

Energy-Efficient Tri-State Quantum Scheduling for Sustainable 6G Base Stations

Himadri Sekhar Mondal

himadrisekhar.mondal@mckvie.edu.in

MCKV Institute of Engineering

Ranjit Haldar

Indian Institute of Information Technology, Kalyani

Asim Kumar Panda

MCKV Institute of Engineering

Balaram Ghosal

Indian Institute of Engineering Science and Technology, Shibpur

Research Article

Keywords: 6G networks, Green communications, Base station sleep scheduling, Energy efficiency, Quantum-inspired optimization, Tri-state power control, Radio access networks, Sustainable networking

Posted Date: October 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11096037/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Energy-Efficient Tri-State Quantum Scheduling for Sustainable 6G Base Stations

Himadri Sekhar Mondal* Ranjit Halder† Asim Kumar Panda‡
Balaram Ghosal§

September 7, 2026

Abstract

The rapid densification of sixth-generation (6G) radio access networks is expected to increase the operational energy demand of cellular infrastructure, making energy-aware base station (BS) sleep scheduling an important research challenge. Conventional BS control mechanisms commonly rely on binary ON/OFF operation or static threshold policies, which provide limited flexibility in balancing energy consumption and service responsiveness under time-varying traffic. To address this challenge, this paper proposes a *Tri-State Quantum Rotation Scheduler (TQRS)* for energy-aware 6G BS management. The framework represents each BS using three operating states, namely deep sleep, light sleep, and active mode, and employs a probability-based search mechanism with rotation-inspired updates to explore the resulting scheduling space. A feasibility-repair mechanism is used to maintain service constraints during candidate evaluation, followed by local refinement of the selected scheduling solution. The scheduling objective considers energy consumption and service delay while enforcing zero unserved traffic. The proposed scheduler is evaluated using a frozen 24-slot traffic trace and a common 15-BS network environment. The evaluation includes conventional reference policies, an internal Greedy+Repair heuristic, population-based metaheuristics including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Whale Optimization Algorithm (WOA), and deep reinforcement learning baselines including Proximal Policy Optimization (PPO) and Deep Q-Network (DQN). In the frozen comparative experiment, TQRS consumes 19,053 Wh, corresponding to a 47.0750% reduction relative to the Always Active reference of 36,000 Wh, while maintaining zero unserved traffic and a normalized mean delay of 0.929060, corresponding to approximately 18.5812 ms. WOA and the internal Greedy+Repair heuristic obtain lower energy consumption in this single-run comparison, with 17,692 Wh, and also slightly lower mean delay. Therefore, the study does not claim numerical superiority of TQRS over all competing methods. Instead, the results evaluate TQRS as a feasibility-aware tri-state scheduling framework and examine whether its probability-based search mechanism can provide a useful alternative operating strategy under time-varying traffic. The results also highlight the

*MCKV Institute of Engineering, email: himadrisekhar.mondal@mckvie.edu.in

†Indian Institute of Information Technology, Kalyani, email: ranjit_jrf26@iiitkalyani.ac.in

‡MCKV Institute of Engineering, email: asimkumar.panda@mckvie.edu.in

§IIEST, Shibpur, email: 2023itp002.balaram@students.iiests.ac.in

importance of realistic traffic validation, robustness analysis, and broader network modeling for practical 6G deployment.

Keywords: 6G networks, Green communications, Base station sleep scheduling, Energy efficiency, Quantum-inspired optimization, Tri-state power control, Radio access networks, Sustainable networking

1 Introduction

The sixth-generation (6G) wireless network is established to support massive connectivity. It must serve ultra-low latency, intelligent services, and immersive applications. These features require a dense deployment of radio access infrastructure [1,2]. As a result, the number of base stations (BSs) will increase significantly [2–4]. This growth improves coverage and capacity, but it also raises network energy consumption [5,6]. Among all network components, BSs consume a major share of operational power. Many BSs remain lightly loaded during off-peak hours, night periods, or localized traffic drops [7,8]. However, they often continue operating in active mode. This creates avoidable energy waste and increases operating cost. It also conflicts with the global objective of sustainable and green communication systems [9–11]. Sleep scheduling has emerged as an effective solution to this problem. In this approach, underutilized BSs are switched to low-power states when traffic demand is low. They are reactivated when demand rises [12,13]. Proper sleep control can reduce energy use while preserving service quality. Also it has to mention practical scheduling is not simple. Aggressive sleeping may increase delay, reduce available capacity, and create service instability. Frequent wake-up transitions may also introduce additional energy cost [14]. Most existing studies use binary BS control, where a station is either active or fully sleeping. This assumption is simple, but it does not reflect realistic hardware behavior [15–17]. Modern BS platforms can operate in multiple power modes with different response times. In particular, a light sleep mode can keep partial readiness and allow faster recovery than deep sleep. Therefore, multi-state scheduling is more practical for future 6G systems.

Another challenge is the optimization complexity of dynamic sleep control. The scheduler must decide the operating state of multiple BSs across changing traffic conditions, and the search space grows rapidly with network size. Classical greedy or threshold rules are fast and can be highly effective for a particular workload, but their deterministic search behavior does not provide a general mechanism for exploring alternative tri-state trajectories. Metaheuristic methods provide broader exploration, although their effectiveness depends on the search representation, feasibility handling, and parameter configuration [18–21]. Quantum-inspired optimization offers an alternative classical search mechanism based on probabilistic state representation and adaptive updates. In this work, TQRS is therefore investigated as an alternative search strategy rather than assumed to be superior to established heuristics. Such methods do not require quantum hardware and can run on classical computers while using principles inspired by quantum state evolution. This makes them attractive as lightweight alternatives for practical network optimization.

Motivated by these observations, this work develops a tri-state quantum-inspired sleep scheduling framework for green 6G BSs. Each BS can operate in deep sleep, light sleep, or active mode. The proposed method uses a probability-based, rotation-inspired search mechanism to update scheduling probabilities according to solution quality. The design

jointly considers energy consumption, service delay, wake-up transition cost, and packet-loss constraints. To evaluate the framework, we use a common frozen 24-slot traffic trajectory and a 15-BS network environment. The comparison includes deterministic reference policies, an internal Greedy+Repair heuristic, population-based metaheuristics, and lightweight deep-RL methods. The resulting evaluation is intended as a controlled single-workload comparison rather than a statistical multi-run superiority study.

The main contributions of this paper are summarized as follows:

- We develop a practical tri-state BS power model with deep sleep, light sleep, and active operating modes.
- We formulate a QoS-aware sleep scheduling problem that jointly minimizes energy use while controlling delay and packet loss.
- We propose a quantum rotation based optimization framework for adaptive BS state selection under dynamic traffic demand.
- We incorporate wake-up transition penalties to capture realistic switching overhead between sleep and active modes.
- We validate the framework using a controlled frozen workload and compare it with deterministic, metaheuristic, and deep-RL scheduling methods.

The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 presents the network and energy models. Section 4 formulates the optimization problem. Section 5 describes the proposed scheduler. Section 6 explains the experimental setup. Section 7 discusses the results. Finally, Section 8 concludes the paper.

2 Related Work

Energy efficiency has become a major design objective for future sixth-generation (6G) communication systems. The rapid growth of network densification, intelligent services, and continuous connectivity is expected to increase the power consumption of radio access networks. Therefore, efficient base station (BS) sleep scheduling has received growing attention in recent literature. Existing studies have investigated binary sleep control, multi-state power management, machine learning-based scheduling, and heuristic optimization. More recent research in energy-constrained IoT environments has also emphasized dynamic state adaptation, traffic-aware scheduling, and learning-based power management. This section reviews these research directions and identifies the specific methodological gap addressed by the present work.

2.1 Energy-Efficient Base Station Sleep Scheduling

BS sleep scheduling is an effective mechanism for reducing network energy consumption during periods of low or spatially uneven traffic demand. In this approach, lightly loaded BSs are temporarily transitioned to low-power states while sufficient service capacity is retained through active BSs. Several recent studies have proposed traffic-aware BS switching frameworks for beyond-5G and 6G networks. A two-phase cell switching model

combined demand estimation with adaptive BS control and demonstrated substantial energy reduction under dynamic traffic conditions [22]. Another work introduced a two-time-scale energy-saving strategy using BS deep sleep, symbol shutdown, and power-allocation coordination [23]. The authors reported improved long-term efficiency while preserving quality of service. Flexible BS sleeping has also been investigated in decoupled radio access architectures, where control and data planes are separated. Such designs allow more aggressive BS sleeping without completely losing network accessibility [18].

The relevance of adaptive state management is also evident in recent energy-efficient IoT scheduling studies. Khan et al. proposed Dynamic and Adaptive Scheduling of Cognitive Sensors (DASCS), in which cognitive sensors dynamically adjust their operational states according to traffic conditions and sensing roles. The approach employs a multi-state scheduling mechanism to reduce redundant sensing and communication while improving energy efficiency and network stability [24]. This work demonstrates that state-aware scheduling can provide an effective mechanism for adapting resource availability to changing workload conditions. However, DASCS is developed for cognitive sensors in a wireless mesh IoT environment and focuses on sensor-level sensing and communication rather than cellular BS sleep scheduling.

Similarly, a hybrid scheduling framework combining CADS, EASS, and EDASS was proposed to evaluate dynamic scheduling mechanisms for cognitive IoT sensors. The hybrid approach considers communication overhead, latency, buffering, message transmission, and energy consumption, demonstrating that different scheduling mechanisms exhibit different energy–performance characteristics under varying operating conditions [25]. These results further highlight the importance of jointly considering energy and service-related metrics rather than optimizing energy consumption in isolation.

Although these studies demonstrate the energy-saving potential of adaptive sleep and state control, many approaches still employ binary ON/OFF decisions or application-specific sensor-state models. Such formulations provide computational simplicity but offer limited flexibility when the network must balance energy consumption against service responsiveness during rapidly changing traffic conditions. In particular, an intermediate operating state can provide a compromise between the low power consumption of deep sleep and the faster recovery of an active BS.

2.2 Multi-State Power Management

Modern BS hardware can support multiple operating modes rather than only active and inactive states. Intermediate modes such as standby or light sleep can preserve partial readiness while consuming less energy than full activation. Recent studies in optical and wireless access networks have explored delay-aware sleep synchronization and grouped wake-up policies to balance latency and power savings [26]. Other works on sustainable 6G infrastructure have investigated adaptive activation of communication resources according to traffic demand [27]. These findings indicate that multi-state operation can provide additional flexibility compared with binary control and can reduce the transition burden associated with repeatedly switching a BS between fully active and deep-sleep states.

Recent IoT research provides additional evidence for the usefulness of multi-state power management. Khan et al. developed a reinforcement learning-based dynamic power-management framework for IoT-enabled consumer electronics in which sensor components dynamically transition between active and sleep modes according to network

conditions. The objective was to reduce idle listening, redundant communication, and associated energy consumption while maintaining sensing coverage [28]. The study demonstrates the potential of learning-based state transitions for energy-constrained systems, although its application domain is consumer electronics and cognitive IoT sensors rather than cellular BSs.

More recently, adaptive scheduling of cognitive IoT sensors has also been formulated using reinforcement learning. The ASC-RL framework considers internal sensor states, environmental sensing information, and historical energy and switching information to determine state-changing policies [29]. The approach evaluates energy efficiency together with detection, transmission, false-alarm, and reliability-related measures. Such work demonstrates that state transitions can be conditioned on both current environmental information and historical system behavior rather than being determined by a static threshold.

These studies are closely related to the present problem because they demonstrate the value of dynamic state adaptation rather than purely static power-control rules. However, their system objectives, network architectures, and state-transition mechanisms differ from the cellular BS sleep-scheduling problem considered here. In particular, the present work focuses on a three-state cellular BS representation consisting of deep sleep, light sleep, and active operation, together with explicit feasibility preservation and wake-up transition effects. Thus, the proposed formulation extends the state-adaptation principle toward a constrained cellular BS scheduling problem in which energy, service delay, traffic feasibility, and transition effects must be considered jointly.

2.3 AI and Metaheuristic Based Scheduling

Artificial intelligence has increasingly been adopted for energy-aware network control. Deep-learning-based models have been used for traffic prediction, load balancing, and BS switching decisions in smart radio access networks [30]. Reinforcement learning has also shown promise for adaptive sleep scheduling. A confidence-based node-sleep framework using Q-learning improved network lifetime while preserving sensing coverage in green IoT environments [31]. Metaheuristic methods such as genetic algorithms, particle swarm optimization, and hybrid evolutionary search have also been applied to communication resource optimization [20,32,33]. These approaches can provide stronger search capability than static threshold rules, but they may require repeated iterations, parameter tuning, or training procedures.

Recent reinforcement-learning-based scheduling studies further demonstrate the ability of learning mechanisms to adapt operational states to changing network conditions. The CIoT-DPM framework uses a reward-driven reinforcement learning mechanism to dynamically change sensor states while reducing redundant traffic, idle listening, and overhearing [28]. Likewise, ASC-RL uses state, environmental, and historical parameters to derive an adaptive state-changing policy for cognitive IoT sensors [29]. The DASCs framework also demonstrates that dynamic state adaptation can improve energy utilization and network stability in resource-constrained IoT environments [24]. Collectively, these studies establish the importance of adaptive decision-making for energy-aware network operation.

However, learning-based approaches generally require a training process, environment interaction, reward design, or sufficiently representative training data. For example, ASC-RL employs a PPO-based implementation and evaluates the resulting policy over

multiple training episodes [29]. Such requirements may introduce additional computational and implementation complexity when the scheduling problem is formulated as a large constrained combinatorial optimization problem.

Reinforcement learning and deep-learning methods are particularly attractive for dynamic network control because they can learn state-dependent policies. However, their practical deployment may require substantial training data, repeated interaction with the environment, or additional computational resources. In contrast, the present work investigates a lightweight probability-based search mechanism that can be executed directly on a classical computing platform without requiring a training phase.

Beyond energy and performance optimization, security and trust are also becoming important considerations in highly connected 6G and IoT infrastructures. Comprehensive studies of authentication and attack-detection mechanisms in vehicular ad-hoc networks highlight the increasing importance of protecting connected communication environments against security threats [34]. Recent work on dynamic security computing has further investigated zero-trust-oriented security control and privacy-domain prevention mechanisms for dynamic communication environments [35]. These studies are complementary to the present work: while security-aware resource management is important for future operational networks, the present study deliberately focuses on the energy–delay and service-feasibility aspects of BS sleep scheduling.

