\text{HAR}} = \frac{E_{nm}^{\text{HAR}}}{E_m^{\text{MAX}}}$, where E_{nm}^{HAR} denotes the actual energy harvested by the *n-th* IoT device from the *m-th* UAV. The cost elevates as the harvested energy increases relative to UAV capacity, and is activated when the device's normalized residual energy falls below its offloading threshold (E_n^{THR}).

The overall cost C_{nm} captures the cumulative operational burden arising from the *m-th* UAV serving the *n-th* IoT device.

$$C_{nm} = C_m^{\text{UAV}} + C_{nm}^{\text{TRU}} + C_{nm}^{\text{HAR}}. \tag{5.4}$$

5.2.5 Latency Model

Latency is defined as the time elapsed from task generation to acquiring the computational results. It serves as a critical performance indicator under heterogeneous IoT task deadlines and constrained network resources. In the proposed framework, the task completion latency encompasses distinct components governed by the association variable α_{nm} and the selection variable β_n.

- Local Computation Latency: Tasks executed locally at the IoT device ($\alpha_{nm} = 0$) incur the local computation latency. It is expressed as $L_n^{\mathrm{LOC}} = \frac{D_n \delta_n}{f_n}$, where f_n denotes the processing frequency of the *n-th* IoT device.

- Edge Computation Latency: Tasks assigned to the serving UAV for MEC processing ($\alpha_{nm}=1$ and $\beta_n=1$) incur the edge computation latency. It captures the transmission over the IoT-UAV link and the MEC processing time, further conditioned on the energy state of the IoT device.

$$L_{nm}^{\mathrm{EDG}} = \frac{D_{nm}}{R_{nm}} + \frac{D_{nm}\delta_{nm}}{f_m} + \left[\tilde{E}_n^{\mathrm{RES}} < \tilde{E}_n^{\mathrm{THR}}\right] \cdot \tau^{\mathrm{HAR}} \cdot \tilde{E}_{nm}^{\mathrm{HAR}}, \tag{5.5}$$

where f_m is the processing frequency of the *m-th* UAV, R_{nm} is the achievable rate defined in the communication model, and τ^{HAR} is the harvesting overhead coefficient (in seconds per unit normalized harvested energy). The size of the computational result and propagation delays are considered negligible given the short inter-node distances. The indicator function $[\tilde{E}_n^{\mathrm{RES}} < \tilde{E}_n^{\mathrm{THR}}]$ activates the harvesting overhead when the normalized residual energy falls below the offloading threshold of the *n-th* IoT device.

The overall task completion latency consolidates the complementary local and edge latency components as $L_{nm} = (1 - \alpha_{nm})L_n^{\mathrm{LOC}} + \alpha_{nm}L_{nm}^{\mathrm{EDG}}$, where α_{nm} governs the active latency component, with edge latency incurred only when both $\alpha_{nm} = 1$ and $\beta_n = 1$ are satisfied through the optimization framework. Furthermore, the UAV layer relays rejection feedback to notify IoT devices whose tasks cannot be admitted under the prevailing network conditions, to prioritize service integrity over universal task admission.

5.2.6 Multi-criteria Optimization Problem

We formulate a multi-criteria utility function to balance the competing objectives of trustworthiness, prioritized device admission, cost efficiency, and latency performance across IoT-

UAV interactions. The continuous criteria are normalized to [0,1] through min-max scaling, where $\overline{\mathcal{X}} = \frac{\mathcal{X}-\mathcal{X}_{\min}}{\mathcal{X}_{\max}-\mathcal{X}_{\min}}$ for $\mathcal{X} \in \{T_{nm}, \psi_{nm}, C_{nm}, L_{nm}\}$. The utility function associated with the *m-th* UAV serving the *n-th* IoT device is expressed as $\mathcal{H}_{nm} = \nu_1\overline{T}_{nm}+\nu_2\overline{\psi}_{nm}-\nu_3\overline{C}_{nm}-\nu_4\overline{L}_{nm}$, where ν_q for $q \in \{1,2,3,4\}$ are non-negative weighting coefficients satisfying $\sum_{q=1}^{4}\nu_q = 1$, enabling flexible prioritization of individual criteria. The joint optimization over the continuous UAV positioning variables O_m, the binary device selection variable β_n, and the IoT-UAV association variable α_{nm} results in a mixed integer non-linear program (MINLP). It is NP-hard due to the combinatorial nature of binary decisions and non-linear interdependencies among the optimization variables.

$$\max_{O_m,\alpha_{nm},\beta_n} \sum_{n=1}^{N}\sum_{m=1}^{M} \alpha_{nm}\mathcal{H}_{nm} \tag{5.6}$$

Subject to:

$$C1: \sum_{m=1}^{M}\alpha_{nm} = \beta_n, \qquad \forall\, n \tag{5.6a}$$

$$C2: \sum_{n=1}^{N}\alpha_{nm} \le \vartheta, \qquad \forall\, m \tag{5.6b}$$

$$C3: d_{nm} \le d_m^{\text{MAX}}, \qquad \forall\, n,m \tag{5.6c}$$

$$C4: T_{nm} \ge T_n^{\text{THR}} - \mathbb{M}(1-\alpha_{nm}), \qquad \forall\, n,m \tag{5.6d}$$

$$C5: \alpha_{nm}L_{nm}^{\text{EDG}} \le L_n^{\text{MAX}}, \qquad \forall\, n,m \tag{5.6e}$$

$$C6: \sum_{n=1}^{N}\alpha_{nm}\cdot A_{nm}^{\text{DEM}} \le A_m^{\text{MAX}}, \qquad \forall\, m \tag{5.6f}$$

$$C7: \sum_{n=1}^{N}\alpha_{nm}\cdot E_{nm}^{\text{DEM}} \le E_m^{\text{MAX}}, \qquad \forall\, m \tag{5.6g}$$

$$C8: \tilde{E}_n^{\text{RES}} + \alpha_{nm}\tilde{E}_{nm}^{\text{HAR}} \ge \beta_n\tilde{E}_n^{\text{THR}}, \qquad \forall\, n,m \tag{5.6h}$$

$$C9: \alpha_{nm} \in \{0,1\}, \quad \beta_n \in \{0,1\}, \qquad \forall\, n,m \tag{5.6i}$$

where C1 links each IoT device's association decisions to its selection status β_n, and C2 ensures the number of devices associated with each UAV does not exceed the connectivity capacity ϑ. C3 enforces the coverage prerequisite for valid associations, while C4 governs trust eligibility through a Big-M formulation to activate the threshold T_n^{THR} only for active association pairs. C5 bounds the task completion latency within L_n^{MAX}. C6 and C7 ensure the aggregate computational and energy demands across associated devices do not exceed the UAV capacities A_m^{MAX} and E_m^{MAX}, respectively. C8 ensures the total available energy at each admitted device, including any harvested energy from the serving UAV, satisfies the minimum threshold $\tilde{E}_n^{\mathrm{THR}}$ required for task execution. C9 defines the binary domain of the decision variables.

5.3 Proposed Solution

The proposed framework addresses the optimization problem in (5.6) through a two-stage solution strategy [100]. The first stage establishes UAV positions and defines the IoT device association strategy, while the second stage performs multi-criteria resource optimization through the proposed and comparative optimization algorithms.

5.3.1 First Stage: UAV Deployment and Association Strategy

In this stage, we employ K-means clustering to partition the IoT device distribution into M clusters, with each cluster assigned to one UAV [85]. The centroid of each cluster determines the deployment position of its assigned UAV. K-means is particularly suited to this setting, as density-based approaches such as DBSCAN yield a variable cluster count and hierarchical methods produce irregular shapes less conducive to centroid-based positioning [86]. IoT devices contribute to the clustering proportionally to their task priority ψ_{nm}, ensuring UAV centroids reflect the heterogeneous task distribution across the deployment area. The UAV positions are obtained by minimizing the priority-weighted sum of squared device-to-UAV distances:

$$\min_{\{O_m\}_{m=1}^{M}} \sum_{n=1}^{N} \psi_{nm} \left\| O_n - O_m \right\|^2 . \tag{5.7}$$

The converged UAV positions establish the coverage topology within which trust credentials are evaluated against IoT device thresholds as an admission criterion, governing the following decision strategies:

• Random Association Decision (RD): In this strategy, the framework selects feasible UAV-device pairs through uniform random sampling among all pairs satisfying coverage, trust eligibility, and resource feasibility constraints. The selection operates independently of task priority, serving as a performance baseline that isolates the contribution of priority-aware decision making to overall network utility.

• Intelligent Association Decision (ID): In this strategy, the framework selects feasible UAV-device pairs through the joint evaluation of trust, priority, cost, and latency criteria within the multi-criteria optimization framework. Task priority ψ_{nm} ensures devices with greater urgency and operational significance are preferentially served under constrained network resources.

5.3.2 Second Stage: Multi-Criteria Resource Optimization

The optimization stage applies the algorithms established in Section 3.3, extended here to the trust and priority formulation of (5.6) [99]. The proposed PGO algorithm relaxes the binary decision variables and augments the utility $\alpha_{nm}\mathcal{H}_{nm}$ with a penalty term (Γ), which SQP resolves under the linearized forms of C1–C8 and C9$^{\mathrm{CONT}}$. The BRE, GDY, and BMK algorithms serve as comparisons. Three aspects distinguish the present formulation. The branching in BMK proceeds over α_{nm} alone, with β_n determined implicitly through C1 at each step. The repair procedure removes the associations contributing least to $\mathcal{H}_{nm}$ until C1–C9 are collectively satisfied. Performance under both RD and ID association strategies reveals the contribution of intelligent association decisions to overall network utility. The computational complexity of each algorithm follows the analysis in Section 3.3.3.

5.4 Performance Analysis

5.4.1 Simulation Setup

The performance of the proposed framework is evaluated through simulations conducted over a 1000×1000 m^2 deployment area, where $N = 100$ IoT devices are uniformly distributed and $M \in \{2, \ldots, 6\}$ UAVs serve as aerial MEC nodes [18]. The IoT devices are characterized by diverse trust requirements (T_n^{THR}), task priorities (ϕ_{nm}), and residual energy perturbations (E_n^{PER}) of up to 40% relative to their reported energy levels. These features provide a rigorous analysis of the framework's trust evaluation, priority-aware scheduling, and admission control mechanisms. UAV heterogeneity in processing frequency (f_m) and trust credentials (T_{nm}), combined with trust threshold conditions spanning $T_n^{\mathrm{THR}} \in [0.5, 0.9]$, establishes a comprehensive basis for assessing reliable MEC service delivery under constrained and varying network conditions. Table 5.2 summarizes the simulation parameters and their corresponding values used throughout the evaluation.

5.4.2 Performance Metrics

We assess the performance of the proposed framework through metrics that collectively capture operational effectiveness, resilience to energy perturbation, and sensitivity under diverse network conditions. Three-dimensional IoT-UAV association patterns reveal the spatial and operational admission behavior of the framework across different solution approaches. Trust efficiency is defined as the ratio of the achieved trust score to the normalized task completion latency, which serves as the primary indicator of joint service reliability and responsiveness. The price of robustness captures network resilience through disconnected IoT devices under increasing residual energy perturbations across distinct resource and demand scenarios. The effect of individual optimization criteria on admission decisions, device connectivity,

Table 5.2: Simulation parameters.

Parameter	Symbol	Value
Number of IoT devices	N	100
IoT task data size	D_{nm}	[2, 5] Mb
IoT computation workload	δ_{nm}	1 cycles/bit
IoT processing frequency	f_n	1 GHz
IoT task priority levels	ψ_{nm}	$\{1, \ldots, 5\}$
IoT task criticality score	κ_n	[0, 1]
IoT energy perturbation	E_n^{PER}	[0, 40] %
IoT energy threshold	$\tilde{E}_n^{\text{THR}}$	0.20
IoT trust threshold	T_n^{THR}	[0.5, 0.9]
Minimum trust metric	τ^{MIN}	0.5
Max. devices per UAV	ϑ	50
Number of UAVs	M	$\{2, \ldots, 6\}$
UAV altitude	z_m	[30, 50] m
UAV processing frequency	f_m	[7, 15] GHz
UAV fixed hardware cost	C_m^{FIX}	[0, 1]
UAV coverage radius	d_m^{MAX}	500 m
Network bandwidth	B_{nm}	1 MHz
Channel power gain	g_0	1.42×10^{-4}
Noise power	σ^2	-80 dBm/Hz
Utility function weights	ν_q	Table 5.3
Big-M value	$\mathbb{M}$	10
Penalty coefficient	γ	2
SQP iterations	$\mathcal{I}$	≤ 400
Convergence tolerance	ϵ^{THR}	10^{-6}

and overall network performance is further examined through illustrative utility weight profiles spanning equal, trust, priority, cost, and latency emphasis configurations, as listed in Table 5.3.

5.4.3 Simulation Results

Fig. 5.2 presents the three-dimensional IoT-UAV association patterns for $N = 100$ IoT devices and $M = 5$ UAVs under equal utility weights, trust threshold $T_n^{\text{THR}} = 0.50$, and in the

Table 5.3: Utility parameters weight configurations.

Weight emphasis	**Utility parameters weights**			
	ν^{TRS}	ν^{PRI}	ν^{CST}	ν^{LAT}
Equal	0.25	0.25	0.25	0.25
Trust	0.40	0.20	0.20	0.20
Priority	0.20	0.40	0.20	0.20
Cost	0.20	0.20	0.40	0.20
Latency	0.20	0.20	0.20	0.40

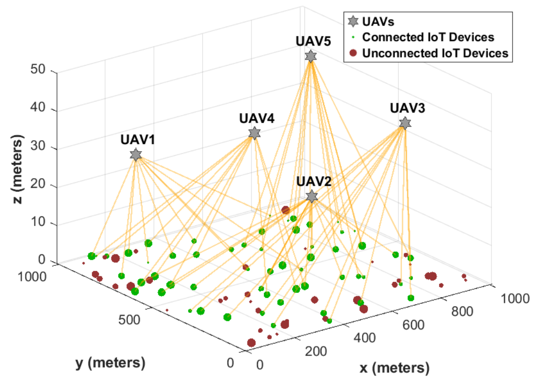
absence of residual energy perturbation, where marker sizes encode quantized task priority levels. BMK-RD and BMK-ID serve as the optimal reference configurations, achieving 62% and 77% connectivity in Fig. 5.2(a) and Fig. 5.2(b), respectively. PGO-RD achieves 61% connectivity in Fig. 5.2(c), attaining a negligible gap of 1% relative to BMK-RD. PGO-ID achieves 70% in Fig. 5.2(d), where the 7% deviation relative to BMK-ID reflects the increased combinatorial complexity of priority-weighted K-means clustering, which favors the exact branch-and-bound search of BMK-ID over the continuous relaxation of PGO-ID. GDY-RD and GDY-ID achieve 45% and 55% in Fig. 5.2(e) and Fig. 5.2(f), respectively. The sequential device selection of the GDY algorithm fails to jointly account for the interdependencies among trust eligibility, resource feasibility, and priority-weighted admission criteria, yielding systematically inferior association decisions relative to both BMK and PGO configurations.

Fig. 5.2(g) and Fig. 5.2(h) correspond to BRE-RD and BRE-ID, which yield the lowest connectivity at 39% and 38%, respectively. BRE-ID fails to produce any appreciable improvement over BRE-RD despite the transition from random to intelligent association decisions. The absence of integrality enforcement causes the interior-point solver to produce fractional association variables that violate the admittance-association coupling of C1 upon rounding. In direct contrast, PGO-RD and PGO-ID employ the penalty term Γ to progressively enforce integrality within the optimization process itself. This preserves the joint association and admission structure, yielding consistent connectivity gains across both association decision strategies.

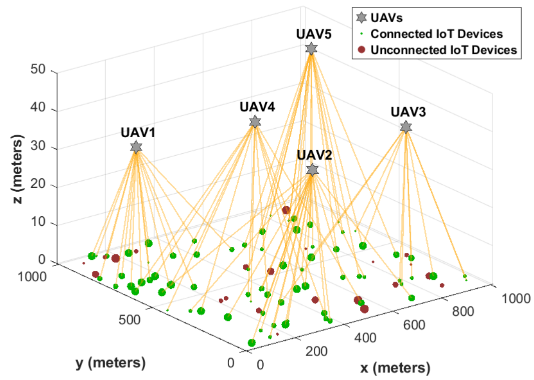
Fig. 5.3 examines trust efficiency across three complementary dimensions under equal utility weights and $E_n^{\mathrm{PER}} = 20\%$. In Fig. 5.3(a), trust efficiency increases consistently with the number of serving UAVs, with BMK-ID and BMK-RD establishing the upper performance bound throughout. PGO-ID and PGO-RD maintain a near-optimal approximation of this bound across all UAV counts, confirming the scalability of the proposed algorithm under varying network conditions. In contrast, GDY-ID and GDY-RD exhibit a minor advantage at lower UAV counts. This reverses at M=5 and M=6 where GDY-RD outperforms GDY-ID as intelligent sequential selection struggles to scale with the expanded association space. BRE-ID and BRE-RD yield no trust efficiency at M=2 and improve only marginally from M=3 onwards, remaining the weakest performers across all UAV counts. Fig. 5.3(b) reveals a consistent improvement across task priority levels, where PGO-ID and PGO-RD closely track the BMK-ID and BMK-RD bound across all five levels, respectively. GDY-ID degrades below GDY-RD at priority level 5, suggesting that intelligent association over-commits to high-priority devices under constrained admission conditions.

Fig. 5.3(c) analyzes trust efficiency against $T_n^{\mathrm{THR}} \in [0.5, 0.9]$ for M=6 and $\psi_{nm} = 5$. Trust efficiency declines consistently across all configurations as stricter thresholds tighten C4, reducing feasible associations and degrading joint service reliability. PGO-ID and PGO-RD closely track the BMK-ID bound across the entire threshold range, demonstrating resilience under increasingly stringent trust requirements. GDY-RD outperforms GDY-ID by an appreciable margin across intermediate thresholds with a representative difference of approximately 0.07. This advantage narrows considerably at $T_n^{\mathrm{THR}} = 0.9$ where maximum trust stringency constrains both configurations equally. BRE-ID and BRE-RD follow a parallel declining trend and remain the weakest performers across all thresholds, confirming that integrality handling rather than trust filtering governs reliable service delivery in the proposed framework.

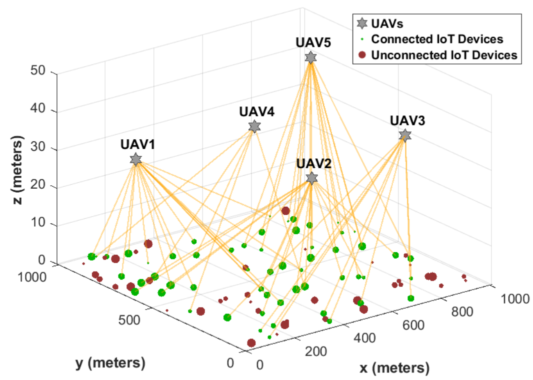
Fig. 5.4 examines the price of robustness under four contrasting resource and demand



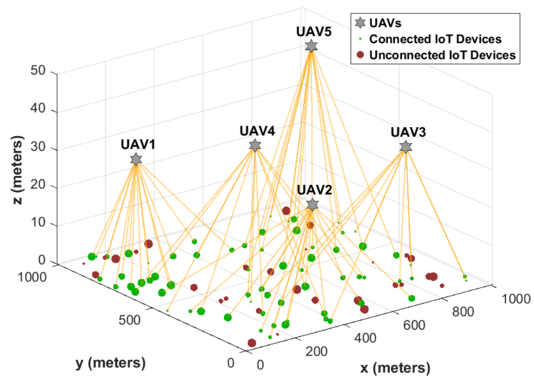
(a) BMK-RD: 62%



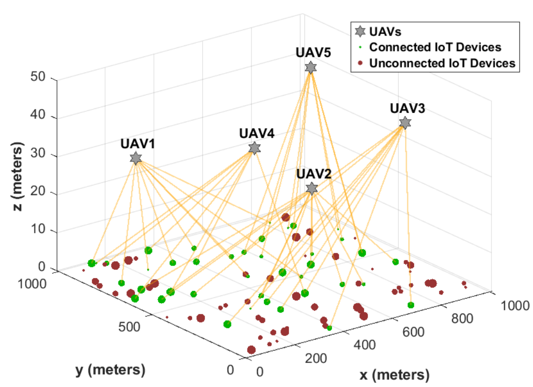
(b) BMK-ID: 77%



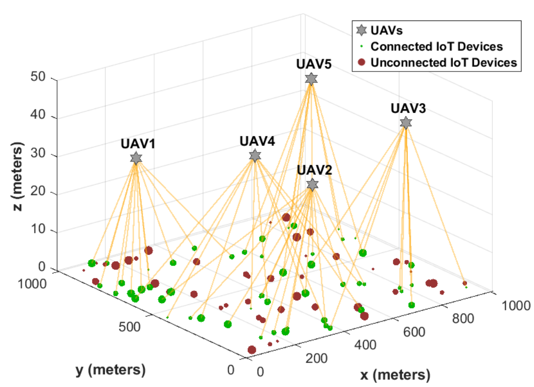
(c) PGO-RD: 61%



(d) PGO-ID: 70%

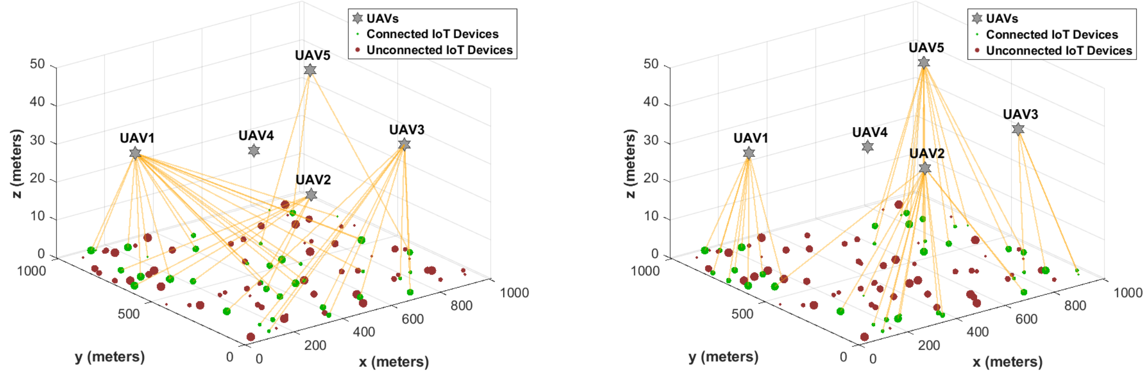


(e) GDY-RD: 45%



(f) GDY-ID: 55%

Figure 5.2: The association of $N = 100$ IoT devices requiring offloading, under random and intelligent associations with $M = 5$ UAVs, with equally weighted utility parameters (all $\nu = 0.25$), trust threshold $T_n^{\text{THR}} = 0.50$, residual energy perturbations $E_n^{\text{PER}} = 0\%$, energy threshold $E_n^{\text{THR}} = 0.20$ and quantized task priority levels $\psi_{nm} = 5$.

