Energy-Efficient Routing in Software-Defined Wireless Sensor Networks Using Genetic Algorithms

Khaled Soltani[1], Fouzi Semchedine[2], Ali Moussaoui[3]
[1]University of Ferhat Abbas Sétif 1, Sétif, Algeria
[2]University of Setif 2, Setif, 19000, Algeria
[3]Department of Computer Science, Intelligent Systems and Cognitive Computing (ISCC) Laboratory, University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
E-mail: soltani-khaled@univ-eloued.dz, fouzi.semcheddine@univ-setif.dz, ali.moussaoui@univ-bba.dz

Keywords: Wireless sensor networks, software-defined wireless sensor networks, genetic algorithms, energy-efficient, flow table

Received: April 05, 2026

Energy-efficient routing remains a major challenge in Software-Defined Wireless Sensor Networks (SD-WSNs), where existing approaches typically optimize routing paths without explicitly integrating routing decisions into SDN flow table generation. This paper proposes a multi-objective Genetic Algorithm (GA)-based framework that formulates routing as a flow table optimization problem executed at the SDN controller. The proposed framework evolves network-wide next-hop assignments using feasibility-preserving crossover and mutation operators combined with tournament selection and elitist replacement. A normalized multi-objective fitness function jointly optimizes energy consumption, end-to-end delay, packet delivery ratio (PDR), and hop count using Analytical Hierarchy Process (AHP)-derived weights. The proposed approach is evaluated through extensive OMNeT++ simulations using IEEE 802.15.4 MAC modeling across network sizes ranging from 200 to 2000 sensor nodes. Experimental results over 30 independent runs demonstrate that the proposed framework significantly outperforms shortest-path routing, LEACH, and GA-energy baselines in terms of network lifetime, delivery reliability, throughput, and latency. In particular, the framework reduces dead-node progression by up to 50.5%, improves packet delivery ratio by 13%, increases throughput by 50%, and decreases end-to-end delay by 40% compared with shortest-path routing. Scalability analysis further shows stable performance under large-scale deployment conditions, retaining 76% of peak energy efficiency at 2000 nodes. The results indicate that GA-driven flow table optimization provides a scalable, transparent, and reproducible alternative to reinforcement learning-based routing approaches for energy-aware SD-WSNs.

Povzetek: Članek predlaga večciljni genetski okvir za SD-WSN, ki usmerjanje neposredno oblikuje kot optimizacijo SDN pretočnih tabel ter s tem hkrati izboljša energijsko učinkovitost, zanesljivost dostave, zakasnitev in razširljivost omrežja.

1 Introduction

Wireless Sensor Networks (WSNs) play a fundamental role in distributed sensing applications such as environmental monitoring, industrial automation, healthcare, and smart infrastructure [1, 2]. In these networks, resource-constrained sensor nodes collaboratively transmit sensed data toward sink nodes, making energy efficiency a critical factor for maintaining network lifetime, connectivity, and communication reliability [2, 3]. Consequently, energy-aware routing remains one of the central research challenges in WSN design.

The integration of Software-Defined Networking (SDN) into WSNs introduces centralized network control, global topology visibility, dynamic route reconfiguration, traffic scheduling, and programmable resource management [5, 6, 7]. By decoupling the control plane from the data plane, Software-Defined Wireless Sensor Networks (SD-WSNs) provide greater flexibility and adaptability compared with traditional distributed architectures. However, SD-WSNs also introduce complex multi-objective optimization challenges, where routing decisions must simultaneously satisfy energy efficiency, latency, reliability, and scalability requirements.

Existing routing approaches exhibit important limitations. Traditional shortest-path routing tends to concentrate traffic around sink-adjacent nodes, leading to rapid energy depletion and energy-hole formation [27, 28]. Clustering-based approaches such as LEACH improve load distribution but remain limited by static clustering behavior and insufficient adaptability to dynamic network conditions [29, 30]. More recently, reinforcement learning (RL)-based methods have demonstrated adaptive routing capabilities, but they often suffer from high training overhead, convergence instability, and limited interpretability under dynamic topologies [23, 20]. Furthermore, most existing approaches optimize abstract routing paths rather than directly optimizing SDN flow table generation, and few simultaneously consider energy consumption, end-to-end delay, packet delivery ratio (PDR), and hop count within a unified optimization framework [11, 5].

Genetic Algorithms (GAs) provide a suitable alternative for addressing these limitations. Unlike RL-based approaches, GAs perform population-based global search using explicit and interpretable fitness functions. When executed at the SDN controller, GA-based optimization enables dynamic generation of deployable flow table configurations without imposing computational overhead on sensor nodes. This centralized optimization strategy enables transparent, reproducible, and multi-objective routing optimization while mitigating energy-hole effects and improving network-wide load balancing [22, 21].