2.4 Quantum-Inspired Optimization for Green Networks

Quantum-inspired optimization methods have attracted attention as classical optimization techniques that borrow selected concepts from quantum information processing, such as probabilistic state representation and population-level search diversity. Importantly, quantum-inspired algorithms do not require quantum hardware. They can therefore be implemented using conventional numerical computation while providing an alternative mechanism for exploring large combinatorial search spaces.

Recent network-optimization studies have investigated quantum-inspired, hybrid quantum-learning, and quantum-assisted search concepts for wireless resource-management problems [30, 33]. These studies suggest that probabilistic search representations can provide an alternative to conventional deterministic heuristics. However, the use of such classical quantum-inspired mechanisms for tri-state cellular BS sleep scheduling remains comparatively limited.

The recent IoT scheduling literature also reinforces the importance of probabilistic and adaptive state-selection mechanisms. Dynamic scheduling frameworks have demonstrated that operational states can be modified in response to traffic, sensing conditions, energy levels, and historical system behavior [24, 29]. However, these approaches primarily employ rule-based or reinforcement-learning-based state-selection mechanisms for cognitive sensors. They do not investigate a quantum-inspired population-based search mechanism for constrained three-state cellular BS scheduling.

In this paper, the term “quantum-inspired” refers specifically to the classical probabilistic representation and rotation-inspired probability update used by the scheduler. The proposed method does not require qubits, quantum hardware, physical quantum evolution, or execution of quantum gates. This distinction is important because the objective of the framework is to investigate whether a quantum-inspired classical search mechanism can provide an effective scheduling strategy for the considered 6G BS management problem.

2.5 Comparative Summary of Existing Studies

To improve readability, the comparison is divided into two compact tables. Table 1 summarizes the principal functional capabilities of representative approaches.

Table 1: Functional Comparison of Existing Studies and Proposed TQRS

Reference	Sleep Model	Multi-State	Delay Aware	Loss Aware	Wake-Up Cost	Dynamic Traffic
Salamatmoghadasi et al. [22]	Binary	No	No	No	No	Partial
Suresh et al. [30]	Binary	No	Partial	No	No	Yes
Allawi et al. [26]	Binary	No	Yes	No	Yes	Yes
Chellappan et al. [10]	Binary	No	Yes	No	Partial	Yes
Sun et al. [18]	Binary	No	Yes	No	Partial	Yes
Fayad et al. [32]	Multi-State	Yes	Yes	Partial	Yes	Yes
Singh et al. [33]	Binary	No	Partial	No	No	Yes
Taneja et al. [27]	Multi-State	Yes	Yes	No	Partial	Yes
Tang et al. [31]	Binary	No	Yes	No	Yes	Yes
Zhu et al. [23]	Hybrid	Partial	Yes	Partial	Yes	Yes
Yang et al. [36]	Binary	No	Partial	No	Yes	Yes
Rajule et al. [20]	Binary	No	No	No	No	Partial
Proposed TQRS	Tri-State	Yes	Yes	Yes	Yes	Yes

2.6 Research Gap and Motivation

The reviewed literature demonstrates that energy-aware BS scheduling has progressed from simple binary switching toward adaptive, multi-state, learning-based, and heuristic approaches. Nevertheless, several technical gaps remain relevant to the scheduling problem considered in this study.

- Many existing BS sleep-scheduling methods primarily use binary ON/OFF operation and therefore provide limited flexibility for balancing energy savings against service responsiveness.
- Existing multi-state approaches demonstrate the value of intermediate operating modes, but explicit integration of deep sleep, light sleep, and active states with a unified scheduling mechanism and feasibility-preserving state selection remains insufficiently explored.
- Energy optimization alone is insufficient for practical BS management. A scheduling decision must also preserve service capacity and control delay, unserved traffic, and the effects associated with transitions between sleep and active states.
- Reinforcement-learning and deep-learning approaches can provide adaptive decision making, but their training requirements and computational overhead can make lightweight deployment challenging, particularly when rapid scheduling decisions are required.

- Evolutionary and metaheuristic approaches provide broader search capability than simple rules, but their effectiveness depends on the representation of the scheduling state, feasibility handling, and search configuration.
- Quantum-inspired optimization provides an alternative classical search paradigm, but its application to tri-state cellular BS sleep scheduling with explicit service constraints and transition effects remains comparatively limited.

Motivated by these gaps, this paper investigates a *Tri-State Quantum Rotation Scheduler (TQRS)* that represents each BS using deep sleep, light sleep, and active states and searches the resulting combinatorial scheduling space using a classical probability-based, rotation-inspired update mechanism. [The framework explicitly incorporates service feasibility and wake-up transition effects during candidate evaluation.](#) The objective is not to claim universal superiority over all existing scheduling methods, but to determine whether the proposed lightweight quantum-inspired search mechanism can provide a feasible and temporally adaptive tri-state scheduling solution under the considered dynamic traffic environment. The resulting performance is interpreted comparatively, including cases in which conventional heuristics or metaheuristics may achieve lower objective values on the evaluated workload.

Notation Table

For clarity, the principal mathematical symbols used throughout the paper are summarized in Table 2.

Table 2: Summary of Notations

Symbol	Description
N	Total number of base stations (BSs) in the network
T	Total number of scheduling time slots
$\mathcal{B}$	Set of base stations, $\{1, 2, \dots, N\}$
$\mathcal{T}$	Set of time slots, $\{1, 2, \dots, T\}$
i	Index of a base station, $i \in \mathcal{B}$
t	Index of a time slot, $t \in \mathcal{T}$
C	Service capacity of one active BS
$D(t)$	Aggregate traffic demand at time slot t
$x_i(t)$	Operating state of BS i at time slot t
$\mathbf{x}(t)$	Scheduling vector of all BS states at time slot t
$0, 1, 2$	Deep sleep, light sleep, and active state, respectively
$n_d(t)$	Number of BSs in deep sleep mode at slot t
$n_l(t)$	Number of BSs in light sleep mode at slot t
$n_a(t)$	Number of active BSs at slot t
$\Gamma(t)$	Total active network capacity at slot t
P_d	Power consumption of deep sleep state
P_l	Power consumption of light sleep state
P_a	Power consumption of active state
$E_s(t)$	Static operating energy at slot t

Continued on next page

Table 2 – continued from previous page

Symbol	Description
$E_w(t)$	Wake-up transition energy at slot t
$E(t)$	Total energy consumption at slot t
Δ_{da}	Transition cost from deep sleep to active state
Δ_{la}	Transition cost from light sleep to active state
$\rho(t)$	Traffic utilization ratio at slot t
$\tau(t)$	Average service delay at slot t
$L(t)$	Packet loss at slot t
w_E, w_D, w_L, w_W	Objective weights for energy, delay, loss, and wake-up latency
$J(t)$	Multi-objective cost function at slot t
$J_e(t)$	Penalty-augmented fitness function
$\Pi(t)$	Constraint penalty term
λ_C, λ_L	Capacity and loss penalty coefficients
Ω	Search space of feasible tri-state schedules
$ \Omega $	Total number of candidate schedules
M	Population size of TQRS
K	Maximum number of optimization iterations
$\mathbf{p}_i$	Probability vector of BS i over three states
$p_{i,j}$	Probability of assigning BS i to state j
θ	Probability rotation step size
ϵ	Minimum probability floor
$\mathbf{x}^*$	Best candidate solution found by TQRS
F_m	Fitness of candidate solution m
$\bar{E}$	Mean energy consumption over all slots
$\bar{\tau}$	Mean delay over all slots
$\bar{L}$	Mean packet loss over all slots
η_q	Relative energy saving of method q
$w_{da}(t)$	Number of deep-sleep-to-active transitions at slot t
$w_{la}(t)$	Number of light-sleep-to-active transitions at slot t
$\tau_{\text{svc}}(t)$	Service-delay component at slot t
$\tau_{\text{wake}}(t)$	Wake-up transition latency at slot t
$\tau(t)$	Total service delay at slot t
$L_{\text{Mbps}}(t)$	Unserved/overflow traffic at slot t
τ_{DA}	Deep-sleep-to-active wake-up latency
τ_{LA}	Light-sleep-to-active wake-up latency

3 System Model

This section defines the network, traffic, power, transition, and service models used for tri-state base station (BS) sleep scheduling. A time-slotted 6G radio access network is considered in which a central controller determines the operating state of each BS at every scheduling slot. The formulation explicitly represents deep sleep, light sleep, and active operation and incorporates the energy associated with state transitions. The model is evaluated using a common 24-slot traffic realization and a 15-BS network configuration.

3.1 Network Architecture

We consider a cellular network composed of N homogeneous base stations (BSs). The scheduling horizon is divided into T discrete time slots. During each slot, a central controller observes the aggregate traffic demand and determines the operating state of every BS. Each BS can operate in deep sleep, light sleep, or active mode. The present study intentionally uses homogeneous BS configurations so that the contribution of the proposed tri-state scheduling mechanism can be isolated from hardware-level heterogeneity. Heterogeneous BS capabilities are discussed as an extension for future investigation.

Let the set of BSs be denoted by

$$\mathcal{B} = \{1, 2, \dots, N\}, \quad (1)$$

and the set of scheduling time slots by

$$\mathcal{T} = \{1, 2, \dots, T\}. \quad (2)$$

For the evaluated scenario,

$$N = 15, \quad T = 24. \quad (3)$$

Each active BS provides a service capacity of 120 Mbps. Therefore, if $n_a(t)$ BSs are active during slot t , the aggregate available service capacity is

$$C = 120 \text{ Mbps}, \quad (4)$$

$$\Gamma(t) = n_a(t)C. \quad (5)$$

The capacity expression is used consistently throughout the feasibility evaluation: a scheduling state is service-feasible only when the available active capacity satisfies the traffic requirement under the adopted service model.

3.2 Traffic Demand Model

Let $D(t)$ denote the aggregate traffic demand during scheduling slot t . The experimental evaluation uses a frozen 24-slot traffic realization rather than independently resampling traffic for each optimization run. This ensures that all compared methods face exactly the same workload and that differences in the reported results arise from the scheduling strategies rather than from different traffic realizations.

The frozen traffic sequence contains 24 hourly-equivalent scheduling slots with a minimum demand of 269 Mbps, a maximum demand of 1250 Mbps, and a cumulative traffic demand of 17,395 Mbps. The sequence represents time-varying low-, medium-, and high-demand operating conditions over the evaluation horizon.

$$D(t) \in \mathbb{R}_{\geq 0}, \quad t \in \mathcal{T}, \quad (6)$$

with the experimentally validated bounds

$$269 \text{ Mbps} \leq D(t) \leq 1250 \text{ Mbps}, \quad \sum_{t=1}^{24} D(t) = 17395 \text{ Mbps} \cdot \text{slot}. \quad (7)$$

The same frozen traffic sequence is supplied to every evaluated method. Consequently, the baseline comparison is controlled with respect to the workload and does not involve method-dependent traffic generation.

3.3 Tri-State Base Station Power Model

Each BS operates in one of three discrete states:

- Deep Sleep (0),
- Light Sleep (1),
- Active (2).

Let $x_i(t) \in \{0, 1, 2\}$ denote the state of BS i at slot t .

$$n_d(t) = \sum_{i=1}^N \mathbb{I}\{x_i(t) = 0\}, \quad (8)$$

$$n_l(t) = \sum_{i=1}^N \mathbb{I}\{x_i(t) = 1\}, \quad (9)$$

$$n_a(t) = \sum_{i=1}^N \mathbb{I}\{x_i(t) = 2\}. \quad (10)$$

The corresponding steady-state power levels used in the experimental environment are

$$P_d = 8 \text{ W}, \quad P_l = 35 \text{ W}, \quad P_a = 100 \text{ W}. \quad (11)$$

The static operating energy during slot t is therefore

$$E_s(t) = n_d(t)P_d + n_l(t)P_l + n_a(t)P_a. \quad (12)$$

3.4 Transition Cost Model

A transition from a sleep state to the active state incurs an additional transition cost. The model distinguishes deep-sleep and light-sleep transitions because the two states represent different levels of power reduction and recovery overhead.

$$\Delta_{da} = 28 \text{ Wh}, \quad (13)$$

$$\Delta_{la} = 8 \text{ Wh}. \quad (14)$$

Let $x_i(t-1)$ denote the previous state of BS i . The transition energy is

$$E_w(t) = \sum_{i=1}^N [\Delta_{da} \mathbb{I}\{x_i(t-1) = 0, x_i(t) = 2\} + \Delta_{la} \mathbb{I}\{x_i(t-1) = 1, x_i(t) = 2\}]. \quad (15)$$

The total energy consumption during slot t is

$$E(t) = E_s(t) + E_w(t). \quad (16)$$

The transition model accounts for the energy penalty of waking a sleeping BS. Transition latency is treated separately in the service-delay formulation rather than being represented as an energy term.

3.5 Service Capacity and Delay Model

The service-delay model is formulated using the available active capacity, traffic utilization, and the temporal cost associated with waking a sleeping BS. In contrast to the previous formulation, the number of light-sleep BSs does not directly reduce service delay. Sleep states can reduce immediately available capacity, while transitions from sleep to active operation introduce additional wake-up latency.

The aggregate active service capacity at slot t is

$$C_a(t) = n_a(t)C, \quad (17)$$

where C denotes the service capacity of one active BS.

For a feasible scheduling state,

$$C_a(t) \geq D(t), \quad (18)$$

The corresponding utilization ratio is defined in Section 3.6.

3.6 Service Delay and Packet Loss Model

Service delay is determined by the offered traffic, the available active capacity, and the latency associated with sleep-to-active transitions. In the corrected formulation, the number of light-sleep BSs does not directly produce a delay reduction. Instead, sleep states affect service through the available active capacity, while required wake-up transitions introduce an additional positive latency.

The traffic-utilization ratio is defined as

$$\rho(t) = \frac{D(t)}{\Gamma(t)}, \quad \Gamma(t) > 0, \quad (19)$$

where $D(t)$ is the aggregate traffic demand and $\Gamma(t)$ is the available active service capacity.