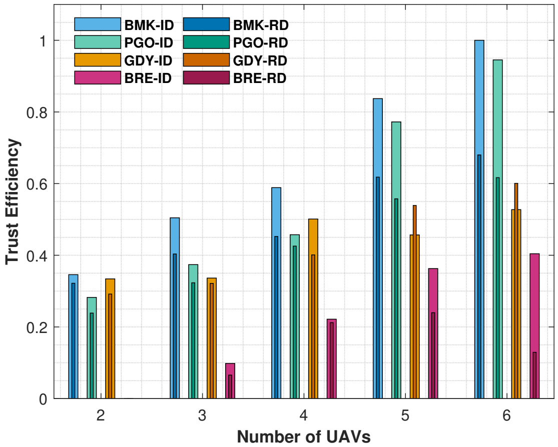


(g) BRE-RD: 39%   (h) BRE-ID: 38%

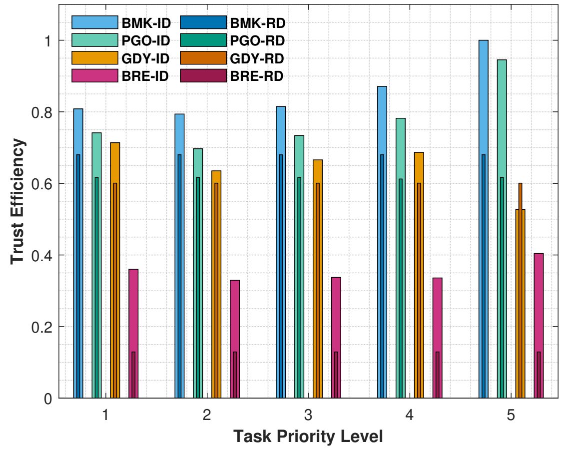
Figure 5.2: (Continued) The association of $N = 100$ IoT devices under baseline relaxation algorithm with the same parameter settings.

scenarios, where disconnections are measured relative to the respective optimal bound at zero perturbation. Fig. 5.4(a) considers low resources (M=2) and low demand ($T_n^{\mathrm{THR}} = 0.5$, $\psi_{nm} = 1$), where PGO-ID maintains a tight approximation to BMK-ID with approximately 3% additional disconnections across the entire perturbation range. GDY-ID and GDY-RD exhibit significantly higher disconnections averaging approximately 15% above PGO-ID. BRE-ID and BRE-RD reach approximately 45% more disconnections than PGO-ID, confirming the fundamental limitation of relaxation without integrality enforcement under constrained network conditions. Fig. 5.4(b) extends the analysis to high resources (M=6) under the same low demand conditions. The PGO-ID to BMK-ID gap widens marginally to approximately 6% on average, reflecting the increased combinatorial complexity introduced by the expanded UAV deployment. GDY-RD exhibits more disconnections than GDY-ID throughout, averaging approximately 12% above PGO-ID. BRE-ID peaks at 30 disconnections at zero perturbation and exhibits non-monotonic behavior across the perturbation range. BRE-RD remains consistently worse than BRE-ID by approximately 15%, reflecting the instability of post-hoc rounding under varying energy uncertainty.

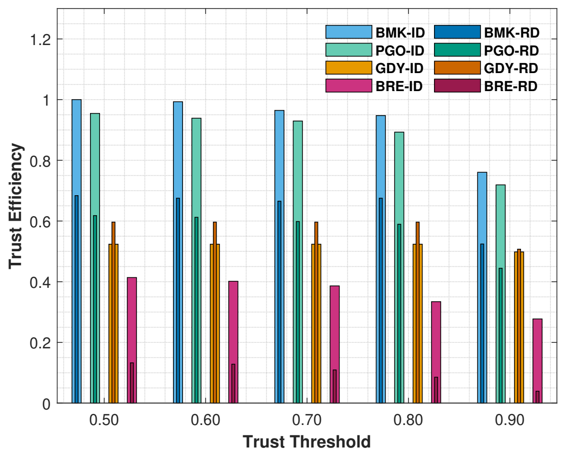
Fig. 5.4(c) and Fig. 5.4(d) introduce high demand conditions ($T_n^{\mathrm{THR}} = 0.9$, $\psi_{nm} = 5$), revealing critical behavioral dimensions across all evaluated algorithms. In Fig. 5.4(c), under



(a) $N = 100$, $T_n^{\text{THR}} = 0.50$, $E_n^{\text{PER}} = 20\%$, $E_n^{\text{THR}} = 0.20$, and $\psi_{nm} = 5$



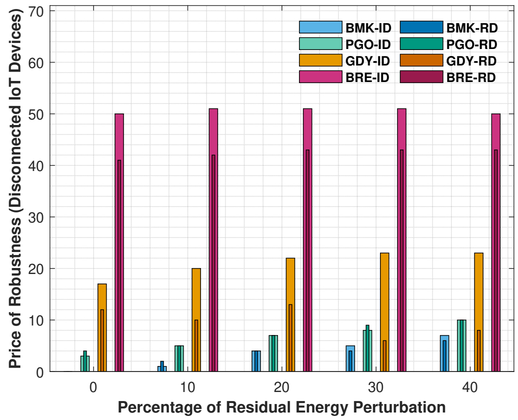
(b) $N = 100$, $M = 6$, $T_n^{\text{THR}} = 0.50$, $E_n^{\text{PER}} = 20\%$, and $E_n^{\text{THR}} = 0.20$



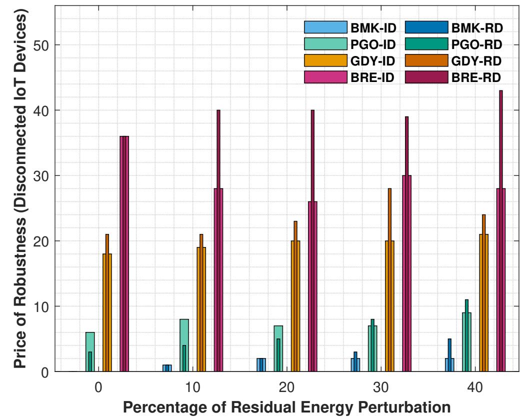
(c) $N = 100$, $M = 6$, $E_n^{\text{PER}} = 20\%$, $E_n^{\text{THR}} = 0.20$, and $\psi_{nm} = 5$

Figure 5.3: Trust efficiency versus UAVs, task priority levels, and IoT trust thresholds for equally weighted utility parameters (all $\nu = 0.25$).

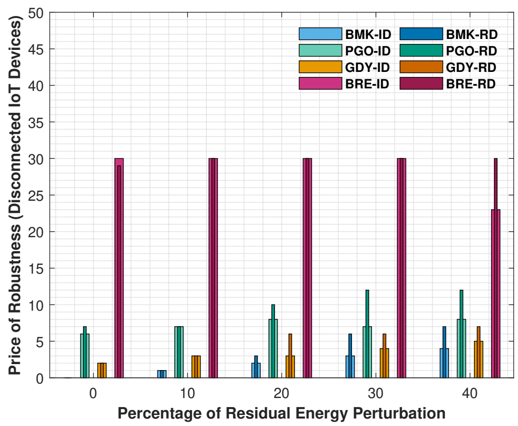
low resources (M=2), GDY-ID and GDY-RD exhibit a marginal advantage over PGO-ID of approximately 3–5%. PGO-ID nonetheless maintains a 4-6% proximity to BMK-ID across the perturbation range, confirming that this marginal deviation does not compromise the overall robustness of the proposed algorithm under severe resource and trust constraints. BRE-ID exhibits a gradual decline from approximately 30% above BMK-ID at zero perturbation to 20% at maximum perturbation, while BRE-RD remains the weakest performer throughout. Fig. 5.4(d) considers high resources (M=6) under the same high demand conditions, where PGO-ID sustains a strong approximation to BMK-ID with a 5-7% gap across



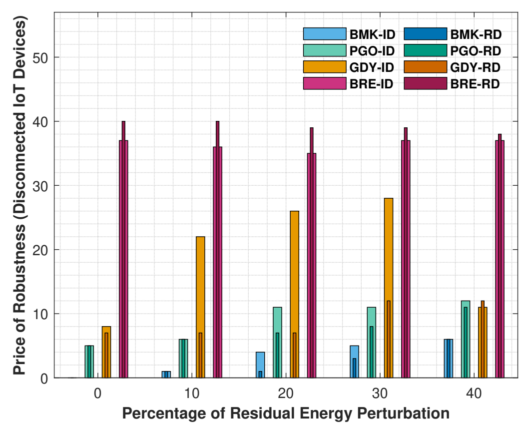
(a) Low resources: $M = 2$, low demand: $N = 100$, $T_n^{\mathrm{THR}} = 0.5$, $E_n^{\mathrm{THR}} = 0.20$, quantized $\psi_{nm} = 1$.



(b) High resources: $M = 6$, low demand: $N = 100$, $T_n^{\mathrm{THR}} = 0.5$, $E_n^{\mathrm{THR}} = 0.20$, quantized $\psi_{nm} = 1$.



(c) Low resources: $M = 2$, high demand: $N = 100$, $T_n^{\mathrm{THR}} = 0.9$, $E_n^{\mathrm{THR}} = 0.20$, quantized $\psi_{nm} = 5$.



(d) High resources: $M = 6$, high demand: $N = 100$, $T_n^{\mathrm{THR}} = 0.9$, $E_n^{\mathrm{THR}} = 0.20$, quantized $\psi_{nm} = 5$.

Figure 5.4: Price of robustness versus residual energy perturbations for equally weighted utility parameters (all $\nu = 0.25$).

all perturbation levels. GDY-ID exhibits non-monotonic behavior, rising sharply from 3% above PGO-ID at zero perturbation to approximately 15-17% at moderate perturbation levels before converging to within 1% at maximum perturbation. BRE-ID and BRE-RD remain the weakest performers, with BRE-RD consistently exceeding BRE-ID disconnections by 1–4% throughout, further confirming that integrality enforcement remains the decisive factor in maintaining reliable admission performance under increasing energy uncertainty.

Fig. 5.5 evaluates the number of connected IoT devices across task priority levels under N=100, M=6, $T_n^{\mathrm{THR}} = 0.6$, and $E_n^{\mathrm{PER}} = 0\%$. The weight profiles in Table 5.3 represent illustrative instances of a flexible design space, enabling network operators to continuously steer the optimization emphasis according to their specific deployment context and service requirements. The framework naturally accommodates a wide range of emphasis configurations, from balanced multi-criteria evaluation to strongly criterion-specific optimization. The non-uniform device distribution across priority levels (16, 22, 18, 20, 24) introduces varying admission competition at each level. The simultaneous enforcement of trust, resource, and energy constraints shapes the achievable connectivity under each weight configuration.

Fig. 5.5(a) presents the equal weight configuration, where PGO-ID maintains near-optimal connectivity relative to BMK-ID across all priority levels, with deviations of at most 1 device at the most competitive levels. Priority level 3 yields the highest proportion of connected devices relative to available devices, with BMK-ID and PGO-ID each connecting 15 out of 18. Priority level 5 presents the highest device density among all levels, where PGO-ID consistently maintains 17 connected devices across equal, trust, and priority emphasis profiles. GDY-ID maintains reasonable connectivity at lower priority levels but deteriorates at levels 4 and 5, connecting 9 and 11 devices against PGO-ID's 13 and 17. This reflects the inability of sequential selection to jointly account for trust eligibility, resource feasibility, and priority-weighted admission interdependencies at higher priority levels. BRE-ID and BRE-RD deliver the weakest admission performance across all priority levels. The absence of integrality enforcement consistently undermines their admission decisions regardless of the priority level. Fig. 5.5(b) and Fig. 5.5(c) reinforce this finding, with PGO-ID sustaining near-optimal per-level connectivity across all priority levels.

Fig. 5.5(d) examines the cost emphasis configuration, where the elevated ν^{CST} more strongly penalizes higher cost associations. GDY-ID connects 2 out of 16 available devices at priority level 1, while PGO-ID maintains 10 under the same conditions. This deviation reflects the fragility of sequential selection under cost-dominated utility landscapes. The penalty term Γ preserves the joint admission structure of PGO-ID and achieves an overall connectivity of 69%, a negligible deviation from its equal weight performance. Fig. 5.5(e)

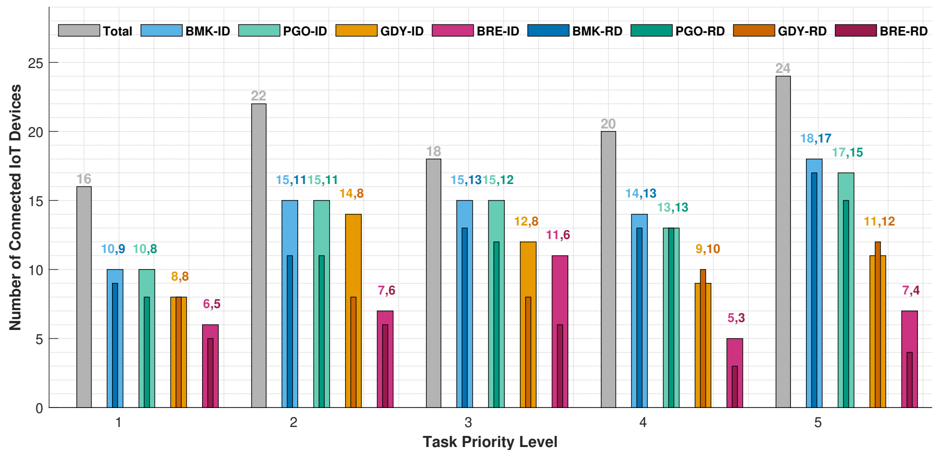
Table 5.4: Overall connectivity (%) across utility weight profiles for $N = 100$, $M = 6$, $T_n^{\mathrm{THR}} = 0.6$, and $E_n^{\mathrm{PER}} = 0\%$, where (a) Equal, (b) Trust, (c) Priority, (d) Cost, and (e) Latency correspond to the weight emphasis profiles in Table 5.3.

Algorithm	**(a)**	**(b)**	**(c)**	**(d)**	**(e)**	**Range**
BMK-ID	72	72	72	72	72	0
BMK-RD	63	63	63	62	62	1
PGO-ID	70	70	70	69	70	1
PGO-RD	59	58	59	59	59	1
GDY-ID	54	48	54	26	62	36
GDY-RD	46	44	46	47	47	3
BRE-ID	36	38	36	36	36	2
BRE-RD	24	25	24	25	25	1

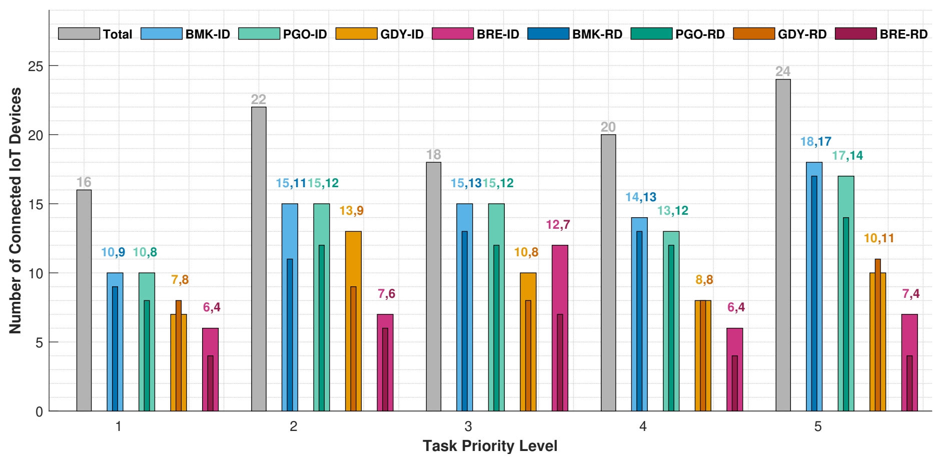
presents a complementary outcome under latency emphasis, where the elevated ν^{LAT} naturally favors geographically proximate device-UAV pairs. This alignment with the K-means deployment topology enables GDY-ID to achieve its strongest overall performance of 62% connected devices. PGO-ID maintains near-optimal 70% connectivity under latency emphasis, affirming its robustness across all weight configurations examined. BRE-ID and BRE-RD remain the weakest performers across both profiles, confirming that weight profile variations cannot compensate for the absence of integrality enforcement.

Table 5.4 summarizes the overall connectivity across all weight profiles for Fig. 5.5, with the range column capturing each algorithm's sensitivity to weight profile variations. PGO-ID achieves 69–70% connectivity across all profiles with a negligible range of 1%, closely tracking BMK-ID's 72% connectivity. GDY-ID exhibits the highest sensitivity at 36%, while BRE-ID and BRE-RD register near-zero ranges reflecting utility insensitivity rather than robustness. The proposed PGO maintains near-optimal admission performance across all weight profiles, confirming its suitability for trustworthy and prioritized aerial edge computing.

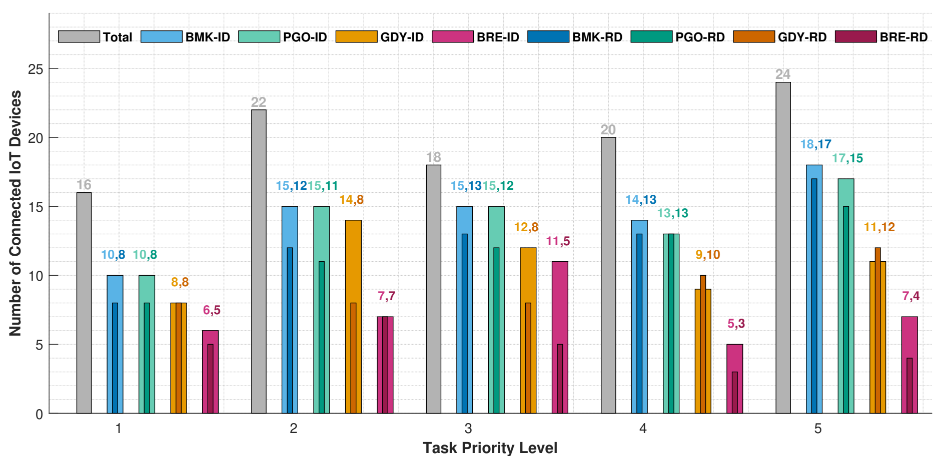
Fig. 5.6(a) presents the achieved utility across the five weight profiles of Table 5.3, ordered as equal, trust, latency, cost, and priority. BMK-ID establishes the optimal upper bound throughout and reaches its highest value under trust emphasis. This peak reflects the higher



(a) Utility parameters weight emphasis: Equal

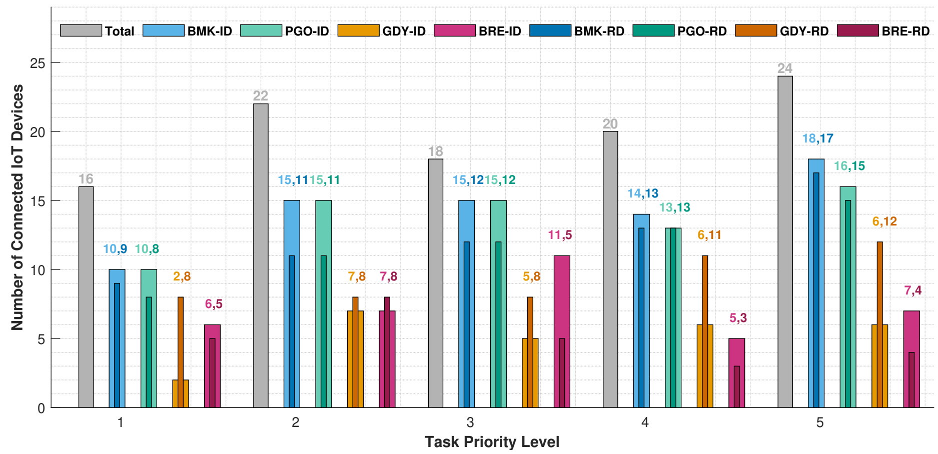


(b) Utility parameters weight emphasis: Trust

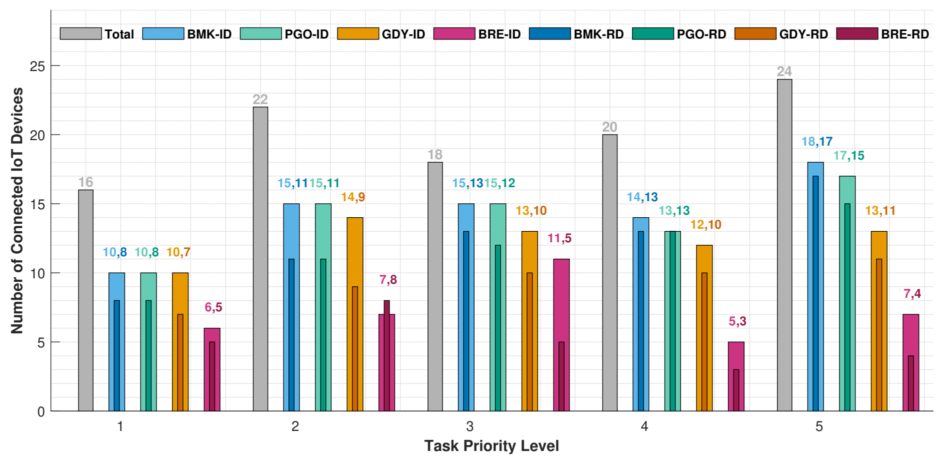


(c) Utility parameters weight emphasis: Priority

Figure 5.5: Number of connected IoT devices versus priority levels, for $N = 100$ IoT devices and $M = 6$ UAVs, having $E_n^{\mathrm{PER}} = 0\%$, $E_n^{\mathrm{THR}} = 0.20$, and $T_n^{\mathrm{THR}} = 0.60$.



(d) Utility parameters weight emphasis: Cost

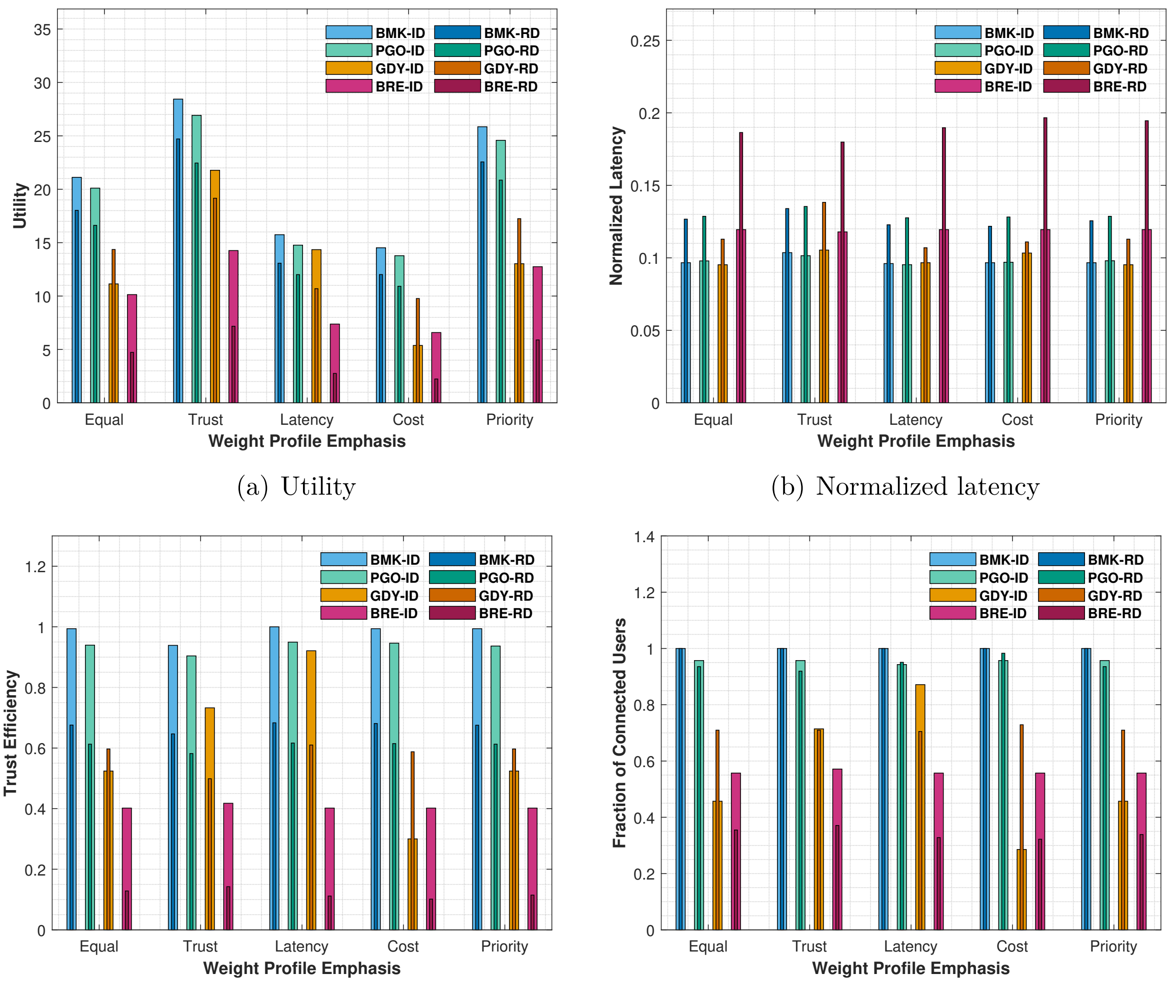


(e) Utility parameters weight emphasis: Latency

Figure 5.5: (Continued) Number of connected IoT devices versus priority levels under cost and latency weight emphasis.

weighting of trust scores within $\mathcal{H}_{nm}$, which contributes more to the aggregate utility than cost or latency penalties. PGO-ID closely approximates this bound across all profiles, trailing BMK-ID by only 1-2 units and confirming that its near-optimal connectivity extends to the underlying utility maximization. GDY-RD outperforms GDY-ID under equal, cost, and priority emphasis. Latency emphasis brings both GDY-ID and GDY-RD closer to PGO-ID and PGO-RD than in any other profile. BRE-ID and BRE-RD remain the weakest performers across all profiles, consistent with the integrality limitations established earlier.