In this work, routing is formulated as a *flow table optimization problem* executed at the SDN controller. The proposed framework evolves complete network-wide next-hop assignments into directly installable forwarding configurations using a multi-objective GA. A normalized fitness function jointly optimizes energy consumption, end-to-end delay, packet delivery ratio, and hop count, while objective weights are derived using the Analytical Hierarchy Process (AHP) and validated through sensitivity analysis.

The main contributions of this work are summarized as follows:

1. A GA-based SDN flow table optimization framework that encodes complete network-wide next-hop assignments as chromosomes, enabling globally optimized and directly deployable forwarding configurations.

2. A normalized multi-objective fitness function that jointly optimizes energy consumption, end-to-end delay, packet delivery ratio, and hop count using AHP-derived weighting factors validated through sensitivity analysis.

3. An extensive OMNeT++-based performance evaluation across network sizes ranging from 200 to 2000 sensor nodes, demonstrating significant improvements in energy efficiency, network lifetime, reliability, throughput, and latency compared with shortest-path, LEACH, and GA-energy routing approaches.

The proposed framework is evaluated through the following research questions:

– **RQ1:** To what extent does the proposed framework improve energy efficiency and network lifetime?

– **RQ2:** How does the framework affect packet delivery ratio and end-to-end delay under varying network scales and traffic conditions?

– **RQ3:** Does the proposed optimization framework maintain scalability and convergence stability in large-scale SD-WSN deployments?

All experimental results are reported over 30 independent simulation runs with 95% confidence intervals, and statistical significance is validated using paired t-tests. The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the system model and problem formulation. Section 4 details the proposed GA-based optimization framework. Section 5 describes the experimental setup, while Section 6 discusses the evaluation results. Finally, Section 7 concludes the paper and outlines future research directions.

2 Related work

The emergence of Software-Defined Networking (SDN) has introduced new opportunities for improving Wireless Sensor Networks (WSNs) through centralized and programmable network management [5]. By decoupling the control plane from the data plane, SDN enables global topology awareness, dynamic routing control, and flexible resource allocation, capabilities that are difficult to achieve in conventional distributed WSN architectures. Research in this domain has evolved through three main directions: (i) early studies establishing SDN-based architectures for WSNs [8, 9], (ii) optimization approaches targeting energy efficiency, routing, and QoS under SDN control, and (iii) recent artificial intelligence and machine learning-based methods enabling adaptive SDN management. This section reviews the most relevant contributions in these areas, identifies their limitations, and positions the proposed work within the current state of the art.

2.1 SDN architectures and control frameworks for WSNs

Applying SDN principles to resource-constrained WSNs introduces important architectural challenges. Traditional SDN frameworks typically assume reliable high-bandwidth communication links and computationally capable devices, assumptions that do not hold in most sensor network environments. Consequently, early research primarily focused on establishing feasible SDN control architectures before addressing optimization objectives.

Bera et al. [9] presented a comprehensive survey of SDN-based technologies for IoT systems, highlighting how centralized programmability can improve scalability, remote management, and cross-layer coordination compared with conventional distributed protocols. Their study also emphasized that energy-aware control and network lifetime optimization remained insufficiently addressed within existing SDN-based architectures, identifying these issues as important open research challenges.

Shabbir et al. [10] addressed part of this limitation by integrating SDN with the Routing Protocol for Low-Power and Lossy Networks (RPL). Their framework demonstrated that centralized SDN control can improve scalability and reduce congestion while preserving the self-healing capabilities of RPL under node failures. However, the proposed architecture relied on relatively static routing decisions and did not dynamically adapt SDN flow table entries according to residual energy variations, limiting its effectiveness for long-term energy-aware operation [12, 13].

Suresh et al. [11] further extended SDN-based architectures toward Software-Defined Wireless Mesh Networks and introduced the concept of ML-SDWSN, which integrates machine learning techniques into SD-WSN management. Their survey systematically analyzed the limitations of existing SDWSN-IoT systems and identified flow table management as a relatively underexplored research direction compared with conventional routing optimization. In particular, they highlighted that most existing approaches optimize routing paths abstractly without directly addressing how controller decisions are translated into deployable SDN forwarding rules. This observation strongly motivates the flow table optimization framework proposed in the present work.

2.2 Energy-efficient routing and QoS optimization in SD-WSNs

Energy-efficient routing has become one of the primary research objectives in SD-WSNs, with existing approaches ranging from single-objective optimization strategies based on residual energy or hop count to more advanced multi-objective formulations incorporating delay, reliability, and QoS constraints.