For a feasible state satisfying $\Gamma(t) \geq D(t)$, the service delay is modeled as a nondecreasing function of utilization:

$$\tau_{\text{svc}}(t) = 2.7 + 4.8\rho(t) + 6 \max(0, \rho(t) - 0.85). \quad (20)$$

The additional term becomes active only when utilization exceeds 0.85, thereby representing the increased service delay associated with operation near the available capacity limit. A minimum service-delay bound is imposed as

$$\tau_{\text{svc}}(t) = \max(2.0, 2.7 + 4.8\rho(t) + 6 \max(0, \rho(t) - 0.85)). \quad (21)$$

Sleep-to-active transitions are modeled explicitly. Let $w_{da}(t)$ and $w_{la}(t)$ denote the numbers of BSs transitioning from deep sleep and light sleep to the active state, respectively:

$$w_{da}(t) = \sum_{i=1}^N \mathbb{I}\{x_i(t-1) = 0, x_i(t) = 2\}, \quad (22)$$

$$w_{la}(t) = \sum_{i=1}^N \mathbb{I}\{x_i(t-1) = 1, x_i(t) = 2\}. \quad (23)$$

Because multiple BSs can be awakened concurrently, the effective wake-up latency of a scheduling slot is determined by the slowest transition that is required:

$$\tau_{\text{wake}}(t) = \max[8 \mathbb{I}\{w_{da}(t) > 0\}, 2 \mathbb{I}\{w_{la}(t) > 0\}, 0] \text{ ms}. \quad (24)$$

Thus, a deep-sleep-to-active transition contributes 8 ms, whereas a light-sleep-to-active transition contributes 2 ms. If both types of transition occur simultaneously, the required wake-up latency is 8 ms because the transitions proceed in parallel and the slowest required transition determines the effective delay.

The total service delay is consequently

$$\tau(t) = \tau_{\text{svc}}(t) + \tau_{\text{wake}}(t). \quad (25)$$

When the available active capacity is insufficient to satisfy the offered traffic, the overflow traffic is

$$L_{\text{Mbps}}(t) = \max(0, D(t) - \Gamma(t)). \quad (26)$$

For an overloaded state, the service-delay component is modeled as

$$\tau_{\text{svc}}(t) = \max\left(5.0, 10.0 + \frac{L_{\text{Mbps}}(t)}{18.0}\right). \quad (27)$$

The corresponding packet-loss ratio is

$$L(t) = \frac{L_{\text{Mbps}}(t)}{D(t)}, \quad D(t) > 0. \quad (28)$$

Therefore, $L(t) = 0$ whenever the active capacity is sufficient, while positive values indicate that the selected BS-state configuration cannot serve the complete offered traffic. Feasibility is consequently evaluated using the active-capacity constraint rather than through an artificial negative delay contribution from light-sleep BSs.

3.7 Operational Assumptions

The following assumptions define the scope of the system model and ensure consistent evaluation across all scheduling methods:

- All BSs are homogeneous in the primary evaluation, with a common service capacity of 120 Mbps and common operating-state power levels.

- Traffic demand is represented at the aggregate network level for the primary scheduling experiment.
- The scheduling decision is made at the beginning of each time slot using the traffic demand observed for that slot.
- A sleep-to-active transition incurs both an energy cost and a temporal wake-up latency.
- Simultaneous wake-up transitions are assumed to occur in parallel; hence, the effective transition latency is determined by the slowest required wake-up transition.
- Packet loss is undesirable and is explicitly incorporated into the feasibility and penalty evaluation.
- Inter-cell interference, user mobility, and user-specific QoS differentiation are outside the scope of the primary network-level abstraction and are identified as extensions for future work.

The resulting model provides a consistent connection between BS-state selection, available service capacity, transition energy, wake-up latency, service delay, and packet loss. These quantities are subsequently used in the optimization formulation presented in Section 4.

4 Problem Formulation

Based on the system model presented in Section 3, the scheduling problem is formulated as a constrained tri-state optimization problem. At each scheduling slot, the controller determines the operating state of every BS from the set {Deep Sleep, Light Sleep, Active}. The objective is to reduce energy consumption while maintaining low service delay, avoiding unserved traffic, and limiting wake-up transition latency.

4.1 Decision Variables

Let

$$x_i(t) \in \{0, 1, 2\}, \quad i \in \mathcal{B}, \quad t \in \mathcal{T}, \quad (29)$$

where

$$x_i(t) = \begin{cases} 0, & \text{Deep Sleep,} \\ 1, & \text{Light Sleep,} \\ 2, & \text{Active.} \end{cases} \quad (30)$$

The scheduling vector at slot t is

$$\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_N(t)]. \quad (31)$$

The complete scheduling trajectory over the evaluation horizon is therefore

$$\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(T)]. \quad (32)$$

4.2 Active-Capacity Requirement

For a traffic demand $D(t)$, the minimum number of active BSs required to serve the offered traffic is

$$n_{\text{req}}(t) = \left\lceil \frac{D(t)}{C} \right\rceil. \quad (33)$$

The active-BS count must satisfy

$$n_a(t) \geq n_{\text{req}}(t). \quad (34)$$

Equivalently, because

$$\Gamma(t) = n_a(t)C, \quad (35)$$

the service-feasibility condition is

$$\Gamma(t) \geq D(t). \quad (36)$$

This formulation avoids imposing an arbitrary additional active BS during high-demand periods. Robustness is instead handled directly through the capacity, delay, loss, and penalty terms of the corrected evaluation model.

4.3 Energy Objective

The total energy consumed during slot t is

$$E(t) = E_s(t) + E_w(t), \quad (37)$$

where $E_s(t)$ is the static operating energy and $E_w(t)$ is the sleep-to-active transition energy defined in Section 3.3.

The energy component is normalized using the implementation-defined reference value

$$E_{\text{norm}} = 1500 \text{ Wh}. \quad (38)$$

Thus, the normalized energy term is

$$J_E(t) = \frac{E(t)}{E_{\text{norm}}}. \quad (39)$$

4.4 Delay Objective

The total delay is composed of service delay and wake-up latency:

$$\tau(t) = \tau_{\text{svc}}(t) + \tau_{\text{wake}}(t). \quad (40)$$

The normalized delay component is

$$J_D(t) = \frac{\tau(t)}{\tau_{\text{norm}}}, \quad (41)$$

where

$$\tau_{\text{norm}} = 20 \text{ ms}. \quad (42)$$

4.5 Packet-Loss Objective

The unserved traffic during slot t is

$$L_{\text{Mbps}}(t) = \max(0, D(t) - \Gamma(t)). \quad (43)$$

The corresponding packet-loss ratio is

$$J_L(t) = \frac{L_{\text{Mbps}}(t)}{D(t)}, \quad D(t) > 0. \quad (44)$$

A zero-loss solution is preferred because it simultaneously satisfies the service-capacity requirement and avoids the loss penalty applied during fitness evaluation.

4.6 Wake-Up Latency Objective

The wake-up latency is explicitly incorporated as a separate optimization component:

$$J_W(t) = \frac{\tau_{\text{wake}}(t)}{\tau_{\text{wake,norm}}}, \quad (45)$$

where

$$\tau_{\text{wake,norm}} = 8 \text{ ms}. \quad (46)$$

The normalization corresponds to the maximum modeled wake-up latency, namely the deep-sleep-to-active transition latency.

4.7 Demand-Aware Multi-Objective Cost

The relative importance of energy, service delay, packet loss, and wake-up latency is adapted according to the current traffic regime. Let (w_E, w_D, w_L, w_W) denote the four objective weights. The slot-wise normalized cost is

$$J(t) = w_E J_E(t) + w_D J_D(t) + w_L J_L(t) + w_W J_W(t). \quad (47)$$

The adopted demand-aware weight policy is

$$(w_E, w_D, w_L, w_W) = \begin{cases} (0.60, 0.30, 0.05, 0.05), & D(t) \leq 750, \\ (0.50, 0.40, 0.05, 0.05), & 750 < D(t) \leq 1100, \\ (0.35, 0.55, 0.05, 0.05), & D(t) > 1100. \end{cases} \quad (48)$$

Hence, energy receives the highest weight under low traffic, whereas delay receives progressively greater importance as the offered traffic approaches the capacity limit. Packet loss and wake-up latency retain explicit but smaller weights in all traffic regimes.

4.8 Feasibility Constraints

The optimization is subject to the following constraints.

4.8.1 State Constraint

$$x_i(t) \in \{0, 1, 2\}, \quad \forall i \in \mathcal{B}, t \in \mathcal{T}. \quad (49)$$

4.8.2 Capacity Constraint

$$n_a(t)C \geq D(t). \quad (50)$$

4.8.3 Non-Negative Loss Constraint

$$L_{\text{Mbps}}(t) = \max(0, D(t) - n_a(t)C). \quad (51)$$

For the preferred feasible solutions,

$$L_{\text{Mbps}}(t) = 0, \quad \forall t \in \mathcal{T}. \quad (52)$$

4.9 Penalty-Augmented Fitness

Because candidate schedules generated during the probabilistic search may temporarily violate the capacity requirement, infeasible candidates are penalized during fitness evaluation. The penalty-augmented fitness is defined as

$$J_e(t) = J(t) + \Pi(t), \quad (53)$$

where the capacity and loss penalties are

$$\Pi(t) = \lambda_C \mathbb{I}\{n_a(t) < n_{\text{req}}(t)\} + \lambda_L \frac{L_{\text{Mbps}}(t)}{D(t)}. \quad (54)$$

The implementation uses

$$\lambda_C = 1000, \quad \lambda_L = 1000. \quad (55)$$

The capacity penalty strongly discourages schedules with insufficient active capacity, while the loss penalty increases continuously with the fraction of unserved traffic. Consequently, feasible zero-loss schedules dominate infeasible schedules during candidate selection.

4.10 Optimization Problem

The scheduling problem can therefore be expressed as

$$\boxed{\min_{\mathbf{x}} \sum_{t=1}^T J_e(t)} \quad (56)$$

subject to

$$x_i(t) \in \{0, 1, 2\}, \quad \forall i, t, \quad (57)$$

$$n_a(t)C \geq D(t), \quad \forall t \quad \text{for feasible solutions}, \quad (58)$$

$$L_{\text{Mbps}}(t) \geq 0, \quad \forall t. \quad (59)$$

The optimization therefore searches for a tri-state scheduling trajectory that minimizes energy while simultaneously controlling service delay, unserved traffic, and wake-up latency. The proposed TQRS algorithm solves this combinatorial problem through probabilistic candidate generation and rotation-based search, as described in Section 5.

4.11 Combinatorial Search Space

Each BS can assume one of three operating states. Therefore, for a single time slot, the number of possible scheduling configurations is

$$|\Omega| = 3^N. \quad (60)$$

For the considered $N = 15$ BS network,

$$|\Omega| = 3^{15} = 14,348,907. \quad (61)$$

The exponential growth of the search space makes exhaustive enumeration increasingly expensive. This motivates the probabilistic search mechanism developed in the next section.

5 Proposed Tri-State Quantum Rotation Scheduler

The proposed Tri-State Quantum Rotation Scheduler (TQRS) is a classical quantum-inspired metaheuristic developed for constrained tri-state base station sleep scheduling. The method does not execute quantum gates, require qubits, or depend on quantum hardware. Instead, it adopts a probabilistic state representation and a rotation-inspired probability update mechanism to explore the combinatorial scheduling space defined in Section 4.

The complete TQRS procedure consists of five principal stages: (i) probabilistic initialization, (ii) categorical candidate generation, (iii) feasibility-aware fitness evaluation, (iv) probability rotation toward the best observed state configuration, and (v) local neighborhood refinement. The resulting procedure provides a lightweight classical search mechanism for tri-state BS scheduling.

5.1 Framework Overview

At each scheduling horizon, TQRS maintains a probability distribution over the three operating states of every BS. Candidate scheduling trajectories are generated by independently sampling the state of each BS from its corresponding categorical probability distribution. The generated candidates are evaluated using the corrected multi-objective fitness function of Section 4. Feasible candidates are preferred over infeasible candidates, while energy, delay, loss, and wake-up latency determine the quality among feasible solutions.

The overall search process can be represented as

$$\mathcal{P}^{(0)} \rightarrow \mathcal{X}^{(1)} \rightarrow \mathcal{F}^{(1)} \rightarrow \mathcal{X}^{*(1)} \rightarrow \mathcal{P}^{(1)} \rightarrow \dots \rightarrow \mathcal{X}^*, \quad (62)$$

where $\mathcal{P}$ denotes the probability representation, $\mathcal{X}$ denotes the generated candidate set, $\mathcal{F}$ denotes the corresponding fitness values, and $\mathcal{X}^*$ denotes the best candidate (See Figure 1).

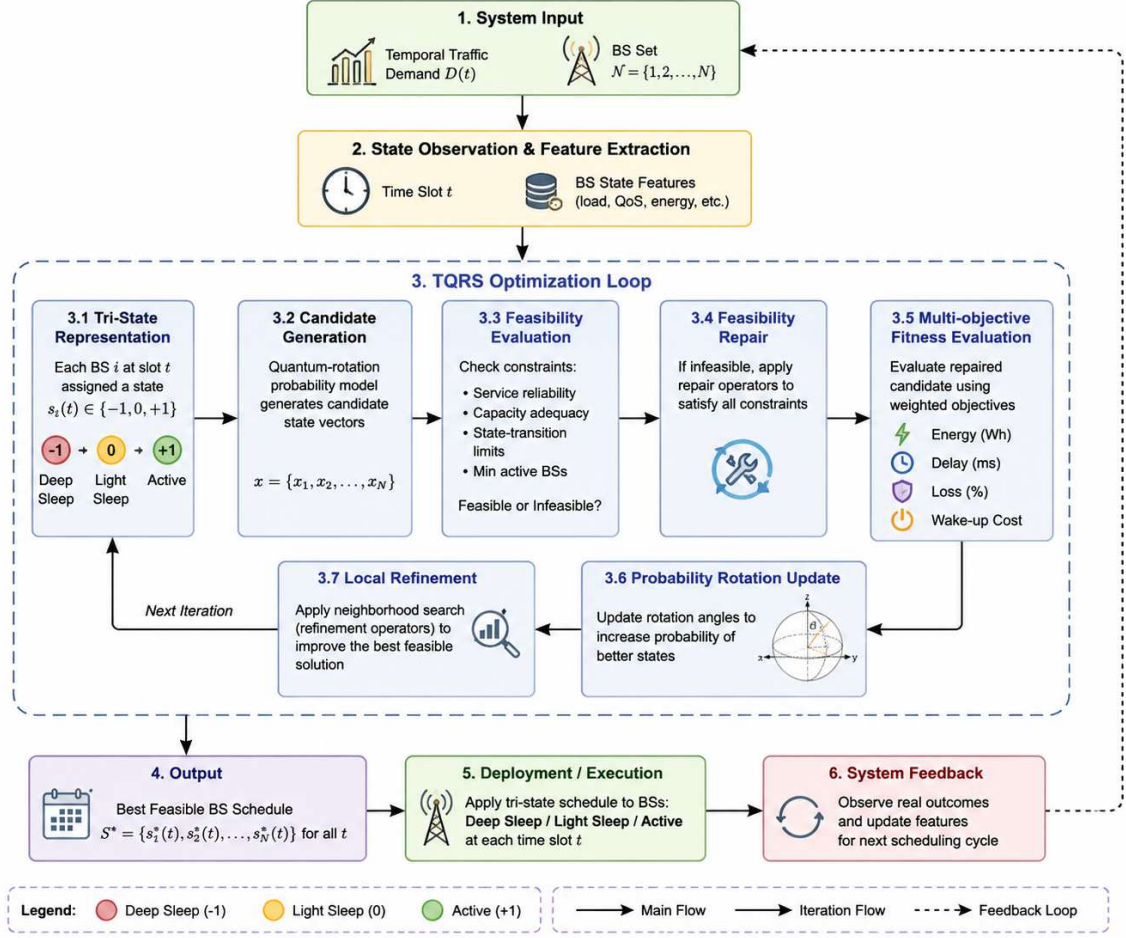


Figure 1: Overview of the proposed Tri-State Quantum Rotation Scheduler (TQRS) for feasibility-aware and energy-efficient base-station sleep scheduling.