Fig. 5.6(b) shows that RD configurations consistently incur higher normalized latency than ID configurations across all algorithms. BMK-ID achieves the lowest latency throughout



(a) Utility

(b) Normalized latency

(c) Trust efficiency

(d) Fraction connected users

Figure 5.6: The performance of algorithms for different weight profiles as described in Table 5.3, for $N = 100$, $M = 6$, $T_n^{\mathrm{THR}} = 0.6$, $E_n^{\mathrm{PER}} = 20\%$, $E_n^{\mathrm{THR}} = 0.2$, and quantized priority levels $\psi_{nm} = 5$.

and PGO-ID closely tracks this lower bound across all profiles. BRE-ID and BRE-RD reach the highest normalized latency among ID and RD configurations, respectively. Fig. 5.6(c) and Fig. 5.6(d) show similar trends as established in Fig. 5.6(a). PGO-ID and PGO-RD achieve near-optimal trust efficiency and connectivity across all profiles. GDY-ID reaches its weakest trust efficiency and connectivity under cost emphasis, mirroring its earlier collapse in utility. BRE-RD remains the weakest performer across both metrics for all RD configurations. These results confirm that integrality enforcement ultimately governs reliable performance across every dimension of the optimization framework.

5.5 Summary

This chapter concludes that reliable aerial service delivery in disaster-affected and resource-constrained environments requires IoT admission decisions that respect trust, priority, and energy limitations simultaneously. These criteria interact directly within the admission process, where strengthening trust eligibility can reduce the pool of admissible devices and reshape priority, cost, and latency outcomes in turn. The proposed PGO algorithm meets these joint requirements, achieving connectivity and trust efficiency close to the optimal benchmark at significantly lower computational cost. This efficiency matters directly in 6G-enabled aerial networks, where realistic service requirements demand timely admission and MEC service delivery. The benchmark establishes the best achievable performance, but its computational demands become infeasible under the resource-constrained conditions typical of such deployments. The proposed PGO algorithm also remains more reliable than relaxation-based and heuristic methods as network conditions grow more demanding. This suggests that the rounding losses typical of relaxation-based approaches, rather than the multi-criteria nature of the problem itself, present the greater obstacle to dependable aerial service delivery.

Chapter 6

Trust-Driven Aerial Edge Computing for Dynamic IoT Networks across Control Architectures

6.1 Introduction

Trust-aware IoT-UAV association in aerial edge computing determines which UAV serves each IoT device, with a supervisory layer evaluating the UAV population against the requirements devices declare [101, 102]. The resulting formulations treat IoT device positions as fixed, so a single set of IoT-UAV assignments serves the whole deployment. Disaster response breaks this assumption, since responders and vehicles carry IoT devices across the service area while operations continue [94]. The authors of [25] addressed such conditions by repositioning UAVs under high-density mobility, reporting that existing studies concentrate on static or low-mobility devices. The authors of [26] and [66] adapted the offloading instead, migrating tasks between UAVs or coupling offloading decisions with trajectory design.

The effectiveness of aerial edge computing depends on the capability of the serving UAV, and is tied to the trust threshold each IoT device declares [32,95,103]. A device that satisfied its threshold at deployment may travel beyond the coverage of the UAV serving it, while the UAVs still within range fall short of what it requires. The aerial network must therefore adopt a repositioning mechanism for the available UAVs as the IoT population redistributes, at a cost in energy and service interruption. Two-timescale designs bound that cost, resolving allocation on the faster scale and deployment on the slower one [68, 104].

The control architecture determines whether UAV repositioning and resource allocation proceed from a central entity or from the UAVs themselves [29, 105, 106]. The authors of [27] compared a centralized scheme against a semi-distributed federated alternative, reporting similar results at lower operation time. The authors of [28] restricted each agent to locally available information, attaining performance close to centralized schemes while scaling with the number of UAVs. The authors of [30] found that centralization offers only conditional benefit, and that state-based critics introduce bias under partial observability.

Fully distributed operation removes the central entity, so UAVs establish a shared view through consensus among themselves [107]. The authors of [108] applied consensus theory to joint trajectory and resource optimization, examining convergence under partial and full connectivity. The authors of [109] recast functionality switching as a distributed consensus problem solved through alternating direction methods. The comparisons across architectures reported reward, latency, and energy consumption. However, none evaluated the trustworthiness a trained policy sustains. This chapter extends trust-aware IoT-UAV association to mobile IoT populations and to control architectures spanning centralized, hybrid (centralized-training, decentralized-execution), and fully distributed operation [99].

6.2 System Model and Problem Formulation

Fig. 6.1 illustrates the system model, where a fleet of UAVs provides trust-aware computational support to ground-based IoT devices with varying trust requirements across a disaster-impacted region lacking terrestrial coverage. The air-to-ground (A2G) links between IoT

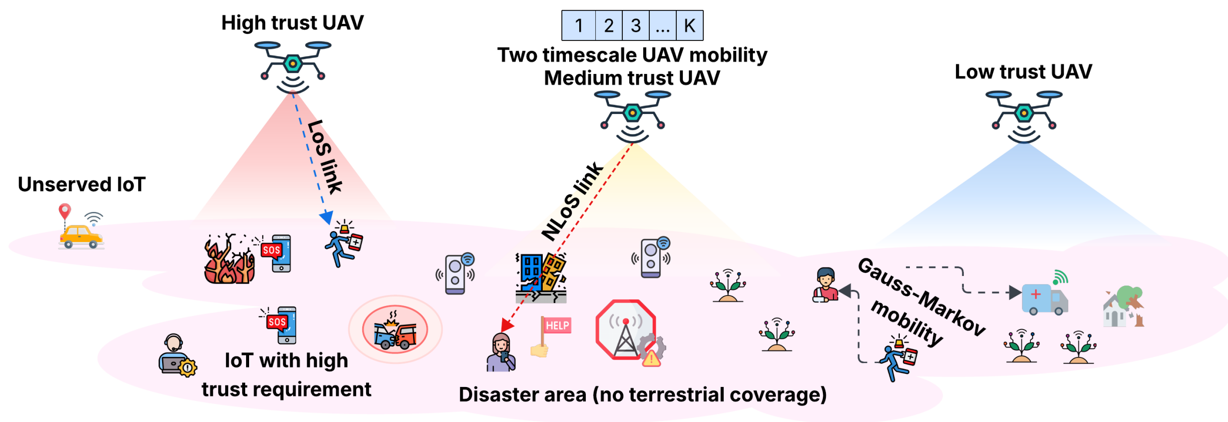


Figure 6.1: Trust-aware aerial network serving IoT devices under varying trust thresholds, channel, and mobility.

devices and their serving UAVs switch between LoS and non-LoS (NLoS) propagation as obstacles, debris, and structural damage vary across the coverage area. The IoT device positions evolve according to a stochastic mobility pattern that reflects the unpredictable movement of survivors and responders. The UAVs adopt a two-timescale positioning strategy to address this channel and mobility volatility, separating fine-grained tracking from coarse-grained repositioning. The minor positional adjustments of UAVs continuously follow nearby devices at the finer timescale. In contrast, UAV reassignments occur at a coarser timescale, triggered whenever a persistent mismatch emerges between a UAV's demonstrated trustworthiness and the trust demands of its associated devices. This mismatch directs repositioning toward the device clusters best suited to each UAV's operational capacity, coupling spatial positioning directly to the trust dynamics.

6.2.1 Network Model

The trust-aware UAV-assisted MEC network operates over a circular disaster-affected coverage area $\mathcal{A}_C$, where UAVs and IoT devices interact in the absence of terrestrial infrastructure. The individual operational episodes span K discrete time slots. UAV $m \in \{1, \ldots, M\}$ serves as an airborne MEC node at position $O_m(k) = (x_m(k), y_m(k), z_m)$ with fixed altitude z_m. The *m-th* UAV tracks its cluster centroid every K_u time slots, making minor lateral adjustments to follow the current IoT device distribution, yielding up to $\lfloor K/K_u \rfloor$ such adjustments per episode. The inter-cluster repositioning occurs at the coarser timescale $K_g = \{GK_u, 2GK_u, \ldots, \lfloor K/(GK_u) \rfloor \cdot GK_u\}$, with $G > 1$ denoting the number of status-update cycles per repositioning event [68]. Task offloading pauses only during these guard slots, as repositioning alters coverage footprints $\mathcal{C}_m(k)$ and channel conditions before new associations stabilize. The IoT device $n \in \{1, \ldots, N\}$ occupies ground position $O_n(k) = (x_n(k), y_n(k), 0)$ at time slot k. It generates the computational tasks according to a Bernoulli process with parameter p^{GEN}, where $a_n(k) \in \{0, 1\}$ represents the task arrival indicator [110].

The safety-critical nature of disaster scenarios necessitates that the *n-th* IoT device maintain a trust threshold $T_n^{\text{THR}} \in [0.1, 0.9]$, quantifying the minimum UAV reliability acceptable for task offloading [101]. The serving area of the *m-th* UAV spans a circular ground footprint of radius $r_m = \frac{z_m}{\tan(\theta)}$, where θ denotes the minimum elevation angle for reliable A2G communication [111]. The IoT-UAV association holds only when both the coverage constraint $d_{nm}^h(k) = \sqrt{(x_n(k)-x_m(k))^2+(y_n(k)-y_m(k))^2} \leq r_m$ and the trust requirement $T_{nm}(k) \geq T_n^{\text{THR}}$ are jointly satisfied. The IoT devices submit at most one offloading request per time slot k. The three-dimensional separation between the *n-th* IoT device and the *m-th* UAV at time slot k is expressed as:

$$d_{nm}(k) = \sqrt{(x_n(k) - x_m(k))^2 + (y_n(k) - y_m(k))^2 + z_m^2} \tag{6.1}$$

The position of the *n-th* IoT device ($O_n(k)$) evolves across time slots according to the Gauss-Markov mobility model that captures the heterogeneous movement patterns. The velocity $v_n(k)$ and direction $\theta_n(k)$ of the IoT device update at the *k-th* time slot as [67]:

$$v_n(k) = \omega_n v_n(k-1) + (1 - \omega_n)\bar{v}_n + \sqrt{1 - \omega_n^2}\, \Phi_n(k), \tag{6.2}$$

$$\theta_n(k) = \omega_n \theta_n(k-1) + (1 - \omega_n)\bar{\theta}_n + \sqrt{1 - \omega_n^2}\, \Psi_n(k), \tag{6.3}$$

where $\omega_n \in (0,1)$ governs per-device trajectory smoothness, $\bar{v}_n$ and $\bar{\theta}_n$ are the mean speed and direction, $\Phi_n(k) \sim \mathcal{N}(0,1)$ and $\Psi_n(k) \sim \mathcal{N}(0,1)$ are independent Gaussian perturbations on speed and direction respectively. The IoT positions update per time slot as $x_n(k) = x_n(k-1) + v_n(k)\cos\theta_n(k)$ and $y_n(k) = y_n(k-1) + v_n(k)\sin\theta_n(k)$, with movement bounded so that each device remains within the coverage of some UAV throughout the episode. This bound does not fix a device to one serving UAV. Its nearest-cluster assignment, and therefore its serving UAV, may change as it moves.

6.2.2 Channel and Communication Model

The A2G link follows a probabilistic path loss model accounting for LoS and NLoS propagation conditions. The LoS probability depends on the elevation angle $\phi_{nm}(k) = \arctan(z_m/d^h_{nm}(k))$ and on environment-specific constants a and b characterizing the propagation environment, as $P^{\mathrm{LoS}}_{nm}(k) = \frac{1}{1+a\exp(-b(\phi_{nm}(k)-a))}$, with $P^{\mathrm{NLoS}}_{nm}(k) = 1 - P^{\mathrm{LoS}}_{nm}(k)$ [112]. The realized propagation state $\xi_{nm}(k) \in \{\mathrm{LoS}, \mathrm{NLoS}\}$ is drawn independently at each slot k according to $P^{\mathrm{LoS}}_{nm}(k)$, capturing the joint effect of stochastic propagation and time-varying device and UAV positions. The path loss under propagation mode $\xi \in \{\mathrm{LoS}, \mathrm{NLoS}\}$ is expressed as:

$$PL^{\xi}_{nm}(k) = 20\log(d_{nm}(k)) + 20\log(f_c) + 20\log\left(\tfrac{4\pi}{c}\right) + \eta_{\xi}, \tag{6.4}$$

where f_c denotes the carrier frequency, c the speed of light, and η_ξ the excess attenuation factor. The realized path loss follows the drawn propagation state as $PL_{nm}(k) = PL^{\xi_{nm}(k)}_{nm}$, yielding the channel gain $g_{nm}(k) = 10^{-PL_{nm}(k)/10}$ [113]. We employ OFDMA to assign a dedicated subchannel of bandwidth B to each IoT device, supporting the uplink transmission rate:

$$R_{nm}(k) = B\log_2\left(1 + \frac{p_n\, g_{nm}(k)}{\sigma^2}\right) \tag{6.5}$$

where p_n and σ^2 denote the transmit power and noise power.

6.2.3 Task and Computation Model

The *n-th* IoT device generates a task at time slot k characterized by the profile $\mathcal{F}_n(k) = \{D_n(k), \delta_n(k), L_n^{\mathrm{MAX}}\}$, where $D_n(k)$ denotes the task data size in bits, $\delta_n(k)$ the required CPU cycles per bit, and L_n^{MAX} the maximum tolerable latency. The binary offloading decision $\alpha_{nm}(k) \in \{0, 1\}$ determines whether this task is offloaded to an associated *m-th* UAV or computed locally, with $\alpha_{nm}(k) = 1$ only admissible when $a_n(k) = 1$.

• Local Computation: The *n-th* IoT device computes the task locally at CPU frequency f_n, incurring latency $L_n^{\mathrm{LOC}}(k) = \frac{D_n(k) \cdot \delta_n(k)}{f_n}$.

• Edge Computation: The *n-th* IoT device transmits the task to an associated *m-th* UAV over the uplink channel, incurring transmission delay $L_{nm}^{\mathrm{TRA}}(k) = \frac{D_n(k)}{R_{nm}(k)}$. The *m-th* UAV allocates CPU frequency $f_{nm}(k)$ to execute the task, incurring processing delay $L_{nm}^{\mathrm{EXE}}(k) = \frac{D_n(k) \cdot \delta_n(k)}{f_{nm}(k)}$. The UAVs process tasks from multiple associated devices in parallel within each time slot, with the total allocated CPU frequency bounded by F_m^{MAX}. The total edge computation latency is $L_{nm}^{\mathrm{OFF}}(k) = L_{nm}^{\mathrm{TRA}}(k) + L_{nm}^{\mathrm{EXE}}(k)$.

• End-to-End Latency: The offloading decision $\alpha_{nm}(k)$ determines the experienced latency as:

$$L_{nm}(k) = (1 - \alpha_{nm}(k))\, L_n^{\mathrm{LOC}}(k) + \alpha_{nm}(k) L_{nm}^{\mathrm{OFF}}(k). \tag{6.6}$$

The computational result returned by the *m-th* UAV is typically orders of magnitude smaller than the input task size $D_n(k)$, rendering the downlink transmission delay and associated energy consumption negligible [114].

6.2.4 Energy Consumption Model

Task offloading from the *n-th* IoT device to the *m-th* UAV incurs energy consumption at both the device and UAV. The *n-th* IoT device expends transmission energy $E_{nm}^{\mathrm{TRA}}(k) = p_n \cdot L_{nm}^{\mathrm{TRA}}(k)$, where p_n denotes its transmit power. The *m-th* UAV incurs execution energy $E_{nm}^{\mathrm{EXE}}(k) = \kappa_m \left(f_{nm}(k)\right)^2 D_n(k)\delta_n(k)$, where κ_m denotes its effective switched capacitance

[115]. The total offloading energy for the IoT-UAV pair is expressed as:

$$E_{nm}(k) = \alpha_{nm}(k) \left(E_{nm}^{\mathrm{TRA}}(k) + E_{nm}^{\mathrm{EXE}}(k) \right). \tag{6.7}$$

This formulation confines transmission and execution energy to instances where the *n-th* IoT device actively offloads its task to the *m-th* UAV. The *m-th* UAV's residual energy $E_m(k)$ depletes from two sources. Propulsion draws a constant per-slot baseline. The optimization controls the remaining dynamic computation and communication components.

6.2.5 Multi-Factor UAV Trustworthiness Model

The trustworthiness of the *m-th* UAV toward the *n-th* IoT device at time slot k incorporates three complementary metrics quantifying energy sustainability, task reliability, and information freshness. The set $\mathcal{N}_m(k) = \{n : \alpha_{nm}(k) = 1\}$ captures all IoT devices actively offloading to the *m-th* UAV at time slot k.

• UAV Energy Metric: The energy trustworthiness metric computes the residual energy ratio of the *m-th* UAV after deducting execution energy commitments to all devices in $\mathcal{N}_m(k)$ at time slot k [41]:

$$T_{nm}^{\mathrm{ENE}}(k) = \frac{E_m(k) - \sum_{n \in \mathcal{N}_m(k)} E_{nm}^{\mathrm{EXE}}(k)}{E_m^{\mathrm{MAX}}}, \tag{6.8}$$

where $E_m(k)$ denotes the residual energy of the *m-th* UAV, E_m^{MAX} its maximum energy capacity, and $T_{nm}^{\mathrm{ENE}}(k)$ remains non-negative through the energy feasibility constraint.

• UAV Task Metric: The task success metric captures the reliability of the *m-th* UAV in completing tasks across all associated devices [116], with $T_{nm}^{\mathrm{TAS}}(1) = 1$ at deployment:

$$T_{nm}^{\mathrm{TAS}}(k) = \frac{\sum_{n \in \mathcal{N}_m(k)} F_{nm}^{\mathrm{SUC}}(k)}{\sum_{n \in \mathcal{N}_m(k)} F_{nm}^{\mathrm{TOT}}(k)}, \tag{6.9}$$

where $F_{nm}^{\mathrm{SUC}}(k)$ and $F_{nm}^{\mathrm{TOT}}(k)$ denote the successfully completed and total tasks between the *n-th* IoT device and the *m-th* UAV up to the *k-th* time slot, respectively.

• UAV Service Freshness Metric: This metric captures the timeliness of the most recent service delivered to the *n-th* IoT device, based on the concept of Age of Information [117]. The service delay state $A_{nm}(k)$ represents the number of time slots elapsed since the last successful update delivered to the *n-th* IoT device by the *m-th* UAV for which $n \in \mathcal{N}_m(k)$. It is expressed as:

$$T_{nm}^{\mathrm{SFR}}(k) = \frac{1}{1 + A_{nm}(k)}. \tag{6.10}$$

The composite trust score for the IoT-UAV pair at the *k-th* time slot aggregates all the underlying metrics. It is expressed as:

$$T_{nm}(k) = w_1 T_{nm}^{\mathrm{ENE}}(k) + w_2 T_{nm}^{\mathrm{TAS}}(k) + w_3 T_{nm}^{\mathrm{SFR}}(k), \tag{6.11}$$

where $w_1, w_2, w_3 \geq 0$ satisfy $w_1 + w_2 + w_3 = 1$, ensuring $T_{nm}(k) \in [0, 1]$. The three metrics $T_{nm}^{\mathrm{ENE}}(k)$, $T_{nm}^{\mathrm{TAS}}(k)$, and $T_{nm}^{\mathrm{SFR}}(k)$ update at each time slot k in response to energy expenditure, task outcomes, and status delivery events, collectively characterizing the time-varying trustworthiness of the *m-th* UAV.

Chapters 3 through 5 recompute trust independently at each association instance. Sensitivity to any single weak dimension therefore strengthens that one evaluation without carrying consequences forward. The metrics in this chapter instead accumulate state across time slots, an aging counter for freshness and a running success ratio for reliability. A single adverse event should not by itself collapse an otherwise reliable trust score. A weighted linear combination absorbs this kind of transient variation while still reflecting sustained weakness in any dimension. The weights w_1, w_2, w_3 also let the comparison across control architectures tune emphasis among energy, reliability, and freshness as conditions evolve.