Zhao et al. [16] combined Hopfield neural networks with a Multi-choice Knapsack optimization model to improve energy efficiency and throughput in WSN routing. Although their approach demonstrated promising performance gains, the reliance on per-node computational processing conflicts with the resource limitations of sensor nodes and reduces adaptability under dynamic network conditions.

Huang et al. [15] proposed an SDN-enabled reinforcement learning (RL) framework for load balancing and redundancy filtering, achieving reduced energy consumption while maintaining QoS requirements. However, RL-based optimization requires continuous interaction with the environment and often suffers from slow convergence, potentially exposing the network to unstable or suboptimal routing decisions during training phases. This limitation motivates the use of controller-side evolutionary optimization strategies such as Genetic Algorithms (GAs), which avoid continuous online retraining.

Rostami et al. [14] further demonstrated that proactive and centralized flow optimization can outperform reactive load-balancing approaches in SDN-based IoT systems, supporting the effectiveness of controller-driven optimization mechanisms.

More recently, Alsaeedi et al. [17] employed a PPO-based deep reinforcement learning model to reduce controller overhead and control-plane traffic in SD-WSNs. While their approach achieved promising adaptive behavior, the learned reward mechanism remains difficult to interpret and requires extensive training data and parameter tuning. In contrast, the proposed GA-based framework relies on an explicit and interpretable multi-objective fitness function derived using the Analytical Hierarchy Process (AHP), improving transparency, reproducibility, and objective-level control.

Complementing routing-level optimization, Khudor et al. [25] introduced a KF-SVM hybrid model for cluster-head data aggregation, reporting up to 60% improvement in network lifetime through redundancy reduction. Nevertheless, such localized optimization techniques operate primarily at the data aggregation level and cannot exploit global network visibility. By contrast, the centralized SDN controller in the proposed framework maintains network-wide awareness, enabling coordinated routing decisions that jointly consider energy consumption, traffic distribution, reliability, and latency across the entire network.

2.3 Artificial intelligence integration in SD-WSN control

The growing complexity of SD-WSN management has accelerated the integration of artificial intelligence (AI) techniques into SDN controllers to support adaptive and intelligent routing decisions. Recent studies have explored deep learning, reinforcement learning, and hybrid AI models to improve energy efficiency, scalability, and QoS performance in dynamic network environments.

Alqaraghuli et al. [19] proposed a 3D-CNN-based deep reinforcement learning (DRL) framework that achieved approximately 18% improvement in network lifetime compared with conventional 2D-CNN approaches. Despite these gains, CNN-based DRL methods typically require large training datasets and may experience performance degradation under rapidly changing network topologies.

Younus et al. [20] introduced an RL-based routing scheme combining energy-awareness and QoS-oriented reward functions. Their approach reported network lifetime improvements ranging from 8% to 33% and packet delivery ratio (PDR) gains between 2% and 24%. However, RL-based optimization remains sensitive to reward-function design, environmental variability, and retraining overhead, limiting reproducibility and long-term stability in highly dynamic SD-WSN environments.

Tooba et al. [18] integrated an LSTM-based intrusion detection mechanism into the SDN control plane, demonstrating that intelligent learning models can be deployed at the controller level without introducing excessive communication latency. Their work indirectly supports the feasibility of centralized intelligent optimization within SD-WSN architectures, including controller-side GA-based routing optimization as proposed in this study.

2.4 Genetic algorithms for energy-efficient routing in SD-WSNs

Genetic Algorithms (GAs) represent an important class of AI-based optimization techniques for SD-WSNs due to their ability to perform population-based global search using explicit fitness-driven optimization. Unlike reinforcement learning (RL) and deep reinforcement learning (DRL) approaches, which rely on continuous interaction with the environment, GAs directly evaluate candidate solutions through predefined objective functions, enabling more interpretable and controllable multi-objective optimization processes [21].

Wang et al. [22] proposed a GA-based Energy-efficient Clustering and Routing (GECR) approach that jointly optimizes clustering and routing decisions to improve energy efficiency and load balancing. Although their method achieved promising energy-related improvements, its single-objective fitness formulation does not explicitly consider delay, reliability, or hop count, potentially producing routing configurations that are energy-efficient but unsuitable for latency-sensitive or reliability-aware applications.

Similarly, Altmemi et al. [26] employed a Dragonfly Algorithm to optimize energy consumption, path length, and load balancing in SD-WSNs. However, their approach optimizes abstract routing paths rather than directly generating SDN flow table entries. In addition, the optimization process relies on behavioral search coefficients rather than explicit objective-level control, which may limit interpretability and reproducibility across deployment scenarios.