5.2 Tri-State Probabilistic Representation

Each BS can assume one of three operating states:

$$\mathcal{S} = \{0, 1, 2\} = \{\text{Deep}, \text{Light}, \text{Active}\}. \quad (63)$$

Instead of assigning a deterministic state before the search begins, TQRS maintains a probability vector for each BS. For BS i , the probability vector is

$$\mathbf{p}_i = [p_{i,0}, p_{i,1}, p_{i,2}], \quad (64)$$

subject to

$$p_{i,0} + p_{i,1} + p_{i,2} = 1, \quad p_{i,j} \geq 0. \quad (65)$$

The three components represent the probabilities of deep sleep, light sleep, and active operation, respectively. The probability representation is a classical numerical representation and should not be interpreted as a physical quantum state.

5.3 Probability Initialization

At initialization, all three states are assigned equal probability:

$$p_{i,0}^{(0)} = p_{i,1}^{(0)} = p_{i,2}^{(0)} = \frac{1}{3}. \quad (66)$$

Uniform initialization avoids imposing a deterministic preference for any operating mode and provides balanced exploration at the beginning of the search.

5.4 Categorical Candidate Generation

For a given probability vector $\mathbf{p}_i$, the state of BS i is sampled according to

$$x_i \sim \text{Categorical}(p_{i,0}, p_{i,1}, p_{i,2}). \quad (67)$$

A candidate scheduling vector is therefore

$$\mathbf{x} = [x_1, x_2, \dots, x_N]. \quad (68)$$

For the complete 24-slot scheduling horizon, the candidate solution is represented as a trajectory

$$\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(T)], \quad (69)$$

where each row contains the tri-state configuration of all N BSs for one scheduling slot.

5.5 Feasibility-Aware Candidate Evaluation

Every generated candidate is evaluated using the complete system and objective model defined in Sections 3 and 4. The evaluation includes operating energy, transition energy, service delay, wake-up latency, unserved traffic, and feasibility.

Let F_m denote the fitness of candidate m :

$$F_m = \sum_{t=1}^T J_e^{(m)}(t). \quad (70)$$

Feasibility is treated as a priority criterion. A feasible candidate with zero unserved traffic is preferred to an infeasible candidate. Among feasible candidates, lower fitness is preferred.

The best candidate at iteration k is therefore

$$\mathbf{X}^{*(k)} = \arg \min_{\mathbf{x}_m \in \mathcal{X}^{(k)}} F_m, \quad (71)$$

subject to the feasibility-first selection rule.

5.6 Rotation-Inspired Probability Update

After candidate evaluation, the probability distribution is updated toward the state configuration of the current best candidate. This operation is called a rotation-inspired update because it shifts probability mass toward the preferred state, analogous in spirit to the directional update used in quantum-inspired evolutionary representations. It is implemented entirely using classical arithmetic on probability values.

Let s_i denote the state sampled for BS i in the current candidate and b_i denote the corresponding state in the best candidate. When

$$s_i \neq b_i, \quad (72)$$

probability mass is transferred from s_i toward b_i :

$$p_{i,s_i}^{\text{new}} = \max(\epsilon, p_{i,s_i} - \theta), \quad (73)$$

and

$$p_{i,b_i}^{\text{new}} = \min(1 - \epsilon, p_{i,b_i} + \theta). \quad (74)$$

The remaining probability component is retained before normalization. The probability vector is then normalized as

$$p_{i,j}^{\text{new}} = \frac{p_{i,j}^{\text{new}}}{\sum_{r=0}^2 p_{i,r}^{\text{new}}}, \quad j \in \{0, 1, 2\}. \quad (75)$$

The implemented rotation parameters are

$$\theta = 0.04, \quad \epsilon = 0.05. \quad (76)$$

The probability floor prevents premature elimination of a state from the search distribution, thereby retaining a degree of exploration throughout the optimization process.

5.7 Demand-Aware Search Behavior

The search behavior is indirectly adapted to traffic conditions through the demand-dependent objective weights defined in Eq. (48). Under low traffic, energy reduction receives greater importance, encouraging the search toward sleep states. As traffic increases, delay receives greater weight, encouraging configurations with greater active capacity and lower transition-related service penalties.

Therefore, the proposed method does not impose a manually fixed operating state for each traffic regime. Instead, traffic-dependent objective weights modify the fitness landscape, and the probability update subsequently guides the search distribution toward configurations that perform well under the current demand condition.

5.8 Feasibility Repair

Because probabilistic sampling can temporarily produce candidates with insufficient active capacity, a feasibility-repair mechanism is applied during candidate evaluation. If

$$n_a(t)C < D(t), \quad (77)$$

the candidate is repaired by promoting appropriate sleeping BS states toward the active state until the capacity requirement is satisfied or no further promotion is possible. The repaired trajectory is subsequently re-evaluated using the same fitness function.

The repair mechanism is therefore not a separate optimization objective. It is a feasibility-preservation operation that prevents invalid schedules from being treated as competitive solutions merely because their nominal energy consumption is low.

5.9 Local Search Refinement

After the probabilistic search terminates, the best trajectory is subjected to a local refinement stage. For each BS and scheduling position, alternative tri-state assignments are examined while the remaining assignments are kept fixed. A trial replacement is accepted only when its feasibility-aware fitness is better than the current solution.

For a current best trajectory $\mathbf{X}^*$, replacing the state of BS i at slot t by $s \in \mathcal{S}$ gives the trial trajectory

$$\tilde{\mathbf{X}}_{i,t}(s) = \mathbf{X}^* \text{ with } x_i(t) \leftarrow s. \quad (78)$$

The replacement is accepted when

$$F(\tilde{\mathbf{X}}_{i,t}(s)) < F(\mathbf{X}^*) \quad (79)$$

and the candidate satisfies the feasibility requirements.

The local refinement therefore acts as an exploitation stage following the global probabilistic search. It does not alter the probability model and does not introduce an additional learning phase.

5.10 Complete TQRS Procedure

The complete procedure is summarized in Algorithm 1.

Algorithm 1 Tri-State Quantum Rotation Scheduler (TQRS)

Require: Traffic trajectory $D(1 : T)$, previous BS-state trajectory, population size M , maximum iterations K , rotation step θ , probability floor ϵ

Ensure: Best feasible BS-state trajectory $\mathbf{X}^*$

1: Initialize all probability vectors:

$$p_{i,0} = p_{i,1} = p_{i,2} = \frac{1}{3}$$

2: **for** $k = 1, \dots, K$ **do**

3: **for** $m = 1, \dots, M$ **do**

4: Generate a candidate trajectory by categorical sampling.

5: Evaluate the candidate using energy, delay, loss, wake-up latency, and feasibility.

6: Apply feasibility repair when required.

7: **end for**

8: Select the best candidate using feasibility-first ranking.

9: **for** each BS i and scheduling position t **do**

10: Compare the sampled state with the corresponding best state.

11: Transfer probability mass by the rotation-inspired update.

12: Apply probability floor ϵ .

13: Normalize the three-state probability vector.

14: **end for**

15: **end for**

16: Apply local tri-state neighborhood refinement to the best trajectory.

17: Re-evaluate the refined trajectory.

18: **return** $\mathbf{X}^*$

5.11 Computational Complexity

Let M denote the population size, K the number of optimization iterations, N the number of BSs, and T the number of scheduling slots. Candidate generation and probability updates require processing the BS-state representation, while candidate evaluation processes the complete trajectory.

The dominant candidate-evaluation complexity is

$$O(MKNT). \tag{80}$$

The probability representation requires three probabilities for each BS-state decision. Hence, the corresponding memory requirement is

$$O(MNT). \tag{81}$$

For the considered $N = 15$ and $T = 24$ configuration, the trajectory representation remains compact, allowing the proposed method to be executed on a conventional CPU. The complexity analysis also makes explicit that increasing the scheduling horizon increases computational cost linearly for a fixed population and iteration count.

5.12 Interpretation of the Quantum-Inspired Mechanism

The term “quantum-inspired” in TQRS refers to the use of a probabilistic state representation and a rotation-inspired probability update strategy. The method is entirely classical: probability vectors are stored as ordinary numerical variables, candidate states are sampled using a categorical distribution, and all updates are performed through classical arithmetic. No quantum circuit, quantum gate, qubit, quantum annealing hardware, or physical quantum evolution is involved.

This distinction is important for reproducibility and interpretation. The contribution of TQRS is therefore not a claim of quantum computational advantage, but the investigation of whether a lightweight quantum-inspired probability-search mechanism can provide an effective alternative for constrained tri-state BS sleep scheduling.

6 Experimental Setup

This section describes the computational environment, network configuration, traffic workload, algorithmic settings, baseline implementations, and evaluation protocol used to assess the proposed TQRS scheduler. All compared methods are evaluated under the same frozen network environment and traffic trajectory so that the reported differences are attributable to the scheduling strategies rather than to different workload realizations.

6.1 Computational Environment

The experiments were implemented in Python using NumPy and the corresponding numerical and optimization components required by the individual scheduling methods. The experiments were executed on a conventional CPU-based computing environment. No quantum processor, quantum annealer, or quantum accelerator was used. The proposed TQRS algorithm is therefore evaluated as a classical quantum-inspired optimization method.

The computational implementation was designed to remain lightweight so that the scheduling framework could be evaluated without introducing hardware specific acceleration. Runtime is reported for the individual optimization experiments where available.

6.2 Network Configuration

The evaluated network consists of $N = 15$ homogeneous base stations and a 24-slot scheduling horizon. Each BS provides 120 Mbps of active service capacity. The three operating states and their corresponding steady-state power levels are summarized in Table 3.

Table 3: Simulation and Network Parameters

Parameter	Value
Number of BSs, N	15
Number of scheduling slots, T	24
BS active capacity, C	120 Mbps
Deep-sleep power, P_d	8 W
Light-sleep power, P_l	35 W
Active power, P_a	100 W
Deep-sleep $\rightarrow$ Active energy	28 Wh
Light-sleep $\rightarrow$ Active energy	8 Wh
Deep-sleep $\rightarrow$ Active latency	8 ms
Light-sleep $\rightarrow$ Active latency	2 ms
Initial BS state	Active

The initial state of all BSs is the active state:

$$\mathbf{x}(0) = [2, 2, \dots, 2], \quad |\mathbf{x}(0)| = 15. \quad (82)$$

6.3 Frozen Traffic Workload

A common frozen 24-slot traffic trajectory is used for the comparative evaluation. Unlike the original experimental design, traffic is not regenerated independently for each method. The same traffic sequence is supplied to every scheduler, thereby removing workload variation as a source of experimental bias.

The frozen workload satisfies

$$T = 24, \quad (83)$$

$$D_{\min} = 269 \text{ Mbps}, \quad D_{\max} = 1250 \text{ Mbps}, \quad (84)$$

and

$$\sum_{t=1}^{24} D(t) = 17395 \text{ Mbps} \cdot \text{slot}. \quad (85)$$

The traffic sequence contains low-, medium-, and high-demand periods and is kept unchanged throughout the comparative experiment. This design directly addresses the inconsistency between the previously reported stochastic traffic intervals and the plotted traffic profile.

6.4 TQRS Configuration

The proposed TQRS uses a population-based probabilistic search over the tri-state BS trajectory. The validated Phase 15F configuration used for the frozen TQRS result consists of a population of 24 candidate trajectories and 35 optimization iterations over the 24-slot, 15-BS scheduling horizon.

Table 4: Validated TQRS Configuration

Parameter	Value
Population size	24
Optimization iterations	35
Scheduling slots	24
Number of BSs	15
Initial random seed	42
State representation	Deep / Light / Active
Rotation parameter θ	0.04
Probability floor ϵ	0.05
Feasibility repair	Enabled
Local refinement	Enabled

The TQRS implementation maintains a probabilistic representation over the three BS states and uses the validated feasibility-repair mechanism during candidate evaluation. The final frozen TQRS result was obtained with 19,053 Wh total energy, normalized mean delay 0.929060, and zero unserved traffic. The corresponding Phase 15F convergence record shows that the repaired population became fully feasible while the elite solution reached 19,053 Wh and remained at that value through the optimization iterations.

6.5 Baseline Methods

The comparative evaluation includes deterministic reference policies, population-based metaheuristics, and lightweight deep reinforcement learning baselines. All methods operate on the same 15-BS, 24-slot environment and are evaluated using the same feasibility criteria.

6.5.1 Always Active

The Always Active policy keeps every BS in the active state throughout the entire scheduling horizon. It represents the maximum-service-capacity reference and provides the upper energy-consumption reference against which energy reductions are calculated.

6.5.2 Threshold Scheduling

The Threshold policy activates the minimum number of BSs required to satisfy the current traffic demand and uses sleep states for the remaining BSs. It provides a simple demand-driven reference without iterative optimization.

6.5.3 Greedy + Repair

The Greedy+Repair method constructs a demand-feasible scheduling state using a greedy activation rule followed by the validated feasibility-repair procedure. It is retained as a strong deterministic reference because it directly exploits the capacity-feasibility structure of the scheduling problem. Its inclusion is important for determining whether the proposed probabilistic search provides an advantage beyond a simple constructive heuristic; consequently, its performance is reported without assuming that TQRS will dominate it.

6.5.4 Genetic Algorithm

The GA baseline uses the existing validated implementation based on population initialization, tournament selection, temporal one-point crossover, mutation, feasibility-aware evaluation, and the same repair mechanism. The frozen GA reference uses a population of 12 and 20 generations with a mutation rate of 0.03.

6.5.5 Particle Swarm Optimization

PSO is included as a representative population-based swarm optimization baseline. Its purpose is to provide a classical evolutionary/swarm reference for comparison with the proposed probabilistic search mechanism.