6.2.6 Problem Formulation

We formulate the joint optimization of task offloading decisions and UAV computation resource allocation through the multi-criteria utility function [14]. It is expressed as $\mathcal{U}_{nm}(k) = \nu^{\mathrm{LAT}}\tilde{L}_{nm}(k) + \nu^{\mathrm{ENE}}\tilde{E}_{nm}(k) - \nu^{\mathrm{TRS}}T_{nm}(k)$, where $\tilde{L}_{nm}(k) = L_{nm}(k)/L_n^{\mathrm{MAX}}$ and $\tilde{E}_{nm}(k) = E_{nm}(k)/E_m^{\mathrm{MAX}}$ normalize latency and energy to [0,1], consistent with $T_{nm}(k) \in [0,1]$, and $\nu^{\mathrm{LAT}}, \nu^{\mathrm{ENE}}, \nu^{\mathrm{TRS}} \geq 0$ satisfy $\nu^{\mathrm{LAT}} + \nu^{\mathrm{ENE}} + \nu^{\mathrm{TRS}} = 1$. The optimization problem ($\mathcal{P}$) is formulated as:

$$\mathcal{P}: \min_{\{O_m(k),\alpha_{nm}(k),f_{nm}(k)\}} \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{m=1}^{M} \alpha_{nm}(k)\mathcal{U}_{nm}(k) \tag{6.12}$$

Subject to:

$$C1: \quad \sum_{m=1}^{M} \alpha_{nm}(k) \leq a_n(k), \qquad \forall n,k \tag{6.12a}$$

$$C2: \quad T_{nm}(k) \geq T_n^{\mathrm{THR}} - \mathbb{M}(1-\alpha_{nm}(k)), \forall n,m,k \tag{6.12b}$$

$$C3: \quad L_{nm}(k) \leq L_n^{\mathrm{MAX}}, \qquad \forall n,k \tag{6.12c}$$

$$C4: \quad \sum_{n=1}^{N} \alpha_{nm}(k) f_{nm}(k) \leq F_m^{\mathrm{MAX}}, \qquad \forall m,k \tag{6.12d}$$

$$C5: \quad E_m(k) - \sum_{n\in\mathcal{N}_m(k)} E_{nm}^{\mathrm{EXE}}(k) \geq E_m^{\mathrm{MIN}}, \ \forall m,k \tag{6.12e}$$

$$C6: \quad d_{nm}^{h}(k) \leq r_m, \qquad \forall n,m,k \tag{6.12f}$$

$$C7: \quad \alpha_{nm}(k) = 0, \qquad \forall n,m,k \in K_g \tag{6.12g}$$

$$C8: \quad \alpha_{nm}(k) \in \{0,1\}, \quad f_{nm}(k) \geq 0, \qquad \forall n,m,k \tag{6.12h}$$

where $\mathbb{M}$ denotes a sufficiently large constant. Constraint C1 restricts each IoT device to associate with at most one UAV per time slot. C2 enforces the trust gate through a Big-M formulation [15], ensuring offloading proceeds only when the *m-th* UAV satisfies the trust threshold T_n^{THR} of the *n-th* IoT device. C3 enforces that latency $L_{nm}(k)$ remains within the maximum tolerable deadline L_n^{MAX}. C4 restricts total computation resources allocated by the *m-th* UAV to its maximum processing capacity. C5 maintains the residual energy of the *m-th* UAV above the minimum operational threshold E_m^{MIN}, ensuring $T_{nm}^{\mathrm{ENE}}(k)$ remains non-negative at all time slots. C6 enforces the coverage feasibility constraint, ensuring the *n-th* IoT device lies within the coverage footprint of the *m-th* UAV at each time slot. C7 suspends task offloading during guard slots K_g to avoid trust metric corruption from unstable channel conditions during UAV repositioning. C8 enforces binary offloading decisions and non-negative computation allocation.

The formulated optimization problem involves binary decision variables $\alpha_{nm}(k)$ coupled with continuous resource allocation $f_{nm}(k)$ over a trust-varying, energy-depleting, and channel-dynamic environment. The problem ($\mathcal{P}$) is a MINLP and is NP-hard [54]. The time-varying nature of $T_{nm}(k)$, $E_m(k)$, and $g_{nm}(k)$ across time slots introduces stochastic state transitions that preclude tractable closed-form solutions and motivate a reinforcement learning (RL) framework wherein agents learn adaptive policies through direct interaction with the environment.

6.3 Proposed Solution

We propose a hierarchical optimization framework to solve $\mathcal{P}$, where the inherent separation between macro-level UAV positioning and micro-level offloading decisions motivates a structured two-timescale solution. We formulate the problem as a Markov Decision Process (MDP) that captures stochastic and trust-varying nature of the environment [29], defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}_t \rangle$. The trust-weighted UAV deployment mechanism operates at the K_u timescale, aligning coverage with IoT device trust demands prior to offloading. RL schemes of decreasing centralization then resolve slot-level offloading and resource allocation decisions. This division assigns each decision to the layer best suited to make it. The deployment mechanism fixes each UAV's position $O_m(k)$ ahead of the slot. Once positions are set, the RL agent learns the offloading decision $\alpha_{nm}(k)$ and resource allocation $f_{nm}(k)$ for every

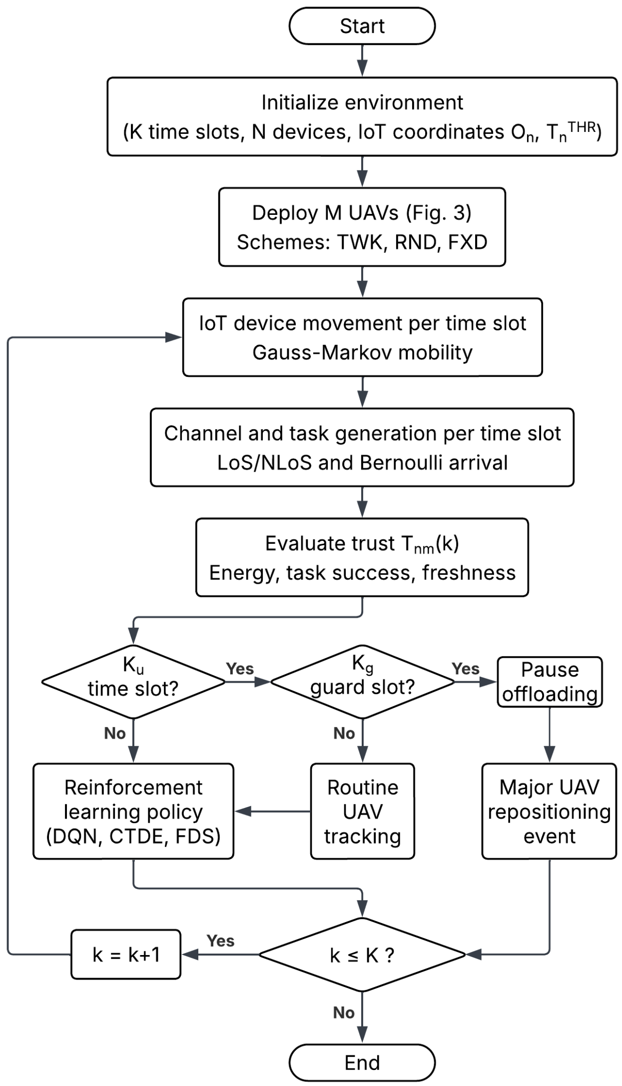


Figure 6.2: Flowchart of the per-episode simulation loop, from environment initialization to RL-based offloading.

IoT-UAV pair. Fig. 6.2 illustrates the per-episode simulation loop underlying the proposed trust-aware IoT offloading and UAV repositioning framework. The loop progresses through environment initialization, UAV deployment, device mobility, and trust evaluation, before the K_u/K_g branching governs routine tracking, offloading, or repositioning. The following subsections describe each of these components.

6.3.1 MDP Formulation

State Space

The system state at the *k-th* time slot captures trust scores, energy levels, channel conditions, task arrival indicators, task profiles, trust thresholds, and UAV positions across all IoT-UAV pairs. It is expressed as $\mathbf{s}(k) = \{T_{nm}(k), E_m(k), g_{nm}(k), a_n(k), D_n(k), \delta_n(k), T_n^{\text{THR}}, O_m(k)\}$. The centralized scheme grants the agent complete access to $\mathbf{s}(k)$ across all $N \times M$ pairs. The CTDE and distributed schemes restrict the *m-th* UAV to the local state $\mathbf{s}_m(k) = \mathbf{s}(k)\big|_{\mathcal{C}_m(k)}$, where $\mathcal{C}_m(k) = \{n : d_{nm}^h(k) \leq r_m\}$ denotes the set of devices within the *m-th* UAV's coverage footprint at the *k-th* time slot. The *m-th* UAV computes or directly measures $T_{nm}(k)$, $E_m(k)$, and $g_{nm}(k)$ independently. The *n-th* IoT device instead supplies $a_n(k)$, $D_n(k)$, $\delta_n(k)$, and T_n^{THR} through its task request header. This combination gives the UAV full access to $\mathbf{s}_m(k)$ without any global information exchange.

Action Space

The action at each time slot k takes the form $\mathbf{a}(k) = \{\alpha_{nm}(k), f_{nm}(k)\}$. Here, $\alpha_{nm}(k) \in \{0, 1\}$ denotes the offloading decision, and $f_{nm}(k) \in [0, F_m^{\text{MAX}}]$ denotes the computation resource allocated to the *n-th* IoT device by the *m-th* UAV. The CTDE and distributed schemes operate on the local action $\mathbf{a}_m(k) = \mathbf{a}(k)\big|_{\mathcal{C}_m(k)}$, restricted to devices within the *m-th* UAV's coverage footprint. The centralized scheme discretizes $f_{nm}(k)$ into J levels $\mathbb{F} = \{f_1, f_2, \ldots, f_J\}$ to enable DQN deployment, while the CTDE and distributed schemes employ MADDPG to handle continuous resource allocation. The action space structurally excludes actions violating C1 through C7.

Reward Function

This function guides the agent toward offloading decisions that minimize latency and energy consumption while maintaining UAV trustworthiness. The *n-th* IoT device and *m-th* UAV pair incur a penalty when $T_{nm}(k) < T_n^{\mathrm{THR}}$, penalizing any failure to meet the device's declared trust requirement. The IoT-UAV reward at each time slot k is expressed as:

$$r_{nm}(k) = -\alpha_{nm}(k)\Big[\mathcal{U}_{nm}(k) + \Gamma^{\mathrm{PNL}}\left[T_{nm}(k) < T_n^{\mathrm{THR}}\right]\Big], \tag{6.13}$$

where $\Gamma^{\mathrm{PNL}} > 0$ denotes the trust violation penalty coefficient. The aggregate reward across all active pairs at each time slot k is $r(k) = \sum_{n=1}^{N}\sum_{m=1}^{M} a_n(k) r_{nm}(k)$. The agent maximizes the discounted cumulative reward over K time slots:

$$\mathcal{J} = \mathbb{E}\left[\sum_{k=1}^{K}\gamma^{k-1} r(k)\right], \tag{6.14}$$

where $\gamma \in [0,1)$ denotes the discount factor. The expectation is taken over stochastic task arrivals, slot-wise LoS/NLoS realizations $\xi_{nm}(k)$, and Gauss-Markov device mobility. The offloading decisions $\alpha_{nm}(k)$ and UAV computational resources $f_{nm}(k)$ deterministically drive energy depletion and service freshness evolution, whereas $g_{nm}(k)$ varies stochastically with LoS/NLoS realizations and the mobility of devices and UAVs.

6.3.2 UAV Deployment Schemes

Fig. 6.3 illustrates the shared starting point for UAV tracking and repositioning across all schemes. The coverage area is partitioned into M regions, with one UAV placed at each region's center. The schemes diverge in how they track IoT device clusters at the K_u timescale and reposition across clusters at the K_g timescale, detailed in the following subsections.

Trust-Weighted K-means Deployment (TWK)

This proposed mechanism couples UAV positioning directly to trust-aware service demand, extending k-means-based clustering methods established for user grouping and UAV positioning in UAV-assisted MEC systems [70]. Its periodic operation benefits from two properties of the underlying algorithm. K-means produces exactly M cluster centroids without

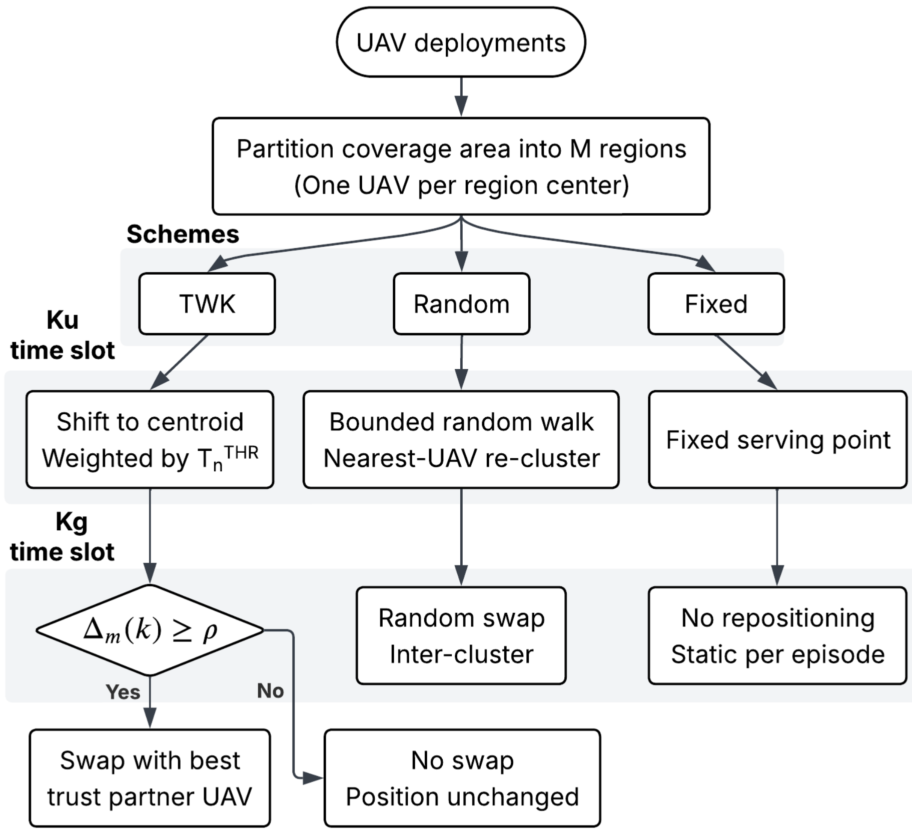


Figure 6.3: Flowchart of the UAV deployment schemes, showing their divergence from a shared initial placement through K_u tracking and K_g repositioning.

post-processing, and updates efficiently as device distributions evolve each K_u slot. TWK systematically weighs each IoT device's contribution to the clustering objective proportionally to its required trust threshold. This mirrors analogous weighted-clustering approaches that prioritize devices by other service-relevant features [118].

$$\min_{c(n)} \sum_{n=1}^{N} T_n^{\mathrm{THR}} \, |O_n(k) - O_{c(n)}(k)|^2, \tag{6.15}$$

where $c(n)$ denotes the cluster assignment of the n-th device. This objective draws UAVs toward devices with higher trust demands, structurally reducing competition for high-priority associations prior to the time slot-level offloading decisions.

The m-th UAV continuously tracks its cluster centroid every K_u time slots. It performs lateral adjustments within its assigned cluster to follow IoT device mobility. The network constructs a system-wide trust mapping across all UAV-IoT cluster pairs prior to each guard slot $k \in K_g$. Let $\mathcal{K}_m(k)$ denote the set of devices assigned to the *m-th* UAV's cluster by nearest-UAV assignment, distinct from the coverage footprint $\mathcal{C}_m(k)$. The network computes the average delivered trust $\bar{T}_m(k) = \frac{1}{|\mathcal{K}_m(k)|}\sum_{n\in\mathcal{K}_m(k)} T_{nm}(k)$ and the average demanded trust $\bar{T}_m^{\mathrm{THR}}(k) = \frac{1}{|\mathcal{K}_m(k)|}\sum_{n\in\mathcal{K}_m(k)} T_n^{\mathrm{THR}}$. The *m-th* UAV's trust mismatch indicator $\Delta_m(k) = \bar{T}_m^{\mathrm{THR}}(k) - \bar{T}_m(k)$ quantifies the gap between cluster demand and UAV capability. Inter-cluster repositioning triggers for any UAV satisfying $\Delta_m(k) \geq \rho$. The triggered UAVs and their clusters are each ranked in descending order, UAVs by trust capability $\bar{T}_m(k)$ and clusters by trust demand $\bar{T}_m^{\mathrm{THR}}(k)$. The UAVs are then reassigned to clusters of corresponding rank.

Random Deployment (RND)

This scheme repositions each UAV through a bounded random walk at the K_u timescale. The step draws a random direction and a capped distance from the UAV's current position. This repositioning proceeds independently of IoT device distribution or trust demand, which may lead to pronounced overlap or gaps in coverage. At each K_g interval, a uniformly random pair of UAVs exchanges position and cluster ownership, while the remaining UAVs stay unchanged. This inter-cluster adjustment mirrors TWK's own swap structure, but the paired UAVs are chosen by chance rather than by the mismatch trigger $\Delta_m(k) \geq \rho$.

Fixed Deployment (FXD)

This scheme assigns each UAV a static position, computed once before the episode by partitioning the coverage area A_C into M equal regions and placing each UAV at its respective region's center. These positions hold for the entire episode, with no adjustment at either the K_u or K_g timescale.

6.3.3 Reinforcement Learning Schemes

Centralized Deep Q-Network (DQN)

The centralized scheme addresses $\mathcal{P}$ through a single DQN agent [119, 120] that aggregates complete network information from all $N \times M$ IoT-UAV pairs at each time slot k. The joint action space comprises $\alpha_{nm}(k) \in \{0, 1\}$ and $f_{nm}(k) \in \mathbb{F}$, together forming a finite discrete structure. This enables exhaustive Q-value enumeration across each device's feasible choices, a direct evaluation that continuous actor-critic methods cannot perform. In this factored architecture, IoT devices independently select their UAVs, so more than J devices may simultaneously choose the same UAV and exceed its capacity. The framework resolves this by capping each UAV at its J devices with the highest trust threshold T_n^{THR}. IoT devices with lower trust thresholds go unserved for that time slot. This ordering keeps the most safety-critical devices served first. A discretized level is feasible for a device only if its fixed allocation meets that device's minimum viable requirement $f_{nm}^{\text{MIN}}(k)$. This quantity denotes the smallest CPU allocation that satisfies the *n-th* device's latency deadline under the current channel realization. Levels below this threshold cannot complete the task in time and are masked before action selection. The agent collects energy, task completion, and service freshness measurements from all UAVs at each time slot k. It computes $T_{nm}(k)$ centrally from these observations and removes self-reporting vulnerability.

The agent's centralized state access grants it complete observability of $\Delta_m(k)$ across all UAV-IoT cluster pairs at every K_u and K_g event. These tracking and repositioning decisions nonetheless remain governed by the active deployment scheme, independent of the RL architecture. The agent identifies active devices satisfying $a_n(k) = 1$ at each time slot k, restricting the effective action space to active IoT-UAV pairs. Action selection follows an ϵ-greedy policy applied to the joint action $\mathbf{a}(k) = \{\alpha_{nm}(k), f_{nm}(k)\}$. This balances exploration during early training with exploitation of learned value estimates as ϵ decays monotonically across episodes. This hard feasibility masking already excludes any choice violating C1 through C7. The penalty Γ^{PNL} in the reward function therefore never triggers for this architecture. It remains structurally relevant to the CTDE and FDS schemes that lack an equivalent hard constraint. The agent approximates the action-value function $Q(\mathbf{s}, \mathbf{a}; \boldsymbol{w})$ through a fully connected deep neural network with trainable weights $\boldsymbol{w}$. ReLU activations drive the hidden layers, a linear output layer yields the Q-values, and the Adam optimizer [121] updates the weights.

The network receives global state $\mathbf{s}(k)$ as input and produces a factored output, assigning the *n-th* IoT device its own block of $M \times J + 1$ Q-values. This block scores a choice $i_n(k) \in \{0, 1, \ldots, M \times J\}$, where M is the number of UAVs and J the number of discretization levels. The value $i_n(k) = 0$ denotes no offload. The value $i_n(k) = (m-1)J + j$ maps the *m-th* UAV's *j-th* level to a unique index within the block. Together, these indices span all $M \times J$ possible (UAV, level) combinations. The agent maintains a target network $Q(\mathbf{s}, i_n; \boldsymbol{w}^{\mathrm{TGT}})$ as a slowly evolving copy of the live network, adopting a soft update convention in place of DQN's original periodic hard copy [122]. Its parameters $\boldsymbol{w}^{\mathrm{TGT}}$ update through soft assignments $\boldsymbol{w}^{\mathrm{TGT}} \leftarrow \tau^{\mathrm{UPD}} \boldsymbol{w} + (1 - \tau^{\mathrm{UPD}}) \boldsymbol{w}^{\mathrm{TGT}}$, where $\tau^{\mathrm{UPD}} \ll 1$ denotes the soft update factor ensuring gradual tracking of the live network weights. This decoupling prevents Q-value targets from shifting rapidly during training, stabilizing the gradient updates. The replay buffer $\mathcal{D}$ stores one transition per IoT device per time slot, $(\mathbf{s}(k), i_n(k), r_n(k), \mathbf{s}', \mathcal{C}_n)$, where $r_n(k)$ denotes device n's own reward and $\mathcal{C}_n$ its coverage set at $\mathbf{s}'$. The agent samples mini-batches of size B_s uniformly across these device-level transitions to minimize the temporal difference loss:

$$\mathcal{L}(\boldsymbol{w}) = \mathbb{E}_{(\mathbf{s}, i_n, r_n, \mathbf{s}', \mathcal{C}_n) \sim \mathcal{D}} \left[\left(r_n + \gamma \max_{i'_n \in \mathcal{A}_n(\mathcal{C}_n)} Q(\mathbf{s}', i'_n; \boldsymbol{w}^{\mathrm{TGT}}) - Q(\mathbf{s}, i_n; \boldsymbol{w}) \right)^2 \right], \tag{6.16}$$

where $\mathcal{A}_n(\mathcal{C}_n) = \{i'_n = 0\} \cup \{i'_n = (m-1)J + j : (n, m) \in \mathcal{C}_n \text{ and } T_{nm}(k) \geq T_n^{\mathrm{THR}}\}$ denotes device n's coverage- and trust-feasible choices, with no-offload ($i'_n = 0$) always included. This target set omits the latency-feasibility criterion enforced during action selection. Consequently, the bootstrap target may evaluate the network at choices the policy

Algorithm 5 Centralized DQN Training

Require: Slots per episode K, minibatch size B_s, soft-update factor τ^{UPD}, discount factor γ, initial exploration ϵ^{INIT}, exploration decay ϵ^{DEC}, minimum exploration ϵ^{MIN}

Initialize weights $\boldsymbol{w}$, target weights $\boldsymbol{w}^{\mathrm{TGT}} \leftarrow \boldsymbol{w}$, buffer $\mathcal{D}$, $\epsilon \leftarrow \epsilon^{\mathrm{INIT}}$
for each episode **do**
    Observe initial state $\mathbf{s}$
    for $k = 1$ to K **do**
        if guard slot **then**
            $\mathbf{a} \leftarrow \mathbf{0}$ Guard slot: offloading suspended (C7)
        else
            For each device n: select i_n via ϵ-greedy policy over $Q(\mathbf{s}, \cdot; \boldsymbol{w})$, masked by coverage, trust, and latency feasibility; assemble $\mathbf{a}$ from all i_n; enforce capacity cap
        end if
        Execute $\mathbf{a}$; compute r_{nm} (6.13) and $r = \sum_{n,m} a_n \, r_{nm}$
        Observe next state $\mathbf{s}'$
        for each device $n = 1$ to N **do**
            $r_n \leftarrow r_{nm^*}$ for serving UAV m^*, else 0
            $\mathcal{C}_n \leftarrow$ coverage set of device n at $\mathbf{s}'$
            Push $(\mathbf{s}, i_n, r_n, \mathbf{s}', \mathcal{C}_n)$ to $\mathcal{D}$
        end for
        if $|\mathcal{D}| \geq B_s$ **then**
            Sample minibatch; update $\boldsymbol{w}$ by minimizing the temporal difference loss (6.16)
            $\boldsymbol{w}^{\mathrm{TGT}} \leftarrow \tau^{\mathrm{UPD}} \boldsymbol{w} + (1 - \tau^{\mathrm{UPD}}) \boldsymbol{w}^{\mathrm{TGT}}$
        end if
        $\mathbf{s} \leftarrow \mathbf{s}'$
    end for
    $\epsilon \leftarrow \max(\epsilon^{\mathrm{DEC}} \cdot \epsilon,\ \epsilon^{\mathrm{MIN}})$
end for

itself would never select. This gap between $\mathcal{A}_n(\mathcal{C}_n)$ and the true feasible set is left for future refinement. Training concludes with the agent adopting a greedy policy, selecting $i_n(k) = \arg\max_{i_n} Q(\mathbf{s}(k), i_n; \boldsymbol{w})$ independently for each device at each time slot k, without further exploration. The centralized architecture requires global state aggregation scaling as $O(NM)$ per slot in training and execution. It is a persistent overhead that grows with network size and motivates the decentralized schemes. Algorithm 5 summarizes the complete training procedure of the centralized DQN scheme.