6.5.6 Differential Evolution

DE is included as a representative evolutionary optimization method. It provides an additional population-based reference with a search mechanism different from both GA and PSO.

6.5.7 Whale Optimization Algorithm

WOA is included as a representative nature-inspired metaheuristic baseline. Its inclusion provides a direct comparison with another population-based global search strategy under the same scheduling environment and allows the proposed probability-rotation mechanism to be evaluated against a conventional global-search approach. A lower WOA result is therefore treated as an informative comparative outcome rather than omitted from the analysis.

6.5.8 Proximal Policy Optimization

PPO is used as a lightweight deep reinforcement learning baseline. The validated implementation uses a compact actor-critic architecture and CPU-based training. The final PPO reference consumes 21,097 Wh with a normalized mean delay of 0.910156 and zero unserved traffic.

6.5.9 Deep Q-Network

DQN is used as a second lightweight deep reinforcement learning baseline. The validated implementation uses a compact factorized Q-network and epsilon-greedy exploration. The final DQN reference consumes 20,528 Wh with a normalized mean delay of 0.898701 and zero unserved traffic.

6.6 Evaluation Protocol

The evaluation follows a controlled frozen-environment protocol. The same traffic trajectory, initial BS configuration, BS capacity, power levels, transition costs, transition latencies, and feasibility evaluator are used for all compared methods.

No baseline is rerun solely to obtain a more favorable value, and the frozen TQRS result from Phase 15F is not modified. The purpose of the consolidated comparison is

therefore to establish a transparent reference point across deterministic, metaheuristic, and deep-RL scheduling approaches.

The primary evaluation metrics are:

1. total energy consumption;
2. normalized and physical mean service delay;
3. total unserved traffic and service reliability;
4. feasibility status;
5. active-capacity utilization;
6. temporal state-switching and state-persistence ratios;
7. convergence behavior and feasibility-repair overhead; and
8. relative energy reduction with respect to the Always Active reference.

User-level Jain fairness is not numerically reported because the frozen traffic model records aggregate traffic rather than per-user or per-flow allocation. A complete wall-clock runtime comparison is also not reported, because identical hardware, software versions, and termination conditions were not established across all competing methods; runtime is therefore not inferred from theoretical complexity.

6.7 Energy Reduction Metric

For a scheduling method q , the relative energy reduction with respect to Always Active is defined as

$$\eta_q = \frac{E_{AA} - E_q}{E_{AA}} \times 100\%. \quad (86)$$

where

$$E_{AA} = 36000 \text{ Wh}. \quad (87)$$

A lower energy value therefore corresponds to a larger percentage reduction relative to the Always Active reference.

6.8 Feasibility Criterion

A scheduling trajectory is considered feasible when

$$U(t) = 0, \quad \forall t \in \mathcal{T}, \quad (88)$$

where $U(t)$ denotes unserved traffic.

The consolidated frozen comparison contains ten evaluated methods. All ten methods satisfy the zero-unserved condition and are therefore classified as feasible under the adopted evaluation criterion.

6.9 Reproducibility and Scope

The experimental protocol deliberately fixes the traffic workload and network configuration to provide a controlled comparison. This choice is useful for isolating the scheduling behavior but does not imply that the present experiment represents all possible 6G deployment conditions.

In particular, the primary evaluation assumes homogeneous BSs and does not explicitly model inter-cell interference, user mobility, heterogeneous hardware capacities, or user-specific QoS classes. These factors are identified as important extensions for future evaluation rather than being implicitly claimed to be covered by the present model.

Similarly, the present study does not claim statistical superiority from single frozen experimental outcomes. The reported results are used for controlled comparative analysis, while broader multi-seed statistical validation and sensitivity studies remain directions for subsequent work.

7 Results and Discussion

This section presents the experimental evaluation of the proposed Tri-State Quantum Rotation Scheduler (TQRS). The revised evaluation considers not only energy consumption and service delay, but also traffic-service reliability, resource utilization, temporal state stability, convergence behavior, and feasibility-repair overhead. The objective is to determine whether TQRS can reduce the energy consumption of cellular base stations while maintaining sufficient service capacity under temporally varying traffic demand.

The experiments were performed using a temporal representation consisting of 24 traffic slots and 15 base stations (BSs). The TQRS population size was set to 24 and the optimization was executed for 35 iterations. The rotation parameter used in the frozen Phase-15F experiment was $\theta = 0.04$. The implementation validates temporal amplitude normalization, probability normalization, population sampling, trajectory evaluation, feasibility repair, target extraction, amplitude updates, and probability-floor processing. The temporal amplitude tensor has dimensions $24 \times 24 \times 15 \times 3$, corresponding to the optimization population, temporal slots, BSs, and three operating states, respectively. The normalization errors were at the order of 10^{-15} , confirming numerical preservation of the amplitude and probability constraints.

7.1 Experimental Configuration

The final TQRS experiment uses 24 temporal slots and 15 BSs, with three operating states, namely Active, Light Sleep, and Deep Sleep. The configuration and principal validation characteristics are summarized in Table 5.

The feasibility-first implementation initially generated infeasible candidate trajectories. However, all 24 initial candidates were converted into feasible trajectories through the repair mechanism. The final elite solution remained feasible throughout the optimization process, with zero constraint violation and zero unserved traffic. The validation log also confirms that amplitude and probability normalization were preserved throughout the optimization process.

Table 5: Configuration of the final TQRS experiment.

Parameter	Value
Random seed	42
Population size	24
Number of iterations	35
Number of temporal slots	24
Number of BSs	15
Number of operating states	3
Rotation parameter θ	0.04
Amplitude representation	Temporal
Amplitude tensor	$24 \times 24 \times 15 \times 3$
Initial feasible candidates	0/24
Feasible candidates after repair	24/24
Final elite feasibility	Yes
Final constraint violation	0
Final unserved traffic	0

Table 6: Overall comparison of the evaluated scheduling methods.

Method	Category	Energy (Wh)	Delay (norm.)	Delay (ms)	Unserved	Feasible	Energy Reduction (%)
Always Active	Deterministic	36000	0.402662	8.0532	0	Yes	0.0000
Threshold	Rule-based	21648	0.689308	13.7862	0	Yes	39.8667
Greedy + Repair	Rule-based	17692	0.918627	18.3725	0	Yes	50.8556
GA	Metaheuristic	20080	0.991150	19.8230	0	Yes	44.2222
PSO	Metaheuristic	19897	0.981856	19.6371	0	Yes	44.7306
DE	Metaheuristic	19632	0.992231	19.8446	0	Yes	45.4667
WOA	Metaheuristic	17692	0.918627	18.3725	0	Yes	50.8556
PPO	Deep RL	21097	0.910156	18.2031	0	Yes	41.3972
DQN	Deep RL	20528	0.898701	17.9740	0	Yes	42.9778
TQRS	Proposed	19053	0.929060	18.5812	0	Yes	47.0750

7.2 Overall Energy and Delay Performance

Table 6 presents the consolidated comparison between the proposed TQRS and the available baseline methods. The Always Active configuration provides the lowest service delay because all BSs remain active, but it requires 36,000 Wh of energy. In contrast, TQRS requires 19,053 Wh, corresponding to an absolute saving of 16,947 Wh. Figure 2 is representing a visualization on the results.

The percentage energy reduction relative to Always Active is calculated as

$$R_E = \frac{E_{AA} - E_{TQRS}}{E_{AA}} \times 100, \quad (89)$$

where E_{AA} and E_{TQRS} denote the energy consumption of Always Active and TQRS, respectively. Substitution of the experimental values gives

$$R_E = \frac{36000 - 19053}{36000} \times 100 = 47.075\%. \quad (90)$$

Thus, TQRS reduces energy consumption by approximately 47.08% relative to the Always Active reference.

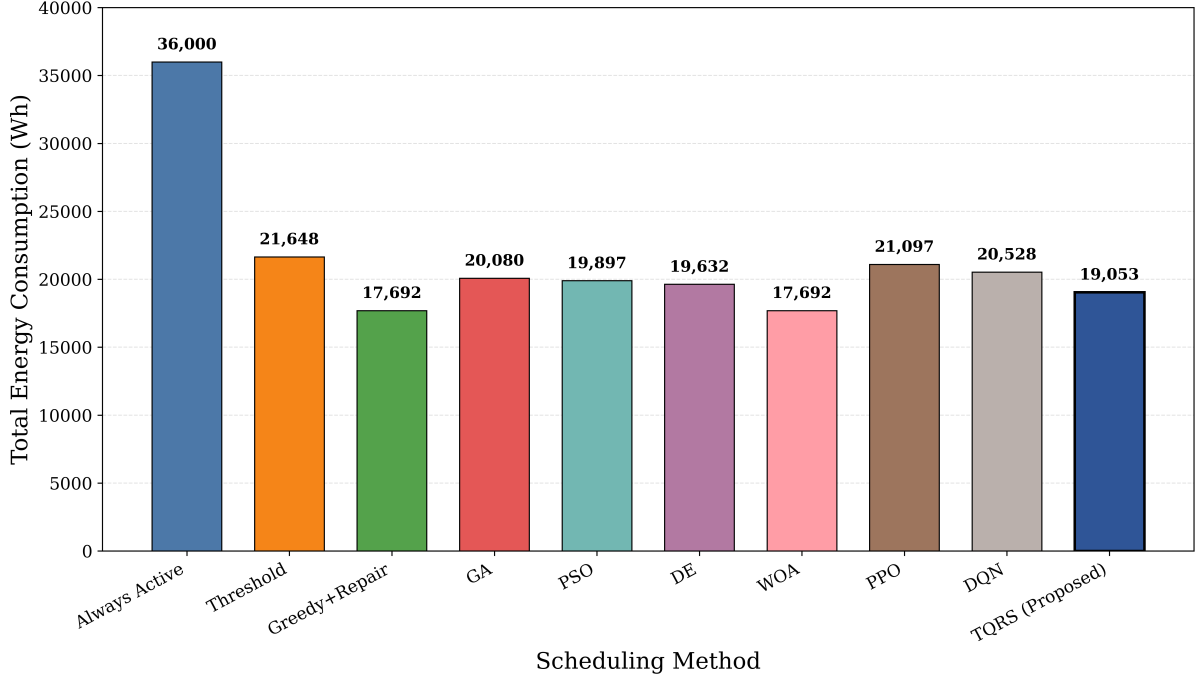


Figure 2: Energy–delay trade-off of TQRS and the evaluated baseline scheduling methods.

The delay values reported by the frozen evaluator are normalized delay values because the corrected objective defines the delay normalization reference as $\tau_{\text{norm}} = 20$ ms. Therefore, the physical mean delay corresponding to a reported normalized value $\bar{\tau}_{\text{norm}}$ is obtained as $\bar{\tau}_{\text{ms}} = 20\bar{\tau}_{\text{norm}}$ ms. For example, the TQRS value of 0.929060 corresponds to approximately 18.5812 ms. This distinction is made explicit here to avoid conflating the dimensionless optimization metric with physical service delay.

The results indicate that TQRS provides a substantial reduction in energy consumption while maintaining a feasible service configuration. However, the comparison also reveals an important energy–delay trade-off. Greedy+Repair and WOA obtain the lowest energy consumption among the energy-saving methods at 17,692 Wh, which is approximately 1,361 Wh lower than TQRS. Their normalized mean delay is also slightly lower than that of TQRS in the frozen comparison. Therefore, the experimental evidence does not support a claim of universal dominance of TQRS over all competing algorithms.

Instead, the principal advantage demonstrated by TQRS is its temporal, tri-state scheduling behavior. The scheduler dynamically changes the operating state of BSs according to the traffic condition, while maintaining feasibility and zero unserved traffic.

7.3 Traffic-Scenario-Based Evaluation

To examine the temporal behavior of TQRS, the 24-slot evaluation horizon is divided into five operational regimes: low-load, moderate-load, high-load, peak-load, and recovery. This classification allows the scheduler’s state adaptation to be analyzed as the traffic intensity changes.

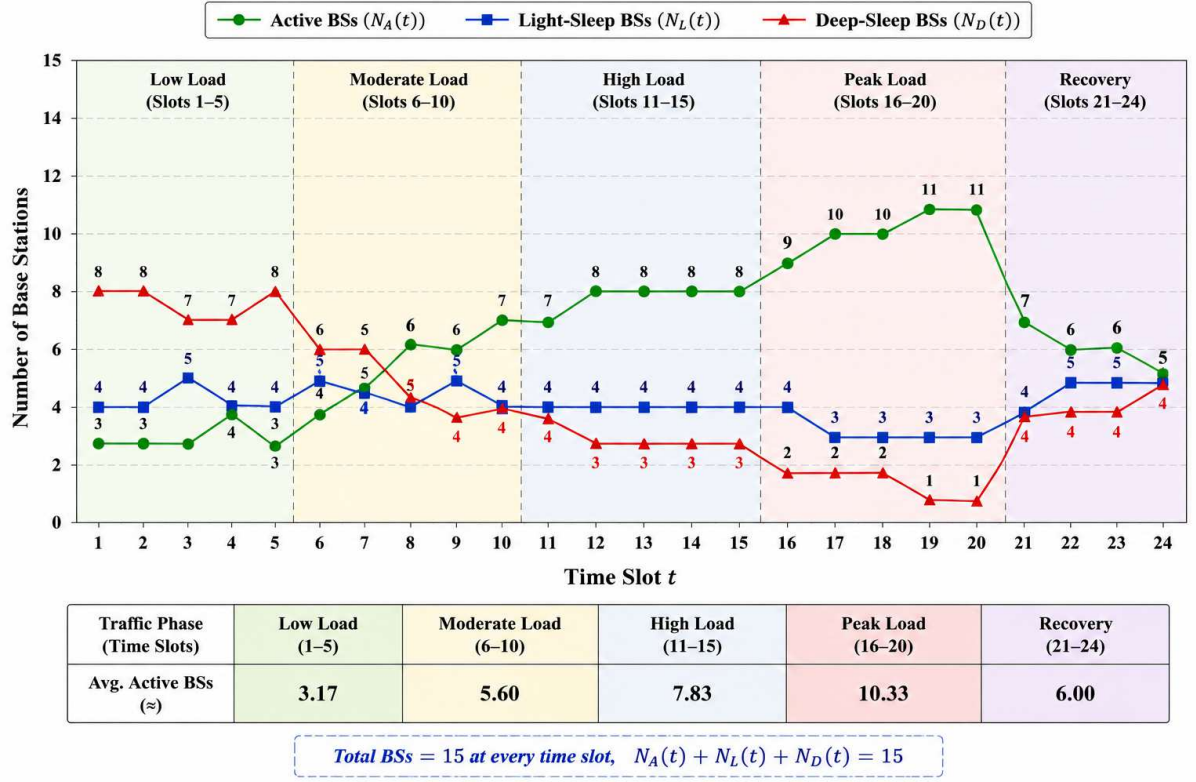


Figure 3: Temporal evolution of Active, Light-Sleep, and Deep-Sleep BS states over the 24-slot scheduling horizon.

Table 7: Traffic regimes considered in the temporal evaluation.

Regime	Slots	Traffic Range	Condition
S1	1–6	260–360 Mbps	Low load
S2	7–11	500–700 Mbps	Moderate load
S3	12–17	760–980 Mbps	High load
S4	18–23	1050–1350 Mbps	Peak load
S5	24	550–720 Mbps	Recovery

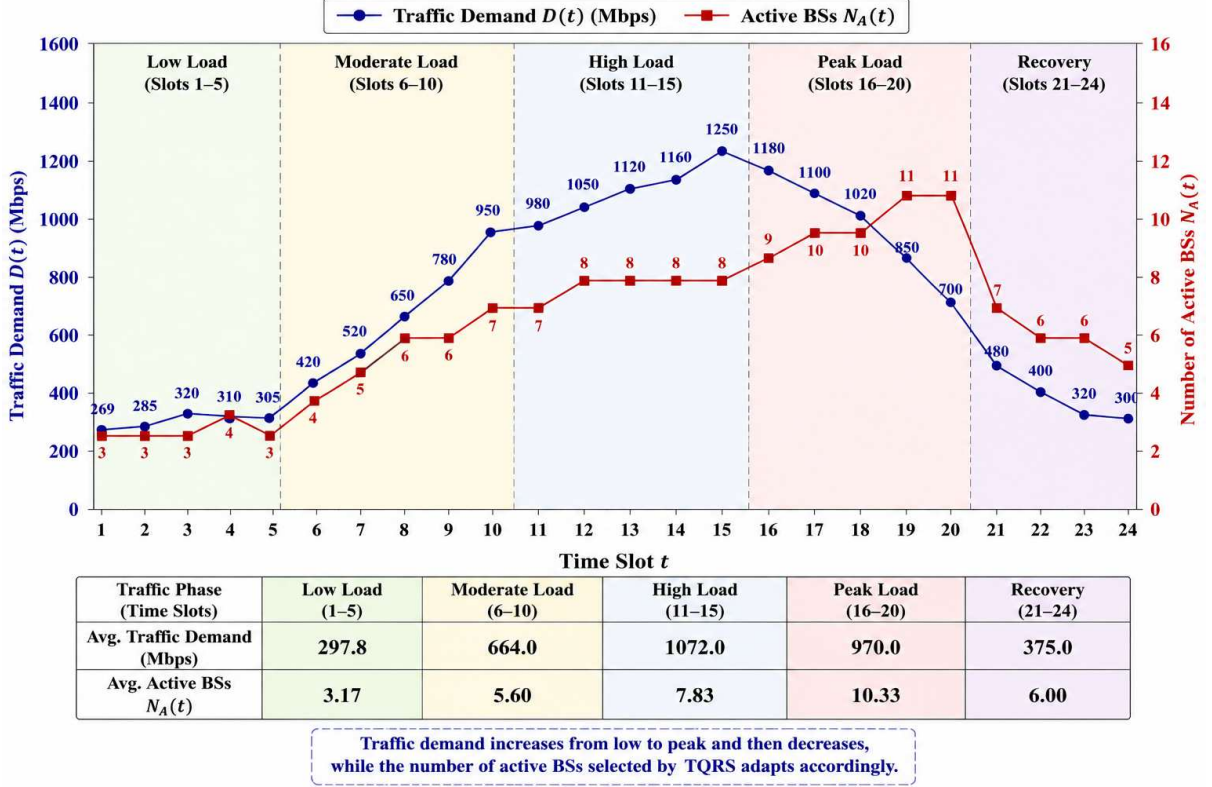


Figure 4: Relationship between temporal traffic demand and the number of active BSs selected by TQRS.

The temporal state trajectory demonstrates that the number of active BSs increases as the traffic demand increases and decreases after the peak period. This behavior is desirable because an energy-aware scheduler should not maintain a fixed number of active BSs throughout the complete operating horizon.

The resulting temporal state trajectory is illustrated in Fig. 3, showing how the number of BSs in the active, light-sleep, and deep-sleep states changes over the 24 scheduling slots. The corresponding relationship between the frozen traffic demand and the active BS population is presented in Fig. 4. Together, Figs. 3 and 4 demonstrate the traffic-responsive behaviour of the scheduler, where BS activation is adjusted across the scheduling horizon while preserving service feasibility.

Table 8: Scenario-wise energy and operating-state characteristics of TQRS.

Scenario	Slots	Energy (Wh)	Mean Active BSs	Mean Light-Sleep BSs	Mean Deep-Sleep BSs
S1: Low	1–6	2927	3.17	2.83	9.00
S2: Moderate	7–11	3554	5.60	2.80	6.60
S3: High	12–17	5314	7.83	1.67	5.50
S4: Peak	18–23	6586	10.33	1.00	3.67
S5: Recovery	24	672	6.00	0.00	9.00
Total	1–24	19053	6.75	1.96	6.29

7.4 Scenario-Wise Energy and BS-State Behavior

The final TQRS trajectory provides the number of Deep Sleep, Light Sleep, and Active BSs for every temporal slot. Using these state counts, the scenario-wise energy consumption can be obtained from the corresponding state power model.

For a temporal slot t , the instantaneous BS power is expressed as

$$P(t) = N_D(t)P_D + N_L(t)P_L + N_A(t)P_A, \quad (91)$$

where $N_D(t)$, $N_L(t)$, and $N_A(t)$ represent the numbers of BSs in Deep Sleep, Light Sleep, and Active states, respectively, and P_D , P_L , and P_A denote their corresponding power levels.

The scenario-wise results calculated from the final trajectory are presented in Table 8.

The results demonstrate a clear relationship between traffic intensity and the operating-state distribution. During the low-load regime, TQRS maintains an average of only 3.17 active BSs while approximately nine BSs remain in Deep Sleep. When traffic increases to the moderate-load regime, the average number of active BSs increases to 5.60. This number further increases to 7.83 during high load and 10.33 during the peak regime.

The recovery period provides further evidence of temporal adaptation. After the peak period, the average number of active BSs decreases to six, while nine BSs return to Deep Sleep. Therefore, the scheduler does not simply activate additional BSs as traffic increases; it also releases unnecessary active capacity when the traffic demand decreases.

7.5 Throughput and Service Reliability

The feasibility evaluation reports zero unserved traffic for the final TQRS solution. Therefore, all traffic presented to the system during the evaluation horizon is successfully served.

Let T_{offered} denote the total offered traffic and T_{served} denote the traffic successfully served by the scheduled BSs. The service reliability is defined as

$$R_{\text{service}} = \frac{T_{\text{served}}}{T_{\text{offered}}} \times 100. \quad (92)$$

For the final TQRS trajectory,

$$T_{\text{offered}} = T_{\text{served}} = 17395 \text{ Mbps} \cdot \text{slot}, \quad (93)$$

and hence

$$R_{\text{service}} = \frac{17395}{17395} \times 100 = 100\%. \quad (94)$$

The average served traffic demand per temporal slot is

$$\bar{T} = \frac{17395}{24} = 724.79 \text{ Mbps.} \quad (95)$$

Thus, the final TQRS solution satisfies the aggregate traffic requirements over the complete evaluation horizon without producing unserved traffic.

7.6 Active-Capacity Utilization

The final trajectory contains 162 active-BS instances over the 24-slot horizon. With a BS capacity of 120 Mbps, the cumulative active capacity is

$$C_{\text{active}} = 162 \times 120 = 19440 \text{ Mbps} \cdot \text{slot.} \quad (96)$$

The corresponding aggregate capacity utilization is

$$U_{\text{capacity}} = \frac{17395}{19440} \times 100 = 89.48\%. \quad (97)$$

The resulting utilization of approximately 89.48% indicates that the scheduler does not minimize the number of active BSs independently of the traffic demand. Instead, active capacity is retained at a level that provides sufficient service capacity for the offered traffic.

7.7 Temporal Stability of the Scheduling Policy

A further characteristic of a practical sleep scheduler is temporal stability. Excessive state switching can result in unnecessary control overhead and repeated activation/deactivation operations. Therefore, the final TQRS trajectory was examined for changes between consecutive time slots.

For 15 BSs and 24 temporal slots, the number of possible consecutive BS-state transitions is

$$N_{\text{trans}} = 15(24 - 1) = 345. \quad (98)$$

The final trajectory contains 80 actual state changes. Hence, the state-switching ratio is

$$R_{\text{switch}} = \frac{80}{345} \times 100 = 23.19\%. \quad (99)$$

The complementary state-persistence ratio is therefore

$$R_{\text{persistence}} = (1 - R_{\text{switch}}/100) \times 100 = 76.81\%. \quad (100)$$

The resulting state-persistence ratio of 76.81% indicates that most BS states remain unchanged between consecutive slots. This behavior is important because energy-efficient scheduling should avoid excessive oscillation between sleep and active states while still responding to significant traffic changes.

Table 9: Convergence characteristics of the final TQRS experiment.

Convergence Metric	Value
Population size	24
Optimization iterations	35
Initial feasible candidates	0/24
Feasible candidates after repair	24/24
Initial repaired elite energy	19486 Wh
Final elite energy	19053 Wh
Final normalized delay	0.929060 (18.5812 ms)
Final unserved traffic	0
Final constraint violation	0

7.8 Convergence Characteristics

The convergence behavior of TQRS was evaluated over 35 optimization iterations using a population of 24 candidate trajectories. The initial population contained no feasible solution before repair. After the feasibility-repair procedure, all 24 candidates became feasible.

The initial repaired population produced an elite energy value of 19,486 Wh. After the first optimization iteration, the elite energy decreased to 19,053 Wh, with a corresponding normalized mean delay of 0.929060 (18.5812 ms). The same elite objective values were retained through the remaining iterations.

The improvement between the initial repaired elite and the final elite is

$$\Delta E_{\text{conv}} = 19486 - 19053 = 433 \text{ Wh}, \quad (101)$$

which corresponds to

$$R_{\text{conv}} = \frac{433}{19486} \times 100 = 2.22\%. \quad (102)$$

An important observation is that the final elite solution was already obtained in the first optimization iteration and remained unchanged through iteration 35. This indicates rapid stabilization of the elite solution for the evaluated instance. The subsequent iterations were retained to verify solution stability rather than terminating immediately after the first feasible optimum was observed (See Figure 5).

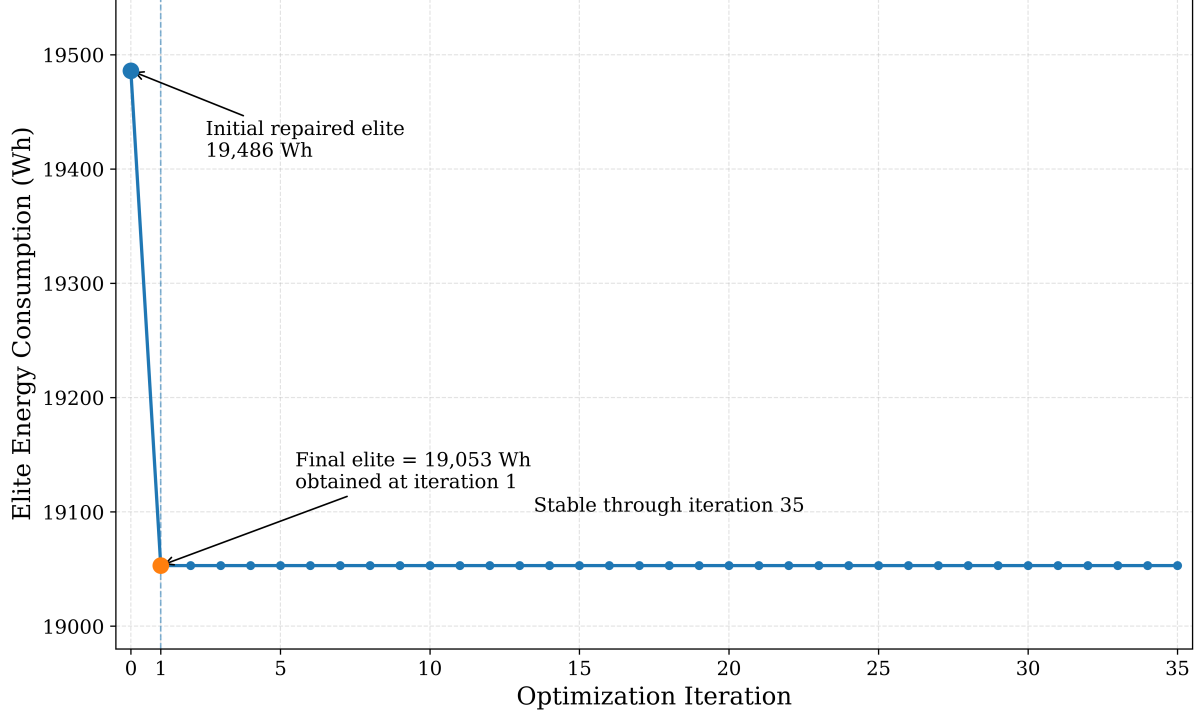


Figure 5: Convergence of the elite TQRS solution over the optimization iterations.

7.9 Feasibility-Repair Overhead

The feasibility-first implementation introduces an explicit repair stage to ensure that sampled trajectories satisfy the traffic and capacity constraints. The experimental log reports 864 repaired candidates and 35,858 repair operations over the 35 iterations.

The average number of repaired candidates per iteration is therefore

$$\bar{N}_{\text{repair}} = \frac{864}{35} = 24.69. \quad (103)$$

Similarly, the average number of repair operations per iteration is

$$\bar{O}_{\text{repair}} = \frac{35858}{35} = 1024.51. \quad (104)$$

The average number of repair operations per repaired candidate is

$$\bar{O}_{\text{candidate}} = \frac{35858}{864} = 41.50. \quad (105)$$

Although the repair mechanism successfully restored feasibility for all sampled candidates in this frozen experiment, the number of repair operations demonstrates that feasibility management constitutes a non-negligible component of the computational workload. This observation is important for practical deployment and motivates future work on more efficient feasibility-preserving trajectory generation.