Centralized Training, Decentralized Execution (CTDE)

This scheme separates global information access during training from local-only execution at deployment. A shared critic computes $T_{nm}(k)$ centrally during training and provides each actor with globally informed gradient signals that shape cooperative offloading policies. This same centralized access enables full TWK deployment prior to each guard time slot $k \in K_g$. The per-UAV actor architecture, unlike the critic, restricts each decision scope to devices within $\mathcal{C}_m(k)$ throughout both training and execution. The actor network's fixed input size limits $\mathbf{s}_m(k)$ to $\mathcal{C}^{\mathrm{MAX}}$ devices per slot, the largest number a single UAV's actor can process at once. The network pads unused entries with zeros and masks them accordingly. Trust threshold T_n^{THR} ranks devices under contention, and the network serves the most safety-critical connections first once demand for a UAV's local slots exceeds $\mathcal{C}^{\mathrm{MAX}}$.

MADDPG [123] suits this compact action space by directly outputting a bounded per-device preference $f_{nm}^{\mathrm{ACT}}(k) \in (0,1)$. This preference is produced through a deterministic policy $\mu(\cdot\,; \boldsymbol{w}_m^{\mathrm{ACT}})$ parameterized by actor weights $\boldsymbol{w}_m^{\mathrm{ACT}}$. The minimum CPU allocation required for device n's task to satisfy its latency deadline under the current channel realization is denoted $f_{nm}^{\mathrm{MIN}}(k)$. The UAV admits devices into $\hat{\mathcal{N}}_m(k) \subseteq \mathcal{C}_m(k)$ in order of decreasing actor preference $f_{nm}^{\mathrm{ACT}}(k)$. This admission continues until their cumulative minimum requirement would exceed F_m^{MAX}. This greedy ordering preserves the actor's own learned priority in determining which devices are served. The selected devices first receive their own minimum viable allocation. Remaining capacity is then distributed among $\hat{\mathcal{N}}_m(k)$ in proportion to actor preference:

$$f_{nm}(k) = f_{nm}^{\mathrm{MIN}}(k) + \frac{f_{nm}^{\mathrm{ACT}}(k)}{\sum_{n' \in \hat{\mathcal{N}}_m(k)} f_{n'm}^{\mathrm{ACT}}(k)} \left(F_m^{\mathrm{MAX}} - \sum_{n' \in \hat{\mathcal{N}}_m(k)} f_{n'm}^{\mathrm{MIN}}(k) \right), \tag{6.17}$$

with $\alpha_{nm}(k) = 1$ for $n \in \hat{\mathcal{N}}_m(k)$ and $\alpha_{nm}(k) = 0$ otherwise. This meets every served device's minimum need before allocating any surplus by preference, satisfying C4 by construction. The network trains across multiple episodes until convergence, with each UAV retaining only

$\boldsymbol{w}_m^{\mathrm{ACT}}$ for deployment. The *m-th* UAV selects $\mathbf{a}_m(k) = \mu(\mathbf{s}_m(k); \boldsymbol{w}_m^{\mathrm{ACT}})$ on local observations alone without inter-UAV communication. Inter-cluster repositioning remains unavailable at execution, as the network-wide trust mapping exists only during training.

The shared twin critics evaluate the joint action-value function $Q(\mathbf{s}, \mathbf{a}; \boldsymbol{w}_i^{\mathrm{CRT}})$, $i \in \{1, 2\}$, during training. The actor and critic networks draw mini-batches from shared replay buffer $\mathcal{D}$ accumulated across training episodes. Target networks $\boldsymbol{w}_m^{\mathrm{ACT,TGT}}$ and $\boldsymbol{w}_i^{\mathrm{CRT,TGT}}$ update through soft assignments $\boldsymbol{w}^{\mathrm{TGT}} \leftarrow \tau^{\mathrm{UPD}}\boldsymbol{w} + (1 - \tau^{\mathrm{UPD}})\boldsymbol{w}^{\mathrm{TGT}}$, where $\tau^{\mathrm{UPD}} \ll 1$ decouples bootstrapped targets from live networks between refresh intervals. The critic update minimizes the temporal difference loss:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}_i^{\mathrm{CRT}}) = \mathbb{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\sim\mathcal{D}}\Bigg[\bigg(r + \gamma \min_{i\in\{1,2\}} Q\big(\mathbf{s}', \mu(\mathbf{s}'_1; \boldsymbol{w}_1^{\mathrm{ACT,TGT}}), \ldots \\ \ldots, \mu(\mathbf{s}'_M; \boldsymbol{w}_M^{\mathrm{ACT,TGT}}); \boldsymbol{w}_i^{\mathrm{CRT,TGT}}\big) - Q(\mathbf{s}, \mathbf{a}; \boldsymbol{w}_i^{\mathrm{CRT}})\bigg)^2\Bigg], \quad i \in \{1, 2\}. \quad (6.18)\end{aligned}$$

The actor update ascends the policy gradient supplied by the first critic, per standard twin-critic convention.

$$\nabla_{\boldsymbol{w}_m^{\mathrm{ACT}}}\mathcal{J}_m = \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\nabla_{\mathbf{a}_m} Q(\mathbf{s}, \mathbf{a}; \boldsymbol{w}_1^{\mathrm{CRT}})\Big|_{\mathbf{a}_m=\mu(\mathbf{s}_m;\boldsymbol{w}_m^{\mathrm{ACT}})} \nabla_{\boldsymbol{w}_m^{\mathrm{ACT}}}\mu(\mathbf{s}_m; \boldsymbol{w}_m^{\mathrm{ACT}})\right]. \quad (6.19)$$

Fig. 6.4 summarizes this loop, which recurs at every time slot throughout training. The twin critics shown here have no counterpart at execution, where the trained actor alone determines each UAV's action.

Fully Distributed Scheme (FDS)

This scheme eliminates all centralized components from both training and execution. It addresses the residual dependency on shared infrastructure that CTDE retains during training. The *m-th* UAV maintains a local actor and a pair of local twin critics, all trained exclusively on local state $\mathbf{s}_m(k)$ and local action $\mathbf{a}_m(k)$. The actor network's fixed input size limits $\mathbf{s}_m(k)$

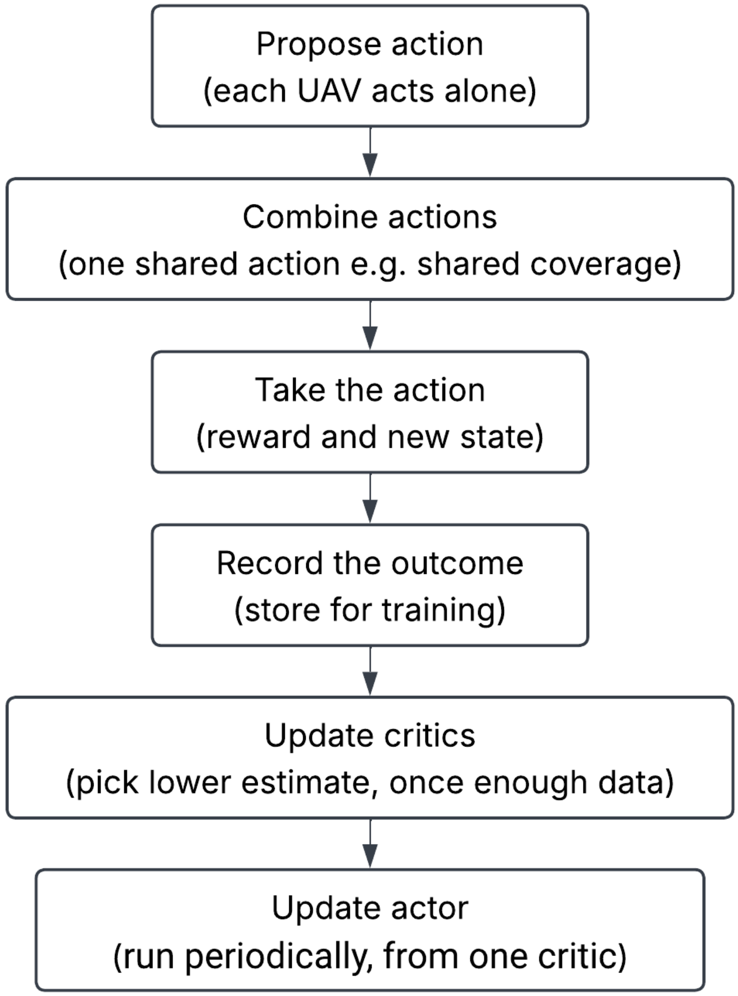


Figure 6.4: Per-slot training loop under CTDE, executed until convergence. Only local action selection persists at execution, as the twin critics operate exclusively during training.

to $\mathcal{C}^{\mathrm{MAX}}$ devices per slot, following the same padding, masking, and trust-priority mechanism used in the CTDE scheme. The local twin critics evaluate the action-value function $Q(\mathbf{s}_m(k), \mathbf{a}_m(k); \boldsymbol{w}_{m,i}^{\mathrm{CRT}})$, $i \in \{1, 2\}$, using only locally observable state-action pairs [30], with a decentralized actor-critic architecture governing the continuous action space through a deterministic policy $\mu(\cdot\,; \boldsymbol{w}_m^{\mathrm{ACT}})$. The actor similarly outputs a bounded per-device preference $f_{nm}^{\mathrm{ACT}}(k) \in (0, 1)$, normalized to satisfy C4 via the same mechanism as CTDE (6.17).

The *m-th* UAV operates as both an RL agent and a distributed control entity, participating in peer-to-peer trust validation every K_u slots. It broadcasts the average delivered trust $\bar{T}_m(k)$, average trust demand $\bar{T}_m^{\mathrm{THR}}(k)$, and local cluster centroid to all peers. This broadcast proceeds alongside its own intra-cluster position adjustments, following the same tracking mechanism as TWK. The received estimates feed into the Laplacian consensus update [107]:

$$\bar{T}_m^{\mathrm{CNS}}(k) = (1-\epsilon^{\mathrm{CON}})\bar{T}_m(k) + \frac{\epsilon^{\mathrm{CON}}}{M-1}\sum_{j\neq m}\bar{T}_j(k), \tag{6.20}$$

where $\epsilon^{\mathrm{CON}} \in (0,1)$ denotes the relative weight between self-reported and peer-averaged trust estimates. The UAVs hold consistent $\Delta_m(k)$, $\bar{T}_m^{\mathrm{THR}}(k)$, and cluster centroids by each guard slot $k \in K_g$. This enables UAVs to independently apply the TWK assignment rule and identify the same inter-cluster repositioning decisions without conflicts or central coordination. The repositioning triggers only when $\Delta_m(k) \geq \rho$ and a UAV with higher consensus trust exists in the network, ensuring movements serve trust-capability improvement.

Target networks $\boldsymbol{w}_m^{\mathrm{ACT,TGT}}$ and $\boldsymbol{w}_{m,i}^{\mathrm{CRT,TGT}}$ decouple bootstrapped targets from live networks through soft parameter updates. The actor and critic networks draw mini-batches from local replay buffer $\mathcal{D}_m$ maintained independently by the *m-th* UAV. The critic update minimizes the local temporal difference loss:

$$\mathcal{L}(\boldsymbol{w}_{m,i}^{\mathrm{CRT}}) = \mathbb{E}_{(\mathbf{s}_m,\mathbf{a}_m,r_m,\mathbf{s}'_m)\sim\mathcal{D}_m}\Bigg[\Big(r_m + \gamma \min_{i\in\{1,2\}} Q(\mathbf{s}'_m, \mu(\mathbf{s}'_m; \boldsymbol{w}_m^{\mathrm{ACT,TGT}}); \boldsymbol{w}_{m,i}^{\mathrm{CRT,TGT}}) \\ - Q(\mathbf{s}_m, \mathbf{a}_m; \boldsymbol{w}_{m,i}^{\mathrm{CRT}})\Big)^2\Bigg], \quad i\in\{1,2\}. \tag{6.21}$$

The actor update ascends the local policy gradient, using only the first critic per standard twin-critic convention:

$$\nabla_{\boldsymbol{w}_m^{\mathrm{ACT}}}\mathcal{J}_m = \mathbb{E}_{\mathbf{s}_m\sim\mathcal{D}_m}\Bigg[\nabla_{\mathbf{a}_m} Q(\mathbf{s}_m, \mathbf{a}_m; \boldsymbol{w}_{m,1}^{\mathrm{CRT}})\big|_{\mathbf{a}_m=\mu(\mathbf{s}_m;\boldsymbol{w}_m^{\mathrm{ACT}})}\nabla_{\boldsymbol{w}_m^{\mathrm{ACT}}}\mu(\mathbf{s}_m; \boldsymbol{w}_m^{\mathrm{ACT}})\Bigg]. \tag{6.22}$$

Algorithm 6 summarizes the training procedure shared by CTDE and FDS.

Algorithm 6 Actor-Critic Training (CTDE / FDS)

Require: K, minibatch size B_s, τ^{UPD}, γ, policy delay d_π, $\sigma^{\mathrm{INIT}}, \sigma^{\mathrm{DEC}}, \sigma^{\mathrm{MIN}}$, ϵ^{CON}

Initialize actors, shared twin critics $\boldsymbol{w}_{1,2}^{\mathrm{CRT}}$ (CTDE) or per-agent $\boldsymbol{w}_{m,1,2}^{\mathrm{CRT}}$ (FDS), targets, buffer(s)
for each episode e **do**
  $\sigma \leftarrow \max(\sigma^{\mathrm{INIT}}(\sigma^{\mathrm{DEC}})^{e-1}, \sigma^{\mathrm{MIN}})$; observe initial state
  for $k = 1$ to K **do**
    Each UAV: compute $\mathbf{s}_m(k)$ $(\leq \mathcal{C}^{\mathrm{MAX}})$; $f_{nm}^{\mathrm{ACT}}(k) \leftarrow \mu(\mathbf{s}_m; \boldsymbol{w}_m^{\mathrm{ACT}}) + \mathcal{N}(0, \sigma)$; decode into $\mathbf{a}_m$ via (6.17); scatter into $\mathbf{a}$
    Resolve multi-UAV conflicts; recompute each $\mathbf{a}_m$ from resolved $\mathbf{a}$
    Execute $\mathbf{a}$; observe r (CTDE) or r_m (FDS); observe next state
    Push transition to $\mathcal{D}$ (CTDE, global) or $\mathcal{D}_m$ (FDS, local)
    if buffer(s) $\geq B_s$ **then**
      Update critic(s) via min-clipped TD-target; soft-update targets
      if iteration mod $d_\pi = 0$ **then**
        Update each actor via $\nabla_{\boldsymbol{w}_m^{\mathrm{ACT}}} \mathcal{J}_m$ (6.19), (6.22); soft-update actor targets
      end if
    end if
    if FDS **and** k mod $K_u = 0$ **then**
      Compute $\bar{T}_m^{\mathrm{CNS}}$, all M UAVs jointly (6.20)
    end if
    $\mathbf{s} \leftarrow \mathbf{s}'$
  end for
end for
return $\{\mu(\cdot; \boldsymbol{w}_m^{\mathrm{ACT}})\}_{m=1}^{M}$

Complexity and Overhead Analysis

Table 6.1 summarizes the information scope, communication overhead, trust verification mechanism, and failure resilience across the three proposed schemes. The centralized scheme achieves the broadest information access and strongest policy coordination through continuous global communication sustained throughout both training and execution. CTDE confines global information access to the training phase, eliminating inter-UAV communication overhead at execution entirely. The fully distributed scheme eliminates centralized dependencies entirely and derives trust through peer consensus among all M UAVs. The three schemes collectively reflect a fundamental tradeoff between policy quality, communication overhead, and deployment resilience.

Table 6.1: Comparison of RL architectures in terms of information scope, communication overhead, and resilience.

Property	Centralized DQN	CTDE	FDS
Training	Global	Global	Local
Execution	Global	Local	Local
Trust	Central	Central (train) / Local (exec.)	Local (consensus)
Resilience	Low	Medium (post-train)	High
Scalability	Low	Medium	High
Comm. overhead/episode	$O(NMK)$	$O((N+M)K)$	$O((N+M)K)+O\left(\frac{K}{K_u}M^2\right)$
N=50, M=4, K=200, K_u=25	40,000	10,800	10,928

6.4 Performance Analysis

6.4.1 Simulation Setup

The simulations evaluate the proposed trust-aware UAV-assisted MEC framework within $\mathcal{A}_C$ (radius 1200m), where $M=4$ UAVs serve $N=50$ IoT devices distributed uniformly across the ground plane without any terrestrial infrastructure. The network and channel parameters follow the multi-UAV IoT edge network setup of [112], with UAVs hovering at $z_m = 150$m under the probabilistic A2G path loss model. The absence of terrestrial infrastructure necessitates three parameter adjustments to reflect disaster emergency conditions. Subchannel bandwidth widens to $B = 5$MHz [124], task data sizes follow $D_n(k) \sim \mathcal{U}[0.5, 2]$Mb, and CPU cycle requirements follow $\delta_n(k) \sim \mathcal{U}[10, 50]$ cycles/bit [52], capturing the lighter but time-critical workloads of disaster-response IoT devices.

The RL architectures share a unified neural network structure comprising two fully connected hidden layers of 256 and 128 neurons. The maximum number of devices considered by each UAV is set to $\mathcal{C}^{\text{MAX}}=30$ per time slot for both the CTDE and FDS schemes. The setting reflects the actor network's fixed input size rather than a physical service limit. As a worst-case check, splitting the UAV's full capacity equally across $\mathcal{C}^{\text{MAX}}$ devices gives each device $F_m^{\text{MAX}}/\mathcal{C}^{\text{MAX}}=0.5$GHz, which completes the largest considered task within 0.2s. This remains comfortably within the 2s latency requirement, confirming that the capacity limit does not compromise feasibility. Actor and critic networks use learning rates of 10^{-4} and 3×10^{-4} respectively, discount factor $\gamma=0.99$, soft update factor 10^{-3}, and mini-batch size of 256. The replay buffer holds 10^5 transitions for the centralized DQN and 10^4 for CTDE and FDS. Training runs for 300 episodes of $K=200$ time slots each. Table 6.2 summarizes all simulation parameters.

Table 6.2: Simulation parameters.

Parameter	**Symbol**	**Value**	**Unit**
Network and Channel			
Number of UAVs	M	4	–
Number of IoT devices	N	50	–
UAV altitude	z_m	150	m
Subchannel bandwidth	B	5	MHz
Carrier frequency	f_c	2	GHz
LoS constant	a	9.61	–
LoS constant	b	0.16	–
LoS attenuation	η_{LoS}	1	dB
NLoS attenuation	η_{NLoS}	10	dB
IoT transmit power	p_n	10	dBm
Noise power	σ^2	-80	dBm
UAV tracking interval	K_u	25	slots
Guard slot interval	K_g	100	slots
Mobility			
Trajectory smoothness	ω_n	0.75	–
Mean device speed	$\bar{v}_n$	3.0	m/s
Mean direction	$\bar{\theta}_n$	$[0, 2\pi]$	rad
Computation and Task			
UAV CPU capacity	F_m^{MAX}	15	GHz
CPU discretization (DQN)	J	30	–
Local state cap (CTDE/FDS)	$\mathcal{C}^{\text{MAX}}$	30	–
IoT local CPU	f_n	1	GHz

Table 6.2 – continued from previous page

Parameter	**Symbol**	**Value**	**Unit**
Task data size	$D_n(k)$	[0.5, 2]	Mb
CPU cycles per bit	$\delta_n(k)$	[10, 50]	cycles/bit
Maximum latency	L_n^{MAX}	2	s
Switched capacitance	κ_m	10^{-28}	–
UAV min energy	E_m^{MIN}	10	J
UAV max energy (per UAV)	E_m^{MAX}	[80, 120]	J
Trust Framework			
Trust weights	w_1, w_2, w_3	1/3 each	–
Utility weights	$\nu^{\mathrm{LAT}}, \nu^{\mathrm{ENE}}, \nu^{\mathrm{TRS}}$	1/3 each	–
Trust threshold	T_n^{THR}	[0.1, 0.9]	–
Consensus step	ϵ^{CON}	0.1	–
Trust penalty	Γ^{PNL}	10	–
Repositioning threshold	ρ	0.15	–
Big-M constant	$\mathbb{M}$	10	–
RL Hyperparameters			
Q-network learning rate	–	10^{-4}	–
Actor learning rate	–	10^{-4}	–
Critic learning rate	–	3×10^{-4}	–
Discount factor	γ	0.99	–
Policy delay	d_π	1	–
Initial exploration / noise	$\epsilon^{\mathrm{INIT}}/\sigma^{\mathrm{INIT}}$	1.0	–
Exploration / noise decay	$\epsilon^{\mathrm{DEC}}/\sigma^{\mathrm{DEC}}$	0.995	–
Min. exploration / noise	$\epsilon^{\mathrm{MIN}}/\sigma^{\mathrm{MIN}}$	0.05	–
Soft update factor	τ^{UPD}	10^{-3}	–
Replay buffer size (DQN)	$\mathcal{D}$	10^5	–
Replay buffer size (CTDE, per-agent FDS)	–	10^4	–
Mini-batch size	B_s	256	–
Training episodes	–	300	–
Time slots	K	200	–

6.4.2 Performance Metrics

This section defines the metrics used to evaluate the proposed framework, covering training behavior, offloading effectiveness, and trust-driven service quality.

• Training Reward: It tracks the effectiveness of RL architectures in learning to offload tasks by combining latency, energy, and trust into a unified signal per episode. The episode reward sums $r(k)$ over the K time slots of an episode, and a higher value indicates better policy. Reward fluctuates from episode to episode, since task arrivals, mobility, and channel conditions vary randomly within each episode.

• Offloading Rate: It quantifies the fraction of devices with an active task that are successfully offloaded to a UAV. Successful offloading occurs when $\alpha_{nm}(k) = 1$ for at least one UAV. This rate rises with strong demand accommodation and falls when unserved demand is substantial.

• Reward per Served Device: It evaluates the average reward earned per successfully offloaded device, isolating decision quality from serving volume. The metric divides the total episode reward by the number of served IoT-UAV pairs. A high value reflects efficient, well-targeted offloading, whereas a low value reflects reward diluted across many marginal associations.

• Mean Trust (Served): It tracks the trust level of devices that are actually offloaded. The metric averages $T_{nm}(k)$ over all IoT-UAV pairs with $\alpha_{nm}(k) = 1$. A high value indicates that the system favors high-trust associations, whereas a value near the minimum threshold indicates the system serves devices as soon as they qualify.

• Load Balance: It quantifies the distribution of computational load across the M UAVs, capturing whether offloading demand concentrates on a few UAVs or spreads evenly across the network. The *m-th* UAV occupies a fraction $\sum_{n\in\mathcal{N}_m(k)} f_{nm}(k)/F_m^{\mathrm{MAX}}$ of its capacity at time slot k, and averaging this fraction over the episode gives its mean utilization, producing one value per UAV. The load balancing score is the standard deviation of these M values divided by their mean. A low score indicates that all UAVs utilize a similar fraction of their capacity, whereas a high score indicates that utilization differs sharply across UAVs.

• Served Latency: It quantifies the end-to-end delay experienced by successfully offloaded devices. The metric averages $L_{nm}(k)$ over all IoT-UAV pairs with $\alpha_{nm}(k) = 1$, reported in seconds. A low value reflects fast task completion, whereas a high value approaches the latency deadline L_n^{MAX}.

• Served Energy: It quantifies the energy consumed in transmitting and executing successfully offloaded tasks. The metric averages $E_{nm}(k)$ over all IoT-UAV pairs with $\alpha_{nm}(k) = 1$, reported in joules. A low value reflects efficient task delivery, whereas a high value reflects substantial energy expenditure.

• Trust Exclusion Rate: It quantifies the fraction of devices denied service due to insufficient trust, distinct from denial caused by coverage or capacity limits. The IoT device with an active task ($a_n(k) = 1$), lying within coverage of at least one UAV ($d^h_{nm}(k) \leq r_m$) is trust-excluded if $T_{nm}(k) < T_n^{\mathrm{THR}}$ holds for every such covering *m-th* UAV. The trust exclusion rate indicates the fraction of task-active, in-range devices that are trust-excluded. A low rate indicates that most of such IoT devices meet at least one covering UAV's trust threshold, whereas a high rate indicates that trust is the dominant cause of unserved demand.

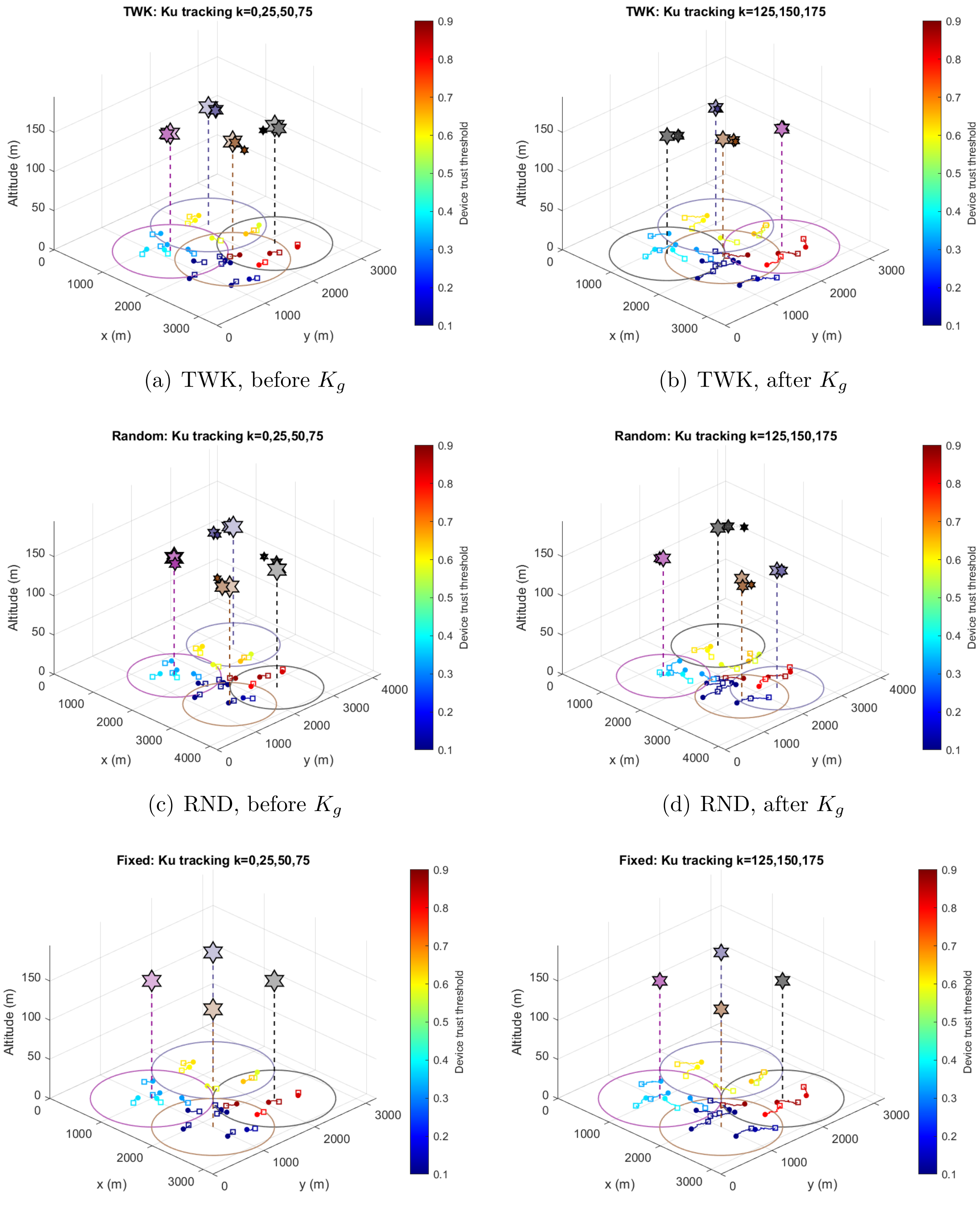
6.4.3 Simulation Results

In this section, we evaluate the performance of the proposed trust-aware framework. We first visualize UAV repositioning under each deployment scheme within a single episode, illustrating intra-cluster tracking at K_u and inter-cluster swaps at K_g. We then report training reward across the RL architectures and deployment schemes. Finally, we examine system sensitivity to the repositioning threshold ρ, the trust threshold T_n^{THR}, and the number of IoT devices N.

Fig. 6.5 illustrates the intra-cluster K_u tracking of UAVs under each deployment scheme within a representative episode. This behavior remains independent of the underlying RL architecture, since UAV positioning is governed entirely by the deployment mechanism itself. IoT device mobility motivates this periodic tracking. We display 20 of the $N = 50$ IoT devices for visual clarity of the plots. The devices employ the Gauss-Markov mobility model with mean speed $\bar{v}_n = 3$ m/s. The filled circles mark each device's initial position and hollow squares mark their final positions, with a colored line tracing the path. The colors correspond to each IoT device's trust threshold T_n^{THR} as shown on the colorbar. TWK, FXD, and RND each govern UAV repositioning through a distinct mechanism in response to this mobility. Fig. 6.5(a) and Fig. 6.5(b) show that TWK shifts UAVs toward the trust-weighted centroid of their cluster. Specifically, Fig. 6.5(b) reflects the relocation of UAVs following the K_g swap, with each UAV repositioned according to its cluster's trust requirements. Fig. 6.5(c) and Fig. 6.5(d) illustrate the bounded random walk adopted by RND, which repositions UAVs independently of device positions or trust demand. Fig. 6.5(e) and Fig. 6.5(f) show the static positioning under FXD, where each UAV remains at its initial hovering point throughout the episode. Collectively, these plots show UAV positioning behavior before ($k = 0, 25, 50, 75$) and after ($k = 125, 150, 175$) the guard-slot event at $k = 100$.

Fig. 6.6 examines how each deployment scheme responds to a mismatch between UAV trust capability and cluster trust demand at the guard slot $k = 100$. The results reveal three distinct outcomes. TWK resolves the mismatch through its design, RND resolves it only incidentally, and FXD leaves it entirely unaddressed. Fig. 6.6(a) and Fig. 6.6(b) show TWK correcting this mismatch through a direct pairing rule. UAV 4 offers the highest delivered trust in the network (0.63), yet at the guard slot it is serving a cluster requiring only 0.36. UAV 1 offers 0.42, yet it is serving the cluster with the highest requirement in the network (0.80). TWK swaps these two assignments so that UAV 4 moves to the high-demand cluster its capability suits, while UAV 1 moves to the low-demand cluster it can comfortably support. This exchange reduces total mismatch across the network from 0.82 to 0.40, aligning each UAV's trust capability with the demand of the cluster it now serves.

Fig. 6.6(c) and Fig. 6.6(d) show RND swapping UAV 1 and UAV 2 without regard to



(a) TWK, before K_g

(b) TWK, after K_g

(c) RND, before K_g

(d) RND, after K_g

(e) FXD, before K_g

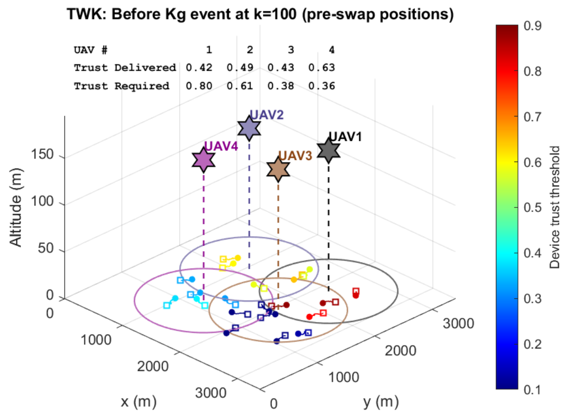
(f) FXD, after K_g

Figure 6.5: Intra-cluster UAV repositioning at K_u slot across deployment schemes, comparing the adjustments before ($k = 0, 25, 50, 75$) and after ($k = 125, 150, 175$) the guard-slot event at $k = 100$.

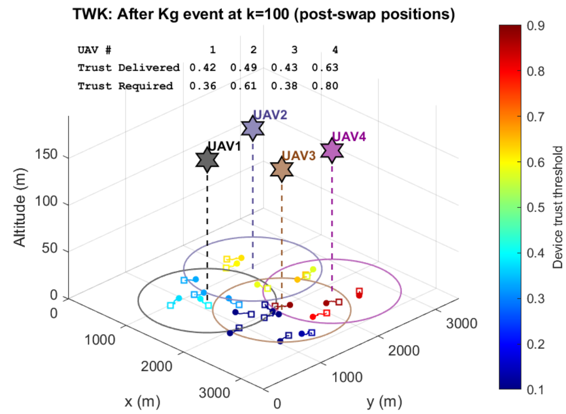
trust. UAV 1 moves to a lower-demand cluster, and its mismatch falls from 0.45 to 0.23. UAV 2 moves to the higher-demand cluster UAV 1 leaves behind, and its mismatch rises from 0.14 to 0.36 by nearly the same amount. Total mismatch across the network remains unchanged at 1.23, since the gain at UAV 1 is offset almost exactly by the loss at UAV 2. Fig. 6.6(e) and Fig. 6.6(f) show FXD holding its UAV positions fixed throughout the episode. Total mismatch across the network remains essentially unchanged at 1.05, since no repositioning takes place. UAV 3's required trust rises slightly from 0.29 to 0.30, and UAV 4's falls from 0.36 to 0.35. This minor reciprocal shift arises from a device drifting across the boundary between its nearest-cluster assignments.

Fig. 6.7 presents the 10-episode moving average of training reward across 300 episodes, reported separately for each RL architecture and each deployment scheme. Stochastic task arrivals, device mobility, and channel conditions vary independently each episode, so reward fluctuates throughout training rather than rising steadily. Fig. 6.7(a) compares reward across the three RL architectures, where DQN attains the highest reward throughout training. It oscillates near 180 with occasional peaks exceeding 200 and a floor that remains above 160. CTDE and FDS both yield substantially lower reward and maintain similar performance for approximately the first 50 episodes. CTDE subsequently improves compared to FDS, with peak reward of approximately 150 around training episode 230. FDS maintains the weakest performance, with its lowest reward of approximately 90 around training episode 110. The gap between CTDE and FDS reflects the structural cost of fully distributed training and execution, under which FDS forgoes the shared critic that continues to benefit CTDE. DQN's advantage over CTDE and FDS stems from its centralized access to complete network information at every decision.

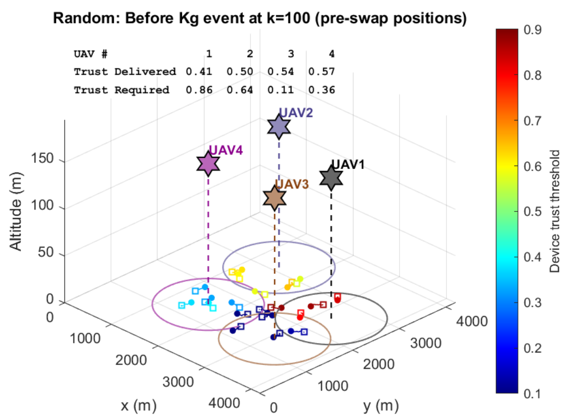
Fig. 6.7(b) compares reward across the three deployment schemes. TWK attains the highest reward throughout training and oscillates near 160, consistent with the trust-capability pairing mechanism its guard-slot swap applies at every repositioning event. Reward under TWK drops to approximately 130 during two pronounced dips near episodes 110 and 250,



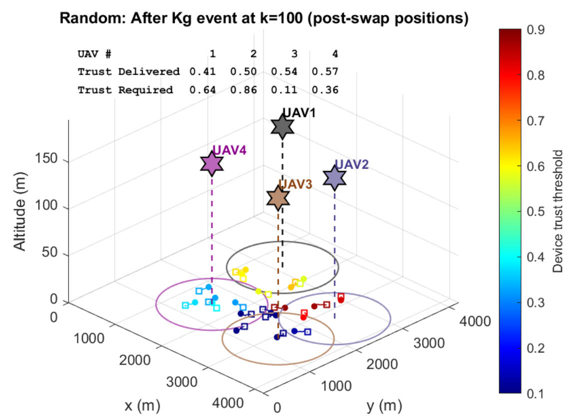
(a) TWK-Kg (prior)



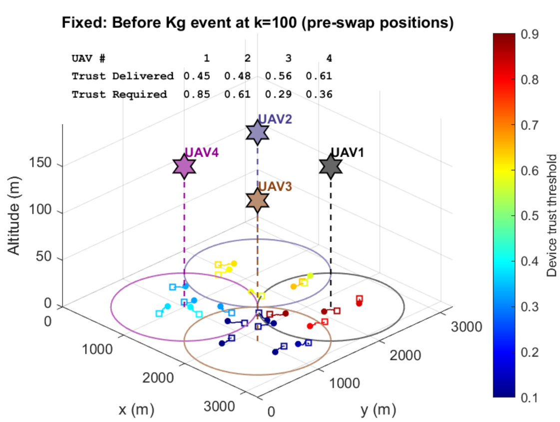
(b) TWK-Kg (post)



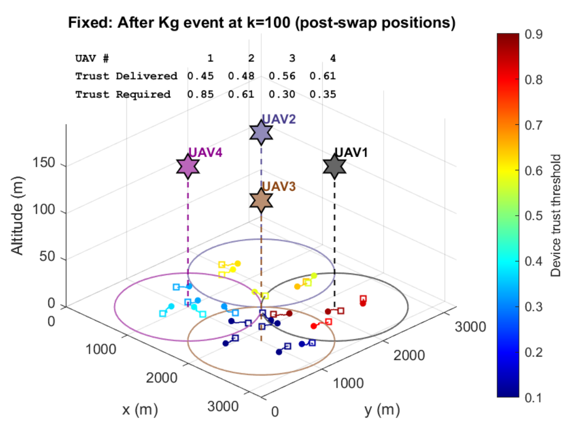
(c) RND-Kg (prior)



(d) RND-Kg (post)

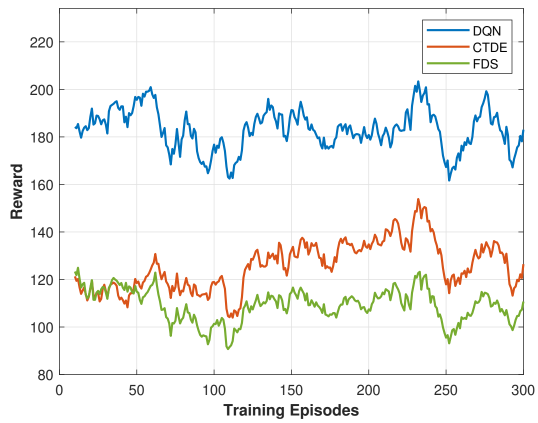


(e) Fix-Kg (prior)

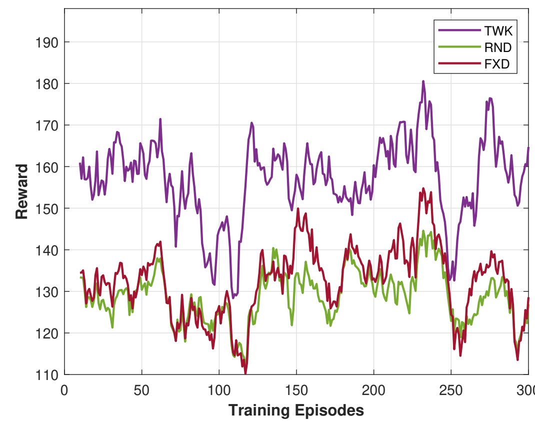


(f) Fix-Kg (post)

Figure 6.6: Inter-cluster UAV repositioning at the guard-slot event $k = 100$ across deployment schemes, comparing UAV trust delivery and cluster trust demand before and after the swap.



(a) RL architecture rewards averaged over deployment schemes.



(b) Deployment schemes rewards averaged over RL architectures.

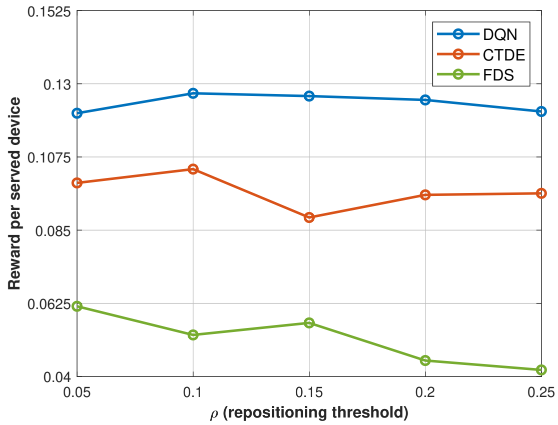
Figure 6.7: Training rewards for $N = 50$ devices, $M = 4$ UAVs, $K = 200$ time slots, $K_u = 25$, and $K_g = 100$.

reflecting the same episode-to-episode stochasticity that affects every scheme rather than any failure specific to TWK. RND and FXD both yield substantially lower reward than TWK and remain close to each other throughout training, with RND's active but trust-blind repositioning offering no measurable advantage over FXD's complete absence of repositioning. FXD occasionally surpasses RND during training by as much as 15, a gap too small and inconsistent to indicate any systematic benefit of one baseline over the other. TWK's advantage over the baselines confirms that reward improves due to trust-guided repositioning, instead of randomized movements.

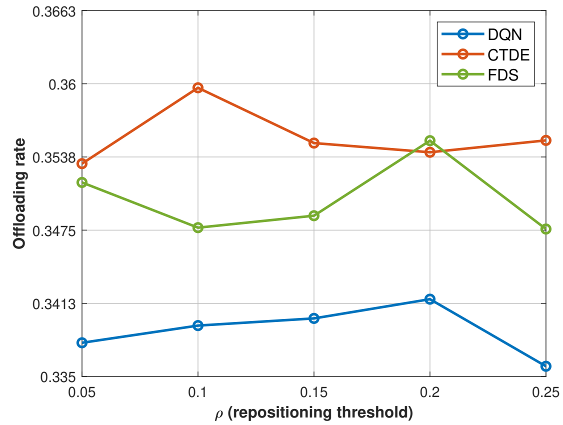
Fig. 6.8 reports the impact of variations in the repositioning threshold ρ across RL architectures under TWK deployment. These results reflect the mean over 10 test episodes under a greedy policy following training. The repositioning threshold governs how readily TWK corrects a mismatch between a UAV's delivered trust and its cluster's required trust. A small ρ triggers a swap at the slightest such mismatch, whereas a larger value tolerates greater mismatch before triggering one. Fig. 6.8(a) shows that DQN sustains the highest reward per served device and remains nearly stable across every value of ρ. Its hard feasibility masking selects efficient allocations independently of cluster tracking, leaving it largely insensitive to the drift a larger ρ permits. CTDE performs second best, declining gradually with modest fluctuation as ρ increases. FDS trails both and declines more steeply over the same range. CTDE and FDS rely on a UAV's own locally observed trust, with FDS additionally reconciling this view against its peers through consensus. Hence, the drift permitted by a larger ρ degrades their performance more visibly than DQN's structurally guarded efficiency.

Fig. 6.8(b) shows the offloading rate achieved by the RL architectures, where CTDE achieves the highest rate overall. CTDE's offloading rate declines gradually with minor fluctuation across most values of ρ, then rises again between $\rho = 0.2$ and $\rho = 0.25$. FDS achieves the second-highest offloading rate and follows a non-monotonic pattern across ρ that briefly places it above CTDE near $\rho = 0.2$. DQN trails both CTDE and FDS despite its centralized and complete view of the network. DQN's reward function penalizes trust violation as a binary condition rather than rewarding trust margin. Its hard feasibility masking also selects a smaller, efficient subset of devices once coverage, trust, and latency checks are satisfied, without incentive to admit more. Fig. 6.8(c) shows that CTDE achieves the highest mean trust served across ρ, with modest performance fluctuations. FDS achieves the second-highest mean trust served and declines consistently as ρ increases. As in Fig. 6.8(b), DQN achieves the lowest mean trust because its reward function grants identical reward to any device served above its trust threshold regardless of the margin by which that threshold is cleared. CTDE and FDS lack DQN's explicit verification of this threshold, and learn to maintain a wider margin to guard against the same binary penalty.

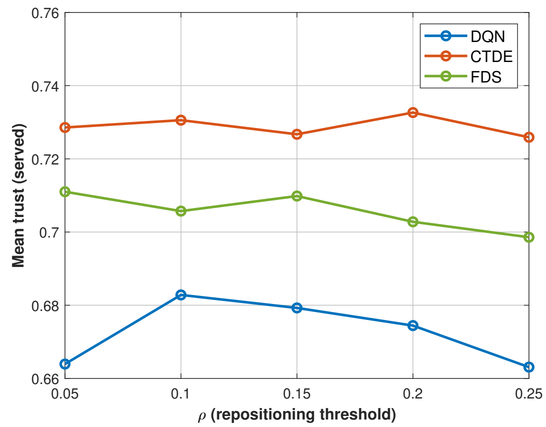
Fig. 6.9 evaluates the performance of deployment-architecture schemes jointly across a set of performance metrics. The trust threshold affects both the RL policy's offloading decisions and the deployment scheme's cluster assignment, which motivates a collective analysis. The trust threshold is applied uniformly across all devices, and each value on the x-axis represents a shared requirement rather than the heterogeneous per-device distribution used elsewhere in this work. These results reflect the mean over 10 test episodes under a greedy policy following training. Fig. 6.9(a) shows reward declining as the trust threshold tightens across nearly every scheme. Reward falls to near zero by $T_n^{\mathrm{THR}} = 0.9$, since UAVs become unable



(a) Reward per served device



(b) Offloading rate



(c) Mean trust (served)

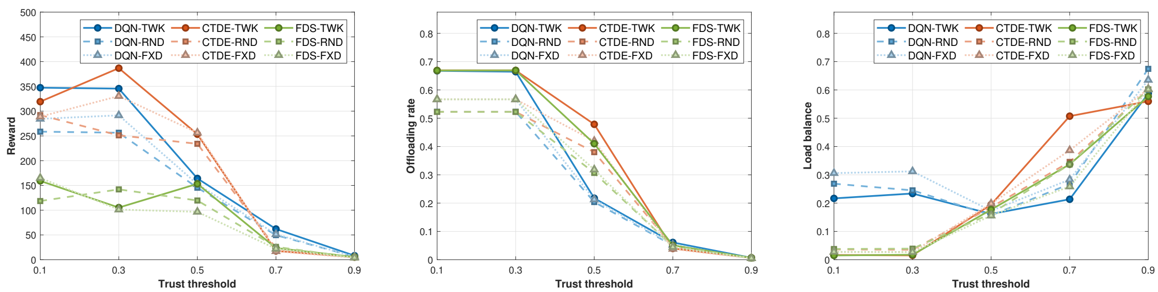
Figure 6.8: Sensitivity to the repositioning threshold under TWK deployment, for $N = 50$ devices, $M = 4$ UAVs, $K = 200$.

to support such demanding requirements for any device at this point. Two schemes deviate briefly from this decline. CTDE-TWK rises from $T_n^{\mathrm{THR}} = 0.1$ to 0.3, and FDS-TWK rises from 0.3 to 0.5. CTDE-TWK and DQN-TWK exhibit comparable performance across the trust threshold range, while FDS-TWK trails both. TWK outperforms RND and FXD within each architecture, with one exception: FDS-RND slightly exceeds FDS-TWK at $T_n^{\mathrm{THR}} = 0.3$.