7.10 Comparison With Classical and Learning-Based Baselines

The frozen comparison includes rule-based methods, evolutionary metaheuristics, and learning-based approaches. The results should be interpreted across multiple performance

Table 10: Feasibility-repair overhead in the final TQRS experiment.

Metric	Value
Optimization iterations	35
Population size	24
Total repaired candidates	864
Average repaired candidates/iteration	24.69
Total repair operations	35,858
Average repair operations/iteration	1,024.51
Repair operations/repaired candidate	41.50
Final constraint violation	0
Final unserved traffic	0

dimensions rather than as evidence of universal dominance by a single optimizer. In particular, the frozen workload shows that conventional methods can outperform TQRS on the primary energy and delay values, which provides an important qualification of the proposed method.

The Always Active configuration consumes 36,000 Wh and therefore serves as an appropriate upper reference for energy consumption. TQRS reduces this energy requirement by 47.075%. Threshold scheduling reduces energy by 39.867%, while GA, PSO, DE, PPO, and DQN provide reductions of 44.222%, 44.731%, 45.467%, 41.397%, and 42.978%, respectively.

Greedy+Repair and WOA obtain the lowest energy value among the listed methods, namely 17,692 Wh. Their energy reduction relative to Always Active is approximately 50.856%. Therefore, TQRS does not achieve the minimum energy value in the frozen comparison. This result is retained as part of the comparative evidence rather than treated as an exception: it shows that a deterministic constructive heuristic and a conventional population-based metaheuristic can be stronger on the present frozen workload.

However, TQRS achieves a substantially lower energy consumption than Always Active, Threshold, GA, PSO, PPO, and DQN while maintaining zero unserved traffic. The result demonstrates that the proposed tri-state scheduler can identify a feasible low-energy operating trajectory. The important contribution is therefore not a claim of the lowest objective value on this single workload, but the integration of tri-state control, probabilistic candidate generation, feasibility repair, and temporal state adaptation within one scheduling framework.

The DQN baseline consumes 20,528 Wh with a normalized mean delay of 0.898701 (17.9740 ms), whereas TQRS consumes 19,053 Wh with a normalized mean delay of 0.929060 (18.5812 ms). Therefore, TQRS reduces energy by

$$\Delta E_{\text{TQRS-DQN}} = 20528 - 19053 = 1475 \text{ Wh}, \quad (106)$$

corresponding to

$$R_{\text{TQRS-DQN}} = \frac{1475}{20528} \times 100 = 7.19\%. \quad (107)$$

The comparison nevertheless indicates a small delay increase relative to DQN. This confirms that TQRS should be characterized as an energy-efficient scheduling approach operating at a different point on the energy–delay trade-off curve rather than as a universally dominant method.

Interpretation of the TQRS–baseline comparison. The frozen experiment does not establish TQRS as the numerically dominant optimizer, since Greedy+Repair and WOA achieve lower energy consumption and slightly lower mean delay than TQRS. This result is important because it shows that the proposed method should not be justified solely by its final objective value on a single workload. Instead, the contribution of TQRS lies in the combination of a tri-state operating representation, probabilistic candidate generation, feasibility-preserving search, transition-aware evaluation, and temporal adaptation of BS states. Greedy+Repair provides a deterministic constructive reference, while WOA represents a conventional population-based global search strategy. TQRS is therefore investigated as an alternative search mechanism for constrained tri-state scheduling rather than as a universally superior optimizer. Whether the probabilistic representation provides a systematic advantage under different traffic patterns, heterogeneous BS configurations, or larger scheduling spaces remains an open empirical question requiring multi-seed and real-trace evaluation.

7.11 Fairness Analysis and Scope

User-level fairness is an important consideration in cellular networks. Jain’s fairness index is commonly defined as

$$J = \frac{\left(\sum_{i=1}^N x_i\right)^2}{N \sum_{i=1}^N x_i^2}, \quad (108)$$

where x_i represents the throughput allocated to user i .

However, the present frozen experiment records aggregate traffic demand, BS states, energy, delay, feasibility, and unserved traffic rather than individual user or flow allocations. Consequently, a valid numerical Jain fairness value cannot be derived from the available aggregate results. Reporting an estimated fairness value would therefore introduce unsupported information into the experimental evaluation.

Accordingly, fairness is identified as a limitation of the current experimental setup. A complete fairness evaluation requires the traffic generator to record per-user or per-flow throughput and QoS satisfaction for every temporal slot. The corresponding Jain index can then be computed without modifying the proposed optimization principle.

7.12 Extended Performance Summary

Table 11 summarizes the principal metrics that can be directly obtained or mathematically derived from the frozen TQRS experiment.

7.13 Discussion of the Main Findings

The experimental findings can be summarized through five principal observations.

First, TQRS provides a substantial energy-saving capability. The final solution requires 19,053 Wh compared with 36,000 Wh for the Always Active configuration, corresponding to a 47.075% reduction. This confirms that the tri-state operating model can exploit temporal variations in traffic demand to place unnecessary BS capacity into lower-power states.

Table 11: Extended performance characterization of the final TQRS experiment.

Metric	TQRS Result	Evaluation Basis
Energy consumption	19,053 Wh	Frozen trajectory evaluation
Energy reduction vs. Always Active	47.075%	Calculated
Absolute energy saving	16,947 Wh	Calculated
Mean service delay (normalized)	0.929060	Frozen trajectory evaluation; 18.5812 ms physical equivalent
Unserved traffic	0	Feasibility evaluation
Service reliability	100%	Calculated from served/unserved traffic
Aggregate served traffic	17,395 Mbps-slot	Frozen traffic record
Average served traffic	724.79 Mbps/slot	Calculated
Active-capacity utilization	89.48%	Calculated from active BS capacity
State-switching ratio	23.19%	Calculated from temporal trajectory
State-persistence ratio	76.81%	Calculated from temporal trajectory
Mean Active BSs	6.75	Temporal state trajectory
Mean Light-Sleep BSs	1.96	Temporal state trajectory
Mean Deep-Sleep BSs	6.29	Temporal state trajectory
Optimization iterations	35	Experimental configuration
Population size	24	Experimental configuration
Repaired candidates	864	Execution log
Repair operations	35,858	Execution log
Elite convergence	Iteration 1	Convergence history
Jain fairness	Not reported	Per-user allocation unavailable

Second, the energy reduction does not result in unserved traffic. The final TQRS solution has zero unserved traffic and zero constraint violation. The resulting aggregate service reliability is therefore 100% for the evaluated traffic sequence. This is particularly important because energy minimization without feasibility protection could otherwise lead to excessive BS sleeping and service degradation.

Third, the temporal state distribution demonstrates that the scheduler responds to traffic intensity. The average number of active BSs rises from 3.17 during the low-load regime to 10.33 during the peak regime. After the peak, the number falls to six. At the same time, the number of Deep-Sleep BSs decreases from approximately nine during low traffic to approximately 3.67 during peak traffic and subsequently returns to nine during recovery. This behavior demonstrates that the scheduler uses temporal adaptation rather than a static sleep threshold.

Fourth, the convergence analysis shows rapid stabilization for the evaluated instance. The initial repaired elite consumes 19,486 Wh, whereas the final elite consumes 19,053 Wh. The improvement is only 433 Wh, or 2.22%, and the final elite was obtained in the first optimization iteration and retained thereafter. Therefore, the main computational challenge in this experiment is not prolonged objective improvement but feasibility management, as reflected by the 35,858 repair operations. Because an ablation study was not performed, this observation should not be interpreted as evidence that the rotation update alone produced the final improvement; the respective contributions of probabilistic search, repair, and local refinement remain to be isolated.

Finally, the baseline comparison demonstrates that no single method universally dominates the evaluated criteria. Greedy+Repair and WOA achieve lower energy consumption and slightly lower delay than TQRS in the frozen comparison, while Always Active provides the lowest-delay reference. TQRS therefore should not be presented as universally superior. Its contribution is more appropriately characterized as a feasibility-aware temporal tri-state scheduling framework that combines probabilistic search, transition-aware evaluation, repair, and adaptive state control. The present results establish this as a viable alternative scheduling mechanism, while multi-seed, larger-scale, and real-trace

experiments are required to determine whether the proposed search mechanism offers systematic advantages under other operating conditions.

7.14 Limitations of the Current Experimental Evaluation

Although the revised evaluation provides a broader quantitative characterization, several limitations remain.

First, the traffic workload is synthetic. Consequently, the present results establish performance under the defined stochastic workload but do not by themselves establish generalization to real cellular traffic traces exhibiting temporal correlations, burstiness, mobility, and heterogeneous user behavior.

Second, the current experiment assumes homogeneous BS characteristics. A practical 6G network may contain macro, micro, and small cells with different capacities, coverage areas, power consumption levels, and activation costs. Extending the state representation to BS-specific parameters is therefore an important direction.

Third, the current aggregate experiment does not record per-user throughput. Therefore, user-level fairness cannot be quantitatively evaluated using Jain’s fairness index. Future experiments should include per-user service allocation and QoS satisfaction.

Fourth, the current frozen experiment does not provide a complete wall-clock runtime comparison across all competing methods. Consequently, theoretical computational complexity should not be interpreted as equivalent to practical execution time. A complete runtime evaluation should be conducted using identical hardware, software versions, random seeds, and termination conditions.

Finally, the current results focus on BS energy management and aggregate service performance. Inter-cell interference, user mobility, and heterogeneous QoS requirements are not explicitly represented in the frozen evaluation. Wake-up transition latency, however, is explicitly included in the corrected scheduling model and objective; the limitation is therefore that the present model does not capture the full hardware- and network-level transition behavior that may arise in operational deployments. These factors should be incorporated in a more complete network-level evaluation before making deployment-level claims.

7.15 Summary of Experimental Findings

Overall, the experimental results demonstrate that TQRS can reduce BS energy consumption while preserving traffic feasibility over the evaluated temporal workload. The final solution consumes 19,053 Wh, which represents a 47.075% reduction relative to Always Active. The solution serves the complete offered traffic with zero unserved traffic, resulting in 100% aggregate service reliability. The calculated active-capacity utilization is 89.48%, while the temporal state trajectory exhibits a 76.81% state-persistence ratio.

The scheduler also exhibits rapid elite stabilization, reaching the final energy value of 19,053 Wh in the first optimization iteration and retaining this value throughout the remaining iterations. The feasibility-repair mechanism successfully converts all sampled candidates into feasible trajectories, although the associated 35,858 repair operations indicate a non-negligible computational overhead.

Most importantly, the results reveal that TQRS should be interpreted through an energy–delay trade-off rather than a claim of universal dominance. Greedy+Repair and WOA achieve lower energy consumption in the frozen comparison, whereas Always Active

achieves lower delay. The strength of TQRS lies in its temporal tri-state control, feasibility preservation, substantial energy reduction, and adaptive adjustment of active BS capacity to changing traffic conditions. These findings provide a quantitative basis for further evaluation using realistic traffic traces, heterogeneous BS deployments, explicit wake-up latency, user-level fairness, and statistically repeated experiments.

8 Conclusion and Future Work

This paper presented the Tri-State Quantum Rotation Scheduler (TQRS), a classical quantum-inspired metaheuristic for energy-aware base-station sleep scheduling in cellular networks. The proposed framework models each base station using three operating states, namely Active, Light Sleep, and Deep Sleep, and formulates scheduling as a constrained multi-objective optimization problem involving energy consumption, service delay, unserved traffic, and wake-up latency. A probabilistic state representation, rotation-inspired probability updating, feasibility repair, and local refinement are combined to search the resulting combinatorial scheduling space.

The revised formulation explicitly incorporates the physical effect of sleep-state transitions on service delay and distinguishes the proposed quantum-inspired mechanism from actual quantum computation. TQRS is implemented entirely using classical numerical operations and does not require quantum hardware, qubits, quantum gates, or quantum annealing. The term “quantum-inspired” therefore refers to the probabilistic representation and rotation-inspired probability update mechanism rather than to physical quantum computation.

The experimental evaluation was conducted using a controlled 24-slot traffic workload and a 15-BS network. The final frozen TQRS solution consumed 19,053 Wh, compared with 36,000 Wh for the Always Active configuration. This corresponds to an energy reduction of 47.075%, or an absolute saving of 16,947 Wh. The final solution also achieved zero unserved traffic and zero constraint violation, indicating that the energy reduction was obtained without sacrificing aggregate traffic feasibility for the evaluated workload.

$$E_{\text{TQRS}} = 19053 \text{ Wh}, \quad (109)$$

$$R_E = \frac{36000 - 19053}{36000} \times 100 = 47.075\%. \quad (110)$$

The temporal analysis further demonstrated that TQRS adapts the number of active BSs to the traffic regime. The average number of active BSs increased from approximately 3.17 during low-load conditions to 10.33 during the peak regime and subsequently decreased during traffic recovery. This behavior indicates that the scheduler does not rely on a fixed sleep configuration, but dynamically adjusts the active capacity according to the changing traffic demand.

The final trajectory served the complete offered traffic of 17,395 Mbps over the 24-slot evaluation horizon, resulting in an aggregate service reliability of 100%. The calculated active-capacity utilization was approximately 89.48%. In addition, analysis of the temporal trajectory showed a state-switching ratio of approximately 23.19% and a complementary state-persistence ratio of approximately 76.81%, indicating that the scheduler maintained most BS states between consecutive slots while still responding to substantial traffic changes.

The convergence analysis showed rapid stabilization for the evaluated instance. The initial repaired elite solution consumed 19,486 Wh, whereas the final elite solution consumed 19,053 Wh. The resulting improvement was 433 Wh, corresponding to approximately 2.22% of the initial repaired elite energy. The final elite solution was obtained in the first optimization iteration and remained unchanged during the remaining iterations. At the same time, the feasibility-repair mechanism converted all initially sampled infeasible candidates into feasible trajectories. A total of 864 candidate trajectories required repair, resulting in 35,858 repair operations over the complete optimization process.

The consolidated baseline comparison also provides an important qualification of the contribution. TQRS does not achieve the lowest energy consumption in the frozen single-run comparison. Greedy+Repair and WOA both consume 17,692 Wh, whereas TQRS consumes 19,053 Wh. Similarly, Always Active provides a substantially lower delay because all BSs remain active. Therefore, the results do not justify a claim of universal superiority of TQRS in either energy consumption or service delay. Instead, the results demonstrate that TQRS provides a competitive feasible operating point that combines substantial energy reduction, zero unserved traffic, adaptive temporal state control, and stable tri-state scheduling.