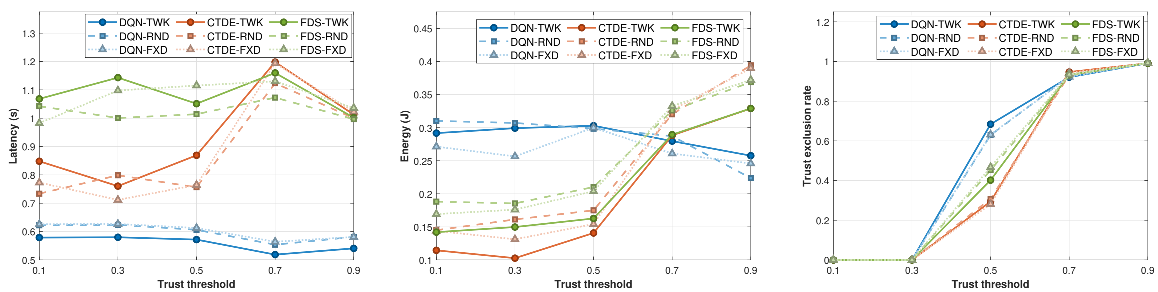
Fig. 6.9(b) shows offloading rate declining as the trust threshold increases. It remains stable from $T_n^{\mathrm{THR}} = 0.1$ to 0.3, and decays toward zero as T_n^{THR} approaches 0.9. CTDE-TWK shows superior performance across schemes, while FDS-TWK maintains a close approximation. DQN-TWK trails both CTDE-TWK and FDS-TWK, indicating reverse performance compared to Fig. 6.9(a). The TWK variant holds its lead within every architecture, which indicates that trust-aware deployment lifts offloading rate independently of the underlying RL scheme. Fig. 6.9(c) shows load balancing across the trust threshold. DQN shows the worst balancing up to $T_n^{\mathrm{THR}} = 0.5$, while FDS and CTDE maintain a better and close approximation to each other. The low-trust UAVs cannot support IoT devices as trust thresholds grow, thus concentrating service on high-trust UAVs and creating a stronger imbalance. DQN becomes the best-balanced architecture at $T_n^{\mathrm{THR}} = 0.5$ and beyond. FDS follows in the middle, while CTDE shows the worst load balancing.

Fig. 6.9(d) shows DQN sustaining the lowest served latency across the full range of trust thresholds. DQN-TWK achieves the lowest latency among its own deployment variants. This advantage follows from DQN's explicit feasibility check, which excludes any discretized allocation level whose fixed capacity chunk cannot complete a device's task within its remaining latency budget, given the device's current channel conditions. CTDE requires higher latency than DQN, rising gradually until $T_n^{\mathrm{THR}} = 0.7$ and then falling as the threshold approaches 0.9. FDS records the highest latency overall but performs marginally better than CTDE between $T_n^{\mathrm{THR}} = 0.7$ and 0.9. CTDE and FDS both lack DQN's built-in check that every candidate action already meets the latency deadline. This leaves their continuous policies a harder trade-off to learn between latency and feasibility.

Fig. 6.9(e) shows DQN incurring the highest served energy up to $T_n^{\mathrm{THR}} = 0.6$, afterwards its energy declines and falls below both CTDE and FDS through 0.7-0.9. CTDE holds the lowest energy through $T_n^{\mathrm{THR}} = 0.7$ and FDS closely follows it across the trust thresholds. The TWK variant of both CTDE and FDS achieves the lowest energy among their deployment counterparts. This crossover is consistent with fewer and more easily served devices remaining eligible as the threshold increases. Fig. 6.9(f) shows trust exclusion rising for every scheme as the threshold tightens, reaching near-total exclusion by $T_n^{\mathrm{THR}} = 0.9$. The clearest separation among schemes appears between $T_n^{\mathrm{THR}} = 0.3$ and 0.7, where CTDE excludes the



(a) Reward (b) Offloading ratio (c) Load balancing



(d) Served latency (e) Served energy (f) Trust exclusion rate

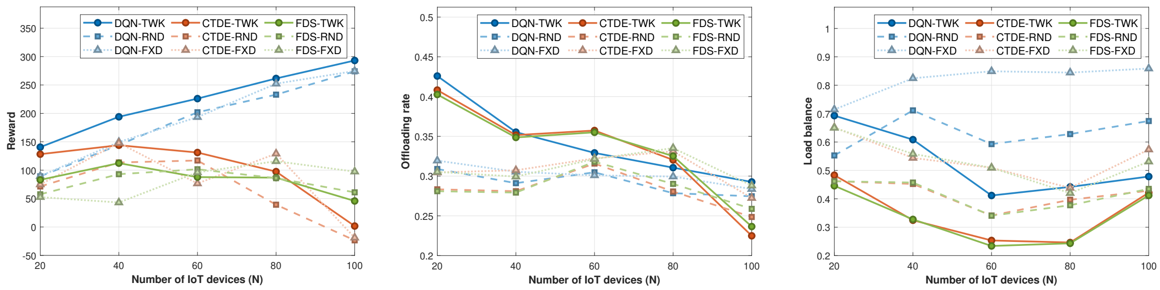
Figure 6.9: Sensitivity of performance metrics to the trust threshold, across architecture-deployment schemes, for $N = 50$ devices, $M = 4$ UAVs, $K = 200$ time slots.

fewest devices, FDS falls in between, and DQN excludes the most. This ordering follows the same trust-margin pattern established for ρ. DQN's policy clears the trust threshold with minimal margin, so a rising threshold quickly pushes more of its served devices below the required line. CTDE's wider margin stems from lacking DQN's explicit verification step and continues to protect more of its served devices as the requirement tightens. CTDE-TWK closely tracks the architecture-level pattern for CTDE, while FDS-TWK performs marginally better than its own deployment variants (FDS-RND and FDS-FXD). DQN-TWK performs slightly worse as compared to both of its deployment variants (DQN-RND and DQN-FXD).

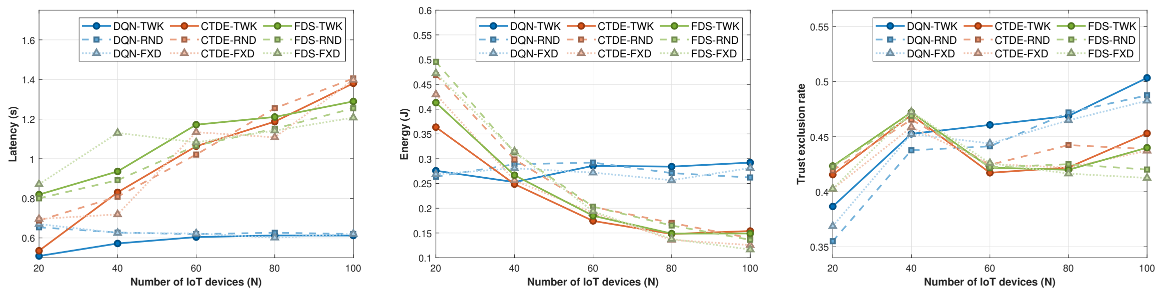
Fig. 6.10 evaluates deployment-architecture schemes as the number of IoT devices grows from $N = 20$ to $N = 100$, with $M = 4$ UAVs. The IoT devices retain their default heterogeneous trust thresholds across $[0.1, 0.9]$, unlike the uniform threshold used in the trust threshold sweep of Fig. 6.9. These results reflect the mean over 10 test episodes under a greedy policy following training. Fig. 6.10(a) shows DQN-TWK achieving the highest reward and increasing consistently as N grows. CTDE follows as the second-best performer, with

FDS trailing and briefly overtaking CTDE only at $N = 100$. CTDE and FDS both degrade steadily as N increases, since each UAV's actor observes only a fixed number of devices capped at $\mathcal{C}^{\text{MAX}}$ and a growing share exceeds this cap as N rises. DQN's actor instead observes every device regardless of N and avoids this constraint entirely. Fig. 6.10(b) shows offloading rate decaying non-uniformly across all schemes. DQN declines more smoothly than CTDE and FDS and performs best at $N = 20$ and $N = 100$. DQN-TWK, CTDE-TWK and FDS-TWK converge to similar offloading rates at $N = 40$. CTDE and FDS both exceed DQN between $N = 60$ and $N = 80$, where CTDE-FXD and FDS-FXD also perform better than CTDE-TWK and FDS-TWK, respectively. Fig. 6.10(c) shows DQN achieving better load balancing than CTDE and FDS overall. CTDE-TWK and FDS-TWK record the worst balancing among all schemes, though both improve between $N = 80$ and $N = 100$.

Fig. 6.10(d) depicts the served latency increasing with N across all schemes. DQN achieves the lowest latency while CTDE follows as the second-lowest and FDS records the highest overall, though FDS improves over CTDE between $N = 80$ and $N = 100$. This rise for CTDE and FDS reflects the same $\mathcal{C}^{\text{MAX}}$ limit, since more devices competing for each UAV's fixed-size local view produce allocations that grow less adequate as latency approaches the deadline. FDS-RND achieves consistently lower latency than FDS-TWK across the full range of N, and FDS-FXD achieves the same advantage over FDS-TWK from $N = 60$ to $N = 100$. Fig. 6.10(e) shows DQN maintaining stable served energy across all values of N. CTDE and FDS both decline consistently, with FDS requiring the most energy and CTDE the middle amount from $N = 20$ to $N = 40$. CTDE and FDS both fall below DQN's energy from $N = 60$ onward. This decline reflects the same fixed-capacity constraint, since more of each UAV's limited slots fill with genuine competing devices at higher N and the shared allocation budget spreads across more devices as each receives a smaller share. The declining energy therefore reflects resource starvation rather than efficiency and is consistent with the reward and latency degradation in Fig. 6.10(a) and Fig. 6.10(d). Lastly, Fig. 6.10(f) shows DQN achieving the lowest trust exclusion rate at $N = 20$ and $N = 40$ before rising steadily



(a) Reward (b) Offloading ratio (c) Load balancing



(d) Served latency (e) Served energy (f) Trust exclusion rate

Figure 6.10: Sensitivity to the number of IoT devices N, for $T_n^{\text{THR}} \in [0.1, 0.9]$, $M = 4$ UAVs, $K = 200$ time slots.

as N increases further. CTDE and FDS both start with higher exclusion than DQN but fall below it from around $N = 60$ onward, despite a slight increase between $N = 80$ and $N = 100$.

6.5 Summary

This chapter presented a trust-driven framework for aerial edge computing under IoT device mobility, where UAV trustworthiness spanning energy sustainability, task reliability, and service freshness was updated at every time slot as device positions evolved. The joint offloading and resource allocation problem was formulated as a mixed-integer nonlinear program admitting an IoT-UAV pair only when the UAV satisfied the threshold its device declared, and a two-timescale decomposition separated trust-weighted repositioning from slot-level allocation. Three RL schemes of decreasing centralization resolved the faster decisions, where centralized learning attained the highest reward while decentralized execution sustained higher mean trust among served devices. Trust-weighted deployment outperformed RND and FXD placement across all three architectures. These findings indicate a trade-off between policy performance and trust assurance that architecture selection must weigh. The practical choice therefore reduces to a question of what a central failure would cost. DQN's centralized design carries low resilience to such a failure, an acceptable risk where reward is the dominant concern and infrastructure remains reliable. CTDE and FDS instead offer greater resilience to this failure, sustaining higher trust at execution, while CTDE alone retains much of DQN's training-time performance. Disaster-affected deployments, where no dependable central infrastructure can be assumed, are better served by this resilience than by DQN's reward advantage.

Chapter 7

Conclusion

6G networks extend computing service into regions beyond the reach of terrestrial infrastructure, and UAVs carrying edge servers deliver that service on demand. Their bounded energy and computing capacity, together with links that vary with position, determine what each UAV completes once a task is assigned to it. This thesis developed a trust-aware framework that measured those conditions and applied the resulting score as a condition on IoT-UAV association. Four settings carried this treatment, spanning static deployments, uncertain IoT device state, differentiated task demand, and mobile IoT populations served under varying control architectures.

This thesis first established trust as a binding condition on IoT-UAV association. Multi-metric evaluation at the HAP layer distinguished UAVs that coverage and capacity alone would rank equally. The proposed PGO algorithm attained near-optimal connectivity at substantially lower computational cost than the branch-and-bound benchmark, demonstrating that the trust condition holds without penalty to solution quality.

A second framework extended the treatment to IoT devices reporting uncertain state. Worst-case evaluation over perturbation bounds, combined with harvesting support for devices below the offloading requirement, sustained associations that reported values alone would have left exposed. Results under intelligent and random UAV deployment confirmed that the proposed algorithm tracked the benchmark algorithm as perturbation increased.

A third framework brought task priority alongside trust within the same IoT device admission decision. Trust determined which UAVs qualified to serve an IoT device, while priority ordered the devices competing for the available capacity. The proposed PGO algorithm achieved connectivity close to the benchmark as the multi-criteria weighting shifted across trust, priority, cost, and latency.

Finally, this thesis introduced a reinforcement learning framework for evaluating trust-aware operation across control architectures. Per-slot trust evaluation over energy sustainability, task reliability, and service freshness supported trust-weighted deployment that repositioned UAVs as demand shifted under IoT mobility. Centralized learning attained the highest reward while decentralized execution sustained higher mean trust and lower trust exclusion, indicating a trade-off between policy performance and trust assurance.

Overall, this thesis treated trust as a constraint within the optimization rather than as an assessment reported alongside it. Aggregation tightened across the frameworks, from additive weighting through a hybrid form to multiplicative combination, so that a strong metric could no longer offset a weak one. Simulation across the four settings showed near-optimal connectivity under static conditions, uncertain IoT device state, competing task demands, and IoT mobility. The trust condition therefore reshapes UAV eligibility across the devices requesting service, without displacing the objectives an aerial deployment carries.

7.1 Future Research Directions

Building on the findings of this thesis, several promising avenues remain open for future investigation:

- **Chance-Constrained Trust Formulations:** The frameworks in this thesis bound uncertainty through worst-case realizations, so an IoT device fails admission whenever any realization within its bound would leave the association infeasible. Chance-constrained formulations replace this with a probabilistic guarantee [60, 99], allowing a deployment to specify how much risk each association may carry.

- **Uncertainty Beyond IoT Positions:** This thesis incorporates uncertainty pertaining to IoT device positions and residual energy levels. This treatment can be extended to UAV positions, channel estimates, trust measurements, and the UAV-HAP link, each currently entering the optimization as a known quantity. This broader scope requires evaluating the accuracy of each quantity at the moment it is used. Modeling this estimation uncertainty would show whether trust-aware association holds once every hierarchical layer carries imprecision, not only the IoT devices.

- **Resilience Across the Network Hierarchy:** This thesis assumes that every IoT device, UAV, and HAP remains operational throughout a deployment. Failure at each tier instead disrupts whatever that tier is meant to serve. IoT devices that stop reporting withhold the situational awareness disaster response depends on. UAV unavailability strands the devices within its coverage until reassignment. HAP failure disrupts trust evaluation and coordination for every UAV it oversees. Extending the framework to detect failure at each tier and reassign affected devices would show how quickly service recovers.

- **Digital Twin Support for Trust Evaluation:** Trust scores in this thesis hold their value between reporting intervals, while the UAV conditions they describe continue to change. A HAP-maintained digital twin would track those conditions between reports, though preliminary work shows that deviation between the twin and the physical UAV affects performance [102]. Representing this deviation within the trust model is the open problem, since the score must reflect both what the twin estimates and how far that estimate may drift.

- **Contention-Aware Scheduling at UAV Edge Servers:** The latency models in this thesis compute task completion from allocated resources, without representing the order in which admitted tasks reach the processor. Devices admitted to the same UAV therefore receive completion estimates that assume no waiting. Queuing formulations that model integration, offloading, and processing stages separately [125] would connect the trust condition to the delay each task experiences under contention.

- **Propulsion-Aware Energy Optimization:** A UAV's total energy budget in this thesis covers both propulsion and MEC service, though only the service-related component enters the optimization. Propulsion remains fixed at a constant baseline, so UAV movement carries no cost within current deployment and association decisions. Propulsion energy could instead be expressed as a function of UAV velocity and flight mode, replacing the constant baseline within the objective. Such a formulation would likely favor configurations with less repositioning, reshaping which IoT devices receive support and how often UAV positions are revised.

- **Quantum-Inspired Optimization for Larger Deployments:** The mixed-integer structure underlying trust-aware association grows harder as IoT device populations expand, and PGO reaches tractability by relaxing the binary variables before a penalty term recovers them. Quantum-inspired methods offer an alternative route, reformulating combinatorial problems for real-time solution on classical hardware [72,126]. Their reported use in UAV trajectory planning [127] and edge resource management [128] covers the placement and allocation decisions this thesis addresses, which makes trust-constrained association a natural extension.

- **Hardware Validation of Trust Metrics:** The trust models developed in this thesis draw on residual energy, computing load, link quality, and task completion, each computed from simulated values. Deployment on embedded UAV platforms would establish whether these quantities can be measured at the rate trust evaluation requires, and whether the resulting scores separate platforms as the simulation results indicate.

Closing Remarks

Overall, this thesis provides a trust-aware foundation for aerial edge computing in 6G networks. Aerial deployments place computing resources where terrestrial infrastructure cannot reach, and they do so through UAVs whose condition changes throughout the service period. IoT device assignment on coverage and capacity alone leaves that condition unexamined. The frameworks developed here bring a measured account of UAV behavior into the assignment, holding it across uncertainty in reported device state, competing task demands, IoT mobility, and varying degrees of control centralization. Trust in this form gives an aerial network grounds for the commitments it makes to the IoT devices it serves.

Bibliography

[1] I. Ishteyaq, K. Muzaffar, N. Shafi, and M. A. Alathbah, "Unleashing the power of tomorrow: Exploration of next frontier with 6G networks and cutting edge technologies," *IEEE Access*, vol. 12, pp. 29445–29463, Feb. 2024.

[2] 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, Secondquarter 2023.

[3] X. Xia, S. M. M. Fattah, and M. A. Babar, "A survey on UAV-enabled edge computing: Resource management perspective," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–36, Sep. 2023.

[4] Z. Ning, H. Hu, X. Wang, L. Guo, S. Guo, G. Wang, and X. Gao, "Mobile edge computing and machine learning in the internet of unmanned aerial vehicles: A survey," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–31, Aug. 2023.

[5] G. Karabulut Kurt, M. G. Khoshkholgh, S. Alfattani, A. Ibrahim, T. S. J. Darwish, M. S. Alam, H. Yanikomeroglu, and A. Yongacoglu, "A vision and framework for the high altitude platform station (HAPS) networks of the future," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 729–779, Mar. 2021.

[6] N.-N. Dao, Q.-V. Pham, N. H. Tu, T. T. Thanh, V. N. Q. Bao, D. S. Lakew, and S. Cho, "Survey on aerial radio access networks: Toward a comprehensive 6G access

infrastructure," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1193–1225, Secondquarter, 2021.

[7] G. K. Pandey, D. S. Gurjar, H. H. Nguyen, and S. Yadav, "Security threats and mitigation techniques in UAV communications: A comprehensive survey," *IEEE Access*, vol. 10, pp. 112858–112897, Oct. 2022.

[8] Y. Liu, Z. He, J. Liang, Z. Li, and Q. Deng, "Multidimensional trust evaluation and task match based workers recruitment scheme for MCS," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–17, Jan. 2026.

[9] J. Kundu, S. Alam, J. C. Das, A. Dey, and D. de, "Trust-based flying Ad Hoc network: A survey," *IEEE Access*, vol. 12, pp. 99258–99281, Jun. 2024.

[10] H. Kang, X. Chang, J. Mišić, V. B. Mišić, J. Fan, and Y. Liu, "Cooperative UAV resource allocation and task offloading in hierarchical aerial computing systems: A MAPPO-based approach," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10497–10509, Jun. 2023.

[11] I. Ud Din, I. Taj, A. Almogren, and M. Guizani, "Federated learning for trust enhancement in UAV-enabled IoT networks: A unified approach," *IEEE Internet of Things Journal*, vol. 12, no. 8, pp. 9596–9603, Apr. 2025.

[12] J. Bai, Z. Zeng, T. Wang, S. Zhang, N. N. Xiong, and A. Liu, "TANTO: An effective trust-based unmanned aerial vehicle computing system for the Internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 5644–5661, Apr. 2023.

[13] J. Bai, J. Gui, G. Huang, M. Dong, T. Wang, S. Zhang, and A. Liu, "A low-cost UAV task offloading scheme based on trustable and trackable data routing," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 9, pp. 5797–5812, Sep. 2024.

[14] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, Mar. 2004.

[15] L. A. Wolsey, *Integer programming.* Hoboken, NJ, USA: John Wiley & Sons, 2nd ed., Sep. 2020.

[16] J. Chen, X. Cao, P. Yang, M. Xiao, S. Ren, Z. Zhao, and D. O. Wu, "Deep reinforcement learning based resource allocation in multi-UAV-aided MEC networks," *IEEE Transactions on Communications*, vol. 71, no. 1, pp. 296–309, Jan. 2023.

[17] W. Melo, M. Fampa, and F. Raupp, "Integrality gap minimization heuristics for binary mixed integer nonlinear programming," *Journal of Global Optimization*, vol. 71, no. 3, pp. 593–612, Feb. 2018.

[18] M. Yahya, M. Naeem, Z. Kaleem, A. H. Alenezi, and W. Ejaz, "Robust multicriterion offloading in digital-twin-assisted UAV networks," *IEEE Internet of Things Journal*, vol. 12, no. 2, pp. 1643–1654, Jan. 2025.

[19] M. Yahya, M. Naeem, and W. Ejaz, "Digital twin UAV networks with IoT spatial perturbations: Robust offloading framework," in *36th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, Istanbul, Turkiye, Sep. 2025.

[20] N. Yamin and G. Bhat, "Uncertainty-aware energy harvest prediction and management for IoT devices," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 5, pp. 1–33, Sep. 2023.

[21] J. Meng, H. Tan, X.-Y. Li, Z. Han, and B. Li, "Online deadline-aware task dispatching and scheduling in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1270–1286, Jun. 2020.

[22] H. Hao, C. Xu, W. Zhang, S. Yang, and G.-M. Muntean, "Joint task offloading, resource allocation, and trajectory design for multi-UAV cooperative edge computing with task priority," *IEEE Transactions on Mobile Computing*, vol. 23, no. 9, pp. 8649–8663, Sep. 2024.

[23] R. Khalid, Z. Shah, M. Naeem, A. Ali, A. Al-Fuqaha, and W. Ejaz, "Computational efficiency maximization for UAV-assisted MEC networks with energy harvesting in disaster scenarios," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 9004–9018, Mar. 2024.

[24] J. You, Z. Jia, C. Dong, Q. Wu, and Z. Han, "Joint computation offloading and resource allocation for uncertain maritime MEC via cooperation of AAVs and vessels," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 11, pp. 18081–18095, Nov. 2025.

[25] L. Liu, H. Miao, C. Qu, Z. Wang, H. Zhang, and Z. Li, "Multi-UAV path planning for mobile edge computing with high-density mobile devices," *IEEE Transactions on Mobile Computing*, vol. 25, no. 7, pp. 10260–10273, Jul. 2026.