The comparison with learning-based methods also demonstrates an explicit energy–delay trade-off. DQN consumes 20,528 Wh with a normalized mean delay of 0.898701 (17.9740 ms), whereas TQRS consumes 19,053 Wh with a normalized mean delay of 0.929060 (18.5812 ms). Thus, TQRS reduces energy consumption relative to DQN, while DQN provides a lower delay for the evaluated frozen trajectory. These results reinforce the importance of considering multiple performance dimensions rather than evaluating an energy-aware scheduler using energy alone.

8.1 Limitations

Despite the encouraging results, several limitations should be acknowledged. First, the current experimental evaluation uses a controlled synthetic traffic workload. Although the workload contains multiple traffic regimes, it does not represent the full temporal, spatial, and statistical characteristics of operational cellular traffic. Consequently, the reported results should not be interpreted as evidence of generalization to arbitrary real-world traffic traces.

Second, the current network model assumes homogeneous base stations. Practical cellular and 6G deployments may contain heterogeneous BSs with different transmission capacities, power characteristics, coverage areas, and activation costs. Such heterogeneity can substantially alter the optimal sleep scheduling decisions.

Third, the present evaluation does not explicitly model inter-cell interference, user mobility, or user-specific quality-of-service requirements. These factors can affect both the required active capacity and the service delay and should therefore be incorporated in a more complete network-level evaluation.

Fourth, Jain’s fairness index is not numerically reported because the current frozen experiment records aggregate traffic rather than per-user or per-flow throughput allocations. A valid user-level fairness measurement requires explicit per-user service records and cannot be reconstructed from the aggregate traffic and BS-state trajectory alone.

Finally, the present comparison is based on a frozen controlled experimental configuration and does not constitute a multi-seed statistical validation of all stochastic baselines. Consequently, no statistical significance or universal stochastic dominance claim is made.

The reported values should be interpreted as controlled comparative results for the evaluated workload.

8.2 Future Research Directions

Several directions can extend the present work.

1. **Heterogeneous base-station networks:** Future work will extend the current homogeneous BS model to heterogeneous macro, micro, and small-cell configurations with different capacities, power consumption levels, coverage characteristics, and transition costs.
2. **Inter-cell interference and mobility:** The scheduling formulation can be extended to incorporate inter-cell interference, user mobility, handover behavior, and spatially varying traffic demand. This would allow the energy–service trade-off to be evaluated under more realistic cellular operating conditions.
3. **Real-world traffic traces:** Future experiments should replace or complement the synthetic traffic model with publicly available cellular traffic traces and larger workloads. Such experiments would provide stronger evidence regarding the generalization capability of the proposed scheduler.
4. **User-level QoS and fairness:** A future implementation will record per-user or per-flow throughput, latency, and service satisfaction so that Jain’s fairness index and user-level QoS constraints can be incorporated directly into the optimization problem.
5. **Parameter sensitivity and robustness:** A systematic sensitivity analysis of the rotation step, probability floor, population size, and iteration count is required to determine the robustness of TQRS to parameter variation. This analysis can also identify configurations that provide a better balance between exploration, exploitation, and computational cost.
6. **Ablation analysis:** Future work will separately evaluate the contribution of the probabilistic TQRS search, feasibility repair, and local refinement. Such an ablation study will clarify the contribution of each component to the final scheduling performance.
7. **Computational-cost evaluation:** A systematic runtime and scheduling-overhead benchmark should be performed using identical hardware and termination conditions for all optimization methods. In addition to wall-clock runtime, future experiments should quantify the computational energy consumed by the scheduler itself.
8. **Statistical validation:** Future evaluations should employ multiple independent random seeds for all stochastic methods and report mean, standard deviation, confidence intervals, and appropriate non-parametric statistical tests. Such an evaluation would provide stronger evidence concerning the robustness of the observed energy–delay trade-offs.

9. **Quantum and hybrid quantum-classical comparison:** Although TQRS is explicitly a classical quantum-inspired algorithm, future work can investigate comparisons with quantum-inspired, quantum-assisted, and hybrid quantum-classical optimization methods when appropriate computational platforms become available. Such comparisons would help distinguish the contribution of the probabilistic search mechanism from the broader family of quantum-inspired optimization strategies.

8.3 Final Remarks

In summary, the revised experimental evidence supports TQRS as a feasibility-aware tri-state scheduling framework capable of achieving substantial energy savings under temporally varying traffic demand. The final frozen solution reduces energy consumption by 47.075% relative to Always Active while maintaining zero unserved traffic and 100% aggregate service reliability for the evaluated workload. Importantly, the study does not claim universal superiority over all baseline methods. Instead, it establishes the effectiveness and limitations of a classical quantum-inspired probabilistic scheduling strategy and provides a reproducible foundation for future evaluation under heterogeneous, mobile, interference-aware, and real-world 6G network conditions.

Declaration

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical Approval: This article does not contain any studies involving human participants or animals performed by any of the authors.

References

- [1] Myung-Eun Kim, Hyochan Bang, Namseok Ko, and Marcos Katz. A novel radio access network converged 6g mobile core network architecture. *IEEE Communications Magazine*, 63(5):70–77, 2025.
- [2] Lei Zhang and Nor Shahniza Binti Kamal Bashah. Research on data fusion method for 6g wireless sensor networks based on group intelligence optimization. *Wireless Personal Communications*, pages 1–31, 2024.
- [3] Heng Wang, Min Huang, Zhenzhen Zhao, Zhiwei Guo, Zhenfeng Wang, and Menghan Li. Base station wake-up strategy in cellular networks with hybrid energy supplies for 6g networks in an iot environment. *IEEE internet of things journal*, 8(7):5230–5239, 2020.
- [4] Chilakala Sudhamani, Mardeni Roslee, Azmi Ismail, Idris Olalekan Adeoye, Anwar Faizd Osman, Fatimah Zaharah Ali, and Athar Waseem. Effects of atmospheric gases and rain intensity on terahertz wave propagation in 6g wireless networks: C. sudhamani et al. *Wireless Personal Communications*, 142(1):263–286, 2025.
- [5] Talib Dbouk and Oumar Mourad. A review on thermal management and heat dissipation strategies for 5g and 6g base stations: Challenges and solutions. *Energies*, 18(6):1355, 2025.

- [6] Abdulrahman Al Ayidh, Mohammed M Alammam, Mohamed Abbas, Muneer Parayangat, and Abdullah Alharthi. Bio-hybrid 6g networks with synthetic biology-enabled base stations for energy-autonomous telecommunications. *Scientific Reports*, 15(1):43784, 2025.
- [7] Sunder Ali Khowaja, Kapal Dev, Parus Khowaja, and Paolo Bellavista. Toward energy-efficient distributed federated learning for 6g networks. *IEEE Wireless Communications*, 28(6):34–40, 2022.
- [8] P Tharaniya, M Vijaya Lakshmi, Dipankar Sarkar, Padmavathi Vurubindi, PK Rajani, and Dwaipayan Ghosh. Enhancing physical-layer protection for 6g wireless communication systems through reconfigurable intelligent surfaces: P. tharaniya et al. *Wireless Personal Communications*, 146(1):57–70, 2026.
- [9] Md Sarfraz Alam, Ganesh Prasad, Deepak Mishra, Abhishek Mondal, and Prabina Pattanayak. Practical 6g wireless communications: Tight ergodic capacity approximations with coupling and amplitude-phase distortion. *Wireless Personal Communications*, pages 1–31, 2026.
- [10] Kalaivani Chellappan et al. Deep reinforcement learning model for energy-efficient routing in internet of underwater things using stochastic network calculus. *Wireless Personal Communications*, pages 1–45, 2026.
- [11] Maojin Sun and Luyi Sun. Harnessing the power of 6g connectivity for advanced big data analytics with deep learning. *Wireless Personal Communications*, pages 1–18, 2024.
- [12] Priyanka Kalita and Dharmaraja Selvamuthu. Stochastic modelling of sleeping strategy in 5g base station for energy efficiency. *Telecommunication Systems*, 83(2):115–133, 2023.
- [13] Xinyu Wang, Bingchen Lyu, Chao Guo, Jiahe Xu, and Moshe Zukerman. A base station sleeping strategy in heterogeneous cellular networks based on user traffic prediction. *IEEE Transactions on Green Communications and Networking*, 8(1):134–149, 2023.
- [14] Preetjot Kaur, Roopali Garg, and Vinay Kukreja. Energy-efficiency schemes for base stations in 5g heterogeneous networks: a systematic literature review. *Telecommunication Systems*, 84(1):115–151, 2023.
- [15] Lin Peng, Aihua Zhou, Junfeng Qiao, Qinghe Sun, Zhonghao Qian, Min Xu, and Sen Pan. Day-ahead coordinated scheduling of distribution networks considering 5g base stations and electric vehicles. *Electronics*, 14(19):3940, 2025.
- [16] S Varalakshmi, R Niruban, Janjhyam Venkata Naga Ramesh, and M Kameswara Rao. An integrated noma-based framework with adaptive network topology and traffic flow control. *Wireless Personal Communications*, 145(1):303–334, 2025.
- [17] Gowtham Raj Veeraswamy Premkumar and Bryan Van Scoy. Optimal positioning of unmanned aerial vehicle (uav) base stations using mixed-integer linear programming. *Drones*, 9(1):44, 2025.

- [18] Yu Sun, Haibo Zhou, Kai Yu, Yunting Xu, Bo Qian, and Lin X Cai. Flexible base station sleeping and resource allocation for green uplink fully-decoupled ran. *IEEE Transactions on Wireless Communications*, 2025.
- [19] Qi Du, Shuqin Mo, Yanan Wang, Shusheng Wang, Tuanfa Qin, Binxin Zhu, and Hanbo Zheng. A meta-heuristic mppt algorithm based photovoltaic storage dc microgrid system applied to 5g base station energy supply. *Renewable Energy*, page 125364, 2026.
- [20] Nilakshee Rajule, Mithra Venkatesan, Radhika Menon, Anju Kulkarni, and Rakeshree Raut. Energy-efficient adaptive sleep mode control in cellular networks using grasshopper optimization and traffic-aware qos management. *Cluster Computing*, 29(3):153, 2026.
- [21] Anjana Koyalil and Sivacoumar Rajalingam. Nature-inspired metaheuristic strategies in energy-efficient wireless sensor networks for enhancing network lifetime: A review. *Wireless Personal Communications*, pages 1–31, 2026.
- [22] Maryam Salamatmoghadasi, Metin Ozturk, and Halim Yanikomeroglu. Two-phase cell switching in 6g vhetnets: Sleeping-cell load estimation and renewable-aware switching toward nes. *IEEE Transactions on Cognitive Communications and Networking*, 12:7034–7047, 2026.
- [23] Renli Zhu, Daosen Zhai, Huan Li, Ruonan Zhang, Lei Liu, Zhaohui Yang, and Victor CM Leung. Two-time scale energy-saving scheme with base station sleeping, symbol shutdown, and power allocation. *IEEE Transactions on Green Communications and Networking*, 2025.
- [24] Muhammad Nawaz Khan, Sokjoon Lee, Tariq Hussain, Razaz Waheeb Attar, Mohsin Shah, and Amal Hassan Alhazmi. Dynamic and adaptive scheduling of cognitive sensors for collaborative target tracking in energy-efficient iot environments. *Journal of Network and Computer Applications*, 248:104400, 2026.
- [25] Muhammad Nawaz Khan, Mohsin Shah, Toqeer Mahmood, Rehan Tariq Chohan, and Arshad Khan. Hybrid approach for evaluating dynamic scheduling mechanisms in iot cognitive sensors for enhanced network performance. *Scientific Reports*, 16(1):276, 2025.
- [26] Yazan M Allawi, Alaelddin FY Mohammed, Eman M Moneer, and Lamia O Widaa. Delay-aware sleep synchronization for sustainable 6g-pon broadband access. *Electronics*, 14(16):3229, 2025.
- [27] Ashu Taneja, Nitin Saluja, Neeti Taneja, Ali Alqahtani, MA Elmagzoub, Asadullah Shaikh, and Deepika Koundal. Power optimization model for energy sustainability in 6g wireless networks. *Sustainability*, 14(12):7310, 2022.
- [28] Muhammad Nawaz Khan, Inam Ullah, Ali Kashif Bashir, Mahmoud Ahmad Al-Khasawneh, Ali Arishi, Norah Saleh Alghamdi, and Sokjoon Lee. Reinforcement learning-based dynamic power management for energy optimization in iot-enabled consumer electronics. *IEEE Transactions on Consumer Electronics*, 71(3):8103–8114, 2025.

- [29] Muhammad Nawaz Khan, Sokjoon Lee, and Mohsin Shah. Adaptive scheduling in cognitive iot sensors for optimizing network performance using reinforcement learning. *Applied sciences*, 15(10):5573, 2025.
- [30] Krishnamoorthy Suresh, Raju Kannadasan, Stanley Vinson Joshua, Thangaraj Rajasekaran, Mohammed H Alsharif, Peerapong Uthansakul, and Monthippa Uthansakul. Sustainable resource allocation and base station optimization using hybrid deep learning models in 6g wireless networks. *Sustainability*, 16(17):7253, 2024.
- [31] Yong Tang, Xianjun Deng, Lingzhi Yi, Yunzhi Xia, Laurence T Yang, and Xiao Tang. Collaborative intelligent confident information coverage node sleep scheduling for 6g-empowered green iot. *IEEE Transactions on Green Communications and Networking*, 7(2):1066–1077, 2022.
- [32] Abdulhalim Fayad and Tibor Cinkler. Energy-efficient joint user and power allocation in 5g millimeter wave networks: A genetic algorithm-based approach. *IEEE Access*, 12:20019–20030, 2024.
- [33] Shailendra Pratap Singh, Naweem Kumar, Akansha Singh, Krishna Kant Singh, Sameh S Askar, and Mohamed Abouhawwash. Energy efficient hybrid evolutionary algorithm for internet of everything (ioe)-enabled 6g. *IEEE Access*, 12:63839–63852, 2024.
- [34] Shi Dong, Huadong Su, Yuanjun Xia, Fei Zhu, Xinrong Hu, and Bangchao Wang. A comprehensive survey on authentication and attack detection schemes that threaten it in vehicular ad-hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):13573–13602, 2023.
- [35] Xiang Wu, Baowen Zou, Chuanchuan Lu, Lili Wang, Yongting Zhang, and Huanhuan Wang. Dynamic security computing framework with zero trust based on privacy domain prevention and control theory. *IEEE Journal on Selected Areas in Communications*, 43(6):2266–2278, 2025.
- [36] Liming Yang, Honglin Hu, Ting Zhou, and Tianheng Xu. Joint coded caching and bs sleeping strategy to reduce energy consumption in 6g edge networks. *Internet of Things*, 24:100915, 2023.