[26] L. Wang, B. Shen, L. Ma, Y. Zhang, Y. Zhao, H. Guo, Z. Yu, and B. Guo, "Joint task offloading and migration optimization in UAV-enabled dynamic MEC networks," *IEEE Transactions on Services Computing*, vol. 18, no. 4, pp. 2143–2157, Aug. 2025.

[27] Y. Nie, J. Zhao, F. Gao, and F. R. Yu, "Semi-distributed resource management in UAV-aided MEC systems: A multi-agent federated reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13162–13173, Dec. 2021.

[28] S. Hwang, H. Lee, M. Kim, and I. Lee, "Multiagent deep reinforcement learning for decentralized multi-UAV mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 14484–14497, May 2025.

[29] W. Zhao, S. Cui, W. Qiu, Z. He, Z. Liu, X. Zheng, B. Mao, and N. Kato, "A survey on DRL-based UAV communications and networking: DRL fundamentals, applications and implementations," *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 3911–3941, 2026.

[30] X. Lyu, A. Baisero, Y. Xiao, B. Daley, and C. Amato, "On centralized critics in multi-agent reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 77, pp. 295–354, May 2023.

[31] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of things," *Journal of network and computer applications*, vol. 42, pp. 120–134, Jun. 2014.

[32] B. Veith, D. Krummacker, and H. D. Schotten, "The road to trustworthy 6G: A survey on trust anchor technologies," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 581–595, Feb. 2023.

[33] Y. Liu, Z. He, A. Liu, X. Dai, Q. Deng, and Z. Li, "TDI: A trust-based distributed incentive scheme to promote information propagation," *IEEE Transactions on Mobile Computing*, pp. 1–16, Dec. 2025.

[34] W. Kong, X. Li, L. Hou, J. Yuan, Y. Gao, and S. Yu, "A reliable and efficient task offloading strategy based on multifeedback trust mechanism for IoT edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13927–13941, Aug. 2022.

[35] Y. Ruan, A. Durresi, and S. Uslu, "Trust assessment for Internet of things in multi-access edge computing," in *IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 1155–1161, Krakow, Poland, May, 2018.

[36] P. Abeysekara, H. Dong, and A. K. Qin, “Edge intelligence for real-time IoT service trust prediction,” *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2606–2619, Aug. 2023.

[37] A. Samanta and T. G. Nguyen, “Trust-based distributed resource allocation in edge-enabled IIoT networks,” *IEEE Access*, vol. 13, pp. 79694–79704, May 2025.

[38] S. Tsikteris, A. B. Rahman, M. S. Siraj, and E. E. Tsiropoulou, “Trust-me: Trust-based resource allocation and server selection in multi-access edge computing,” *Future Internet*, vol. 16, no. 8.

[39] M. A. Al-Tarawneh, H. Kanj, and W. H. F. Aly, “An integrated MCDM framework for trust-aware and fair task offloading in heterogeneous multi-provider edge-fog-cloud systems,” *Results in Engineering*, vol. 26, p. 105228, Jun. 2025.

[40] J. Wang, Z. Li, H. Liu, T. Qiu, and H. Luo, “A trust-based computation offloading framework in mobile cloud-edge computing networks,” *IEEE Transactions on Mobile Computing*, vol. 24, no. 6, pp. 5370–5385, Jun. 2025.

[41] J. Chen, X. Wang, and X. Shen, “Goal-driven trusted collaborator selection and task offloading in dynamic collaborative systems,” *IEEE Internet of Things Journal*, vol. 12, no. 7, pp. 8537–8551, Apr. 2025.

[42] C. Sun, G. Fontanesi, B. Canberk, A. Mohajerzadeh, S. Chatzinotas, D. Grace, and H. Ahmadi, “Advancing UAV communications: A comprehensive survey of cutting-edge machine learning techniques,” *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 825–854, May 2024.

[43] A. A. Baktayan, A. Thabit Zahary, and I. Ahmed Al-Baltah, “A systematic mapping study of UAV-enabled mobile edge computing for task offloading,” *IEEE Access*, vol. 12, pp. 101936–101970, Jul. 2024.

[44] K. Kuru, “Planning the future of smart cities with swarms of fully autonomous unmanned aerial vehicles using a novel framework,” *IEEE Access*, vol. 9, pp. 6571–6595, Jan. 2021.

[45] J. Guo, G. Huang, Q. Li, N. N. Xiong, S. Zhang, and T. Wang, “STMTO: A smart and trust multi-UAV task offloading system,” *Information Sciences*, vol. 573, pp. 519–540, Sep. 2021.

[46] J. Zheng, J. Xu, H. Du, D. Niyato, J. Kang, J. Nie, and Z. Wang, “Trust management of tiny federated learning in Internet of unmanned aerial vehicles,” *IEEE Internet of Things Journal*, vol. 11, no. 12, pp. 21046–21060, Jun. 2024.

[47] H. Li, X. Li, F. Dunkin, Z. Zhang, and X. Lu, “Adaptive multi-granularity trust management scheme for UAV visual sensor security under adversarial attacks,” *Computers & Security*, vol. 148, p. 104108, Jan. 2025.

[48] J. Guo, A. Liu, K. Ota, M. Dong, X. Deng, and N. N. Xiong, “ITCN: An intelligent trust collaboration network system in IoT,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 203–218, Feb. 2022.

[49] W. Mo, W. Liu, G. Huang, N. N. Xiong, A. Liu, and S. Zhang, “A cloud-assisted reliable trust computing scheme for data collection in Internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4969–4980, Jul. 2022.

[50] S. Li, B. Lu, L. Ale, H. Chen, F. Tan, and J. Huang, “Two-hop partial task offloading and resource allocation in air-ground integrated mobile edge computing network: A DRL-based method,” *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 21443–21456, Jun. 2025.

[51] W. Xu, T. Zhang, X. Mu, Y. Liu, and Y. Wang, “Trajectory planning and resource allocation for multi-UAV cooperative computation,” *IEEE Transactions on Communications*, vol. 72, no. 7, pp. 4305–4318, Jul. 2024.

[52] Y. K. Tun, G. Dán, Y. M. Park, and C. S. Hong, "Joint UAV deployment and resource allocation in THz-assisted MEC-enabled integrated space-air-ground networks," *IEEE Transactions on Mobile Computing*, vol. 24, no. 5, pp. 3794–3808, May 2025.

[53] Y. Gao, P. Wu, X. Yuan, Y. Hu, X. Cao, and A. Schmeink, "Joint UAV 3D deployment and ground device association optimizing for multi-UAV-aided MEC heterogeneous network," *IEEE Transactions on Mobile Computing*, pp. 1–15, Jan. 2026.

[54] G. Sun, Y. Wang, Z. Sun, Q. Wu, J. Kang, D. Niyato, and V. C. M. Leung, "Multi-objective optimization for multi-UAV-assisted mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 14803–14820, Dec. 2024.

[55] X. Yu, X. Zhang, Y. Rui, K. Wang, X. Dang, and M. Guizani, "Joint resource allocations for energy consumption optimization in HAPS-aided MEC-NOMA systems," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 12, pp. 3632–3646, Dec. 2024.

[56] H. Guo, Y. Wang, J. Liu, and C. Liu, "Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond," *IEEE Transactions on Wireless Communications*, vol. 23, no. 1, pp. 347–359, Jan. 2024.

[57] P. Chen, X. Luo, D. Guo, Y. Sun, J. Xie, Y. Zhao, and R. Zhou, "Secure task offloading for MEC-aided-UAV system," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 5, pp. 3444–3457, May 2023.

[58] M. I. Mushtaq, O. Chughtai, Y. Zhang, A. H. Alenezi, and N. Alqahtani, "Framework for optimized resource allocation in multi-user, multi-service, multi-device aerial networks," *IEEE Access*, vol. 12, pp. 54866–54878, Apr. 2024.

[59] A. He, H. Pan, Y. Dai, X. Si, C. Yuen, and Y. Zhang, "ADMM for mobile edge intelligence: A survey," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 5, pp. 3020–3057, Oct. 2025.

[60] M. Abdalla, A. Islam, M. Ali, and A. Hendawi, “Framework to process vehicles uncertain locations for intelligent transportation,” *Intelligent Decision Technologies*, vol. 18, no. 3, pp. 2097–2113, Aug. 2024.

[61] M. Yahya, M. Naeem, Z. Kaleem, and W. Ejaz, “Robust priority aware multi-criterion offloading in digital twin UAVs networks,” *Elsevier Internet of Things*, vol. 34, no. 101763, Nov. 2025.

[62] M. Basharat, M. Naeem, A. M. Khattak, and A. Anpalagan, “Digital-twin-assisted task offloading in UAV-MEC networks with energy harvesting for IoT devices,” *IEEE Internet of Things Journal*, vol. 11, no. 23, pp. 37550–37561, Dec. 2024.

[63] M. Basharat, M. Naeem, A. M. Khattak, and A. Anpalagan, “Efficient offloading in UAV-MEC IoT networks: Leveraging digital twins and energy harvesting,” in *IEEE Conference on Artificial Intelligence (CAI)*, pp. 464–469, IEEE, Singapore, Jun. 2024.

[64] N. H. Nguyen, V.-D. Nguyen, A. T. Nguyen, N. V. Thieu, H. N. Nguyen, and S. Chatzinotas, “Deadline-aware joint task scheduling and offloading in mobile-edge computing systems,” *IEEE Internet of Things Journal*, vol. 11, no. 20, pp. 33282–33295, Oct. 2024.

[65] S. Sharif, S. Manzoor, and W. Ejaz, “Priority-based resource optimisation and user association in integrated networks,” *IET Networks*, vol. 14, no. 1, p. e12140, Jan. 2025.

[66] H. Zhao, Z. Sha, Q. Wang, X. Wang, M. Liu, B. Xu, and H. Zhu, “DQN-JUTO: Joint optimization of UAV trajectory and task offloading using deep Q-Networks for post-disaster edge computing,” *IEEE Transactions on Vehicular Technology*, pp. 1–15, Jul. 2026.

[67] M. Hevesli, A. Mohammed Seid, A. Erbad, and M. Abdallah, “Multi-agent DRL for queue-aware task offloading in hierarchical MEC-enabled air-ground networks,” *IEEE*

Transactions on Cognitive Communications and Networking, vol. 12, pp. 217–236, 2026.

[68] Z. Sun, G. Sun, Q. Wu, L. He, S. Liang, H. Pan, D. Niyato, C. Yuen, and V. C. M. Leung, "TJCCT: A two-timescale approach for UAV-assisted mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 24, no. 4, pp. 3130–3147, Apr. 2025.

[69] B. Li, R. Yang, L. Liu, J. Wang, N. Zhang, and M. Dong, "Robust computation offloading and trajectory optimization for multi-UAV-assisted MEC: A multiagent DRL approach," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4775–4786, Feb. 2024.

[70] D. Rizvi and D. Boyle, "Multi-agent reinforcement learning with action masking for UAV-enabled mobile communications," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 3, pp. 117–132, Dec. 2024.

[71] M. O. Butt, L. Ferdouse, and W. Ejaz, "Bridging the digital divide: 6G networks and enabling technologies for ubiquitous coverage," in *IEEE International Conference on Advanced Telecommunication and Networking Technologies (ATNT)*, vol. 1, pp. 1–4, Johor Bahru, Malaysia, Sep. 2024.

[72] M. O. Butt, N. Waheed, T. Q. Duong, and W. Ejaz, "Quantum-inspired resource optimization for 6G networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 5, pp. 2973–3019, Oct. 2025.

[73] M. Liaq and W. Ejaz, "Computational offloading and delay minimization for UAV-aided edge federated learning," in *IEEE International Conference on Communications (ICC)*, pp. 1292–1297, Denver, CO, USA, Jun. 2024.

[74] R. Zhang, W. Wu, X. Chen, Z. Gao, and Y. Cai, "Terahertz integrated sensing and communication-empowered UAVs in 6G: A transceiver design perspective," *IEEE Vehicular Technology Magazine*, vol. 21, no. 1, pp. 71–80, Mar. 2026.

[75] T. Wang and C. You, "Adaptive uplink scheduling and UAV association in UAV-assisted OFDMA cellular networks: A game-theoretical approach," *IEEE Access*, vol. 12, pp. 63504–63514, May 2024.

[76] Y. Li, D. V. Huynh, T. Do-Duy, E. Garcia-Palacios, and T. Q. Duong, "Unmanned aerial vehicle-aided edge networks with ultra-reliable low-latency communications: A digital twin approach," *IET Signal Processing*, vol. 16, no. 8, pp. 897–908, Apr. 2022.

[77] Y. Zhang, M. A. Kishk, and M.-S. Alouini, "Energy-efficient optimization in aerial IAB networks for emergency communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 2, pp. 4614–4626, Apr. 2025.

[78] C. Han, Y. Wang, Y. Li, Y. Chen, N. A. Abbasi, T. Kurner, and A. F. Molisch, "Terahertz wireless channels: A holistic survey on measurement, modeling, and analysis," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1670–1707, Thirdquarter, 2022.

[79] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[80] A. Hussain, S. Li, T. Hussain, R. W. Attar, F. Ali, A. Alhomoud, and B. Shah, "Integrated energy-efficient distributed link stability algorithm for UAV networks," *Computers Materials & Continua*, vol. 81, no. 2, pp. 2357–2394, Nov. 2024.

[81] M. Jones, S. Djahel, and K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–39, Feb. 2023.

[82] X. Gao, X. Zhu, and L. Zhai, "Minimization of aerial cost and mission completion time in multi-UAV-enabled IoT networks," *IEEE Transactions on Communications*, vol. 71, no. 9, pp. 5335–5347, Sep. 2023.

[83] J. Liu, P. Xie, K. Lin, X. Tu, and R. Fan, “Trust-aware task offloading for cost-effective UAV-based edge computing based on reinforcement learning,” *Neural Computing and Applications*, vol. 37, no. 20, pp. 14765–14782, Jul. 2025.

[84] Z. Fang, J. Wang, J. Liang, Y. Yan, D. Pi, H. Zhang, and G. Yin, “Authority allocation strategy for shared steering control considering human-machine mutual trust level,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2002–2015, Jan. 2024.

[85] L. Zhou, M. Zhang, Q. Wei, Y. Jin, S. Wang, and J. Yan, “Energy-distance-function-based improved K-means for clustering routing algorithm,” *IEEE Internet of Things Journal*, vol. 11, no. 22, pp. 36763–36774, Nov. 2024.

[86] A. A. Bushra and G. Yi, “Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms,” *IEEE Access*, vol. 9, pp. 87918–87935, Jun. 2021.

[87] X. Tang, L. Yang, D. Wang, W. Li, D. Xin, and H. Jia, “A collaborative navigation algorithm for UAV Ad Hoc network based on improved sequence quadratic programming and unscented Kalman filtering in GNSS denied area,” *Measurement*, vol. 242, p. 115977, Jan. 2025.

[88] Y. Chen, H. Kang, J. Li, G. Sun, B. Wang, J. Wang, C. Liang, S. Liang, and D. Niyato, “Joint resource management for energy-efficient UAV-assisted SWIPT-MEC: A deep reinforcement learning approach,” *IEEE Internet of Things Journal*, vol. 12, no. 15, pp. 31448–31465, Aug. 2025.

[89] Z. A. Haider, I. Ullah, A. Abdusalomov, M. Shah, M. Z. Khan, and B. A. Zneid, “Edge-intelligent semantic aggregation in blockchain-secured 6G UAV-assisted Internet of vehicles,” *Journal of Electronic Science and Technology*, vol. 24, no. 1, p. 100350, Mar. 2026.

[90] C. Zhou, W. Wu, H. He, P. Yang, F. Lyu, N. Cheng, and X. Shen, "Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 911–925, Feb. 2021.

[91] S. Ruiz-Villafranca, J. Roldán-Gómez, J. Carrillo-Mondéjar, J. M. C. Gomez, and J. M. Villalon, "A MEC-IIoT intelligent threat detector based on machine learning boosted tree algorithms," *Computer Networks*, vol. 233, p. 109868, Sep. 2023.

[92] E. A. Elsayed, *Reliability engineering*. Hoboken, NJ, USA: Wiley, 3rd ed., Nov. 2020.

[93] M. O. Butt, M. Yahya, M. Naeem, and W. Ejaz, "Toward trustworthy aerial networks: Modeling IoT spatial and residual energy uncertainties," *IEEE Internet of Things Journal*, Sep. 2026.

[94] Y. Wang, Z. Su, Q. Xu, R. Li, T. H. Luan, and P. Wang, "A secure and intelligent data sharing scheme for UAV-assisted disaster rescue," *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, pp. 2422–2438, Dec. 2023.

[95] M. Adil, M. A. Jan, Y. Liu, H. Abulkasim, A. Farouk, and H. Song, "A systematic survey: Security threats to UAV-aided IoT applications, taxonomy, current challenges and requirements with future research directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 1437–1455, Feb. 2023.

[96] M. Huang, A. Liu, N. N. Xiong, and J. Wu, "A UAV-assisted ubiquitous trust communication system in 5G and beyond networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3444–3458, Nov. 2021.

[97] I. A. Ridhawi and M. Aloqaily, "Zero-trust UAV-enabled and DT-supported 6G networks," in *IEEE Global Communications Conference (GLOBECOM)*, pp. 6171–6176, Kuala Lumpur, Malaysia, Dec. 2023.

[98] Y. Zeng and J. Tang, "MEC-assisted real-time data acquisition and processing for UAV with general missions," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1058–1072, Jan. 2023.

[99] M. O. Butt, M. Naeem, and W. Ejaz, "A chance-constrained optimization framework for reliable aerial networking," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Kuala Lumpur, Malaysia, Apr. 2026.

[100] M. O. Butt, M. Naeem, and W. Ejaz, "UAV-assisted resource optimization for priority-aware and trustworthy services," in *ACM MobiCom International Workshop on Drone-Assisted Wireless Communications for 5G and Beyond (DroneCom)*, Austin, TX, USA, Oct. 2026.

[101] M. O. Butt, M. Naeem, and W. Ejaz, "Trust-driven multi-criteria optimization for UAV-assisted IoT networks," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) Workshops*, Istanbul, Türkiye, Sep. 2025.

[102] M. O. Butt, M. Naeem, and M. Basharat, "Digital twin-enabled trustworthy aerial edge computing for 6G networks," in *IEEE Global Communications Conference (GLOBECOM) Workshops*, Taipei, Taiwan, Dec. 2025.

[103] E. T. Michailidis, K. Maliatsos, D. N. Skoutas, D. Vouyioukas, and C. Skianis, "Secure UAV-aided mobile edge computing for IoT: A review," *IEEE Access*, vol. 10, pp. 86353–86383, Aug. 2022.

[104] Z. Yang, J. Yang, Z. Li, L. Zhu, Z. Xiao, and Z. Han, "Joint network configuration and resource management in autonomous UAV-assisted MEC systems via embodied agentic AI," *IEEE Transactions on Cognitive Communications and Networking*, vol. 12, pp. 8372–8386, May 2026.

[105] X. Li, Y. Qin, J. Huo, and W. Huangfu, "Computation offloading and trajectory planning of multi-UAV-enabled MEC: A knowledge-assisted multiagent reinforcement

learning approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 5, pp. 7077–7088, May 2024.

[106] Z. Ren, X. Li, Y. Miao, Z. Li, Z. Wang, M. Zhu, X. Liu, and R. H. Deng, "Intelligent adaptive gossip-based broadcast protocol for UAV-MEC using multi-agent deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 6563–6578, Jun. 2024.

[107] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[108] T. Khurshid, W. Ahmed, R. Ahmad, M. M. Alam, and J. J. Rodrigues, "Distributed dynamic consensus (DDC) protocol for multi-UAV 3D trajectory planning and resource allocation," *Vehicular Communications*, vol. 56, no. 100969, Dec. 2025.

[109] L. Zhong, Y. Liu, X. Deng, C. Wu, S. Liu, and L. T. Yang, "Distributed optimization of multi-role UAV functionality switching and trajectory for security task offloading in UAV-assisted MEC," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 12, pp. 19432–19447, Dec. 2024.

[110] C. Zhu, G. Zhang, and K. Yang, "Fairness-aware task loss rate minimization for multi-UAV enabled mobile edge computing," *IEEE Wireless Communications Letters*, vol. 12, no. 1, pp. 94–98, Jan. 2023.

[111] S. Shakoor, Z. Kaleem, D.-T. Do, O. A. Dobre, and A. Jamalipour, "Joint optimization of UAV 3-D placement and path-loss factor for energy-efficient maximal coverage," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9776–9786, Jun. 2021.

[112] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network," *IEEE*

Transactions on Network and Service Management, vol. 18, no. 4, pp. 4531–4547, Dec. 2021.

[113] S. F. Abedin, M. S. Munir, N. H. Tran, Z. Han, and C. S. Hong, "Data freshness and energy-efficient UAV navigation optimization: A deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5994–6006, Sep. 2021.

[114] X. Cao, J. Xu, and R. Zhang, "Mobile edge computing for cellular-connected UAV: Computation offloading and trajectory optimization," in *IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, Kalamata, Greece, Jun. 2018.

[115] X.-Y. Yu, W.-J. Niu, Y. Zhu, and H.-B. Zhu, "UAV-assisted cooperative offloading energy efficiency system for mobile edge computing," *Digital Communications and Networks*, vol. 10, no. 1, pp. 16–24, Feb. 2024.

[116] Y. Ouyang, W. Liu, Q. Yang, X. Mao, and F. Li, "Trust-based task offloading scheme in UAV-enhanced edge computing network," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 3268–3290, Apr. 2021.

[117] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, May 2021.

[118] L. Bao, J. Luo, H. Bao, Y. Hao, and M. Zhao, "Cooperative computation and cache scheduling for UAV-enabled MEC networks," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 2, pp. 965–978, Jun. 2022.

[119] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[120] Y. Ding, H. Han, W. Lu, Y. Wang, N. Zhao, X. Wang, and X. Yang, "DDQN-based trajectory and resource optimization for UAV-aided MEC secure communications," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 6006–6011, Apr. 2024.

[121] Y. Wang, Y. He, G. Liu, and K. Li, "Cost-optimized periodic DAG-structured task offloading in multi-user MEC systems using reinforcement learning," *ACM Transactions on Internet Technology*, vol. 26, no. 1, pp. 1–27, Jan. 2026.

[122] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, Stockholm, Sweden, Jul. 2018.

[123] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, vol. 30, California, USA, Dec. 2017.

[124] S. Qi, B. Lin, Y. Deng, X. Chen, and Y. Fang, "Minimizing maximum latency of task offloading for multi-UAV-assisted maritime search and rescue," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 9, pp. 13625–13638, Sep. 2024.

[125] M. Liaq and W. Ejaz, "Delay optimization in hierarchical UAV-aided federated learning for straggling IoT devices," *IEEE Internet of Things Journal*, vol. 13, no. 9, pp. 18334–18350, May 2026.

[126] T. Q. Duong, L. D. Nguyen, B. Narottama, J. A. Ansere, D. Van Huynh, and H. Shin, "Quantum-inspired real-time optimization for 6G networks: Opportunities, challenges, and the road ahead," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1347–1359, Aug. 2022.

[127] Y. Li, A. H. Aghvami, and D. Dong, "Intelligent trajectory planning in UAV-mounted wireless networks: A quantum-inspired reinforcement learning perspective," *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 1994–1998, Sep. 2021.

[128] D. Wang, B. Song, P. Lin, F. R. Yu, X. Du, and M. Guizani, "Resource management for edge intelligence (EI)-assisted IoV using quantum-inspired reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12588–12600, Jul. 2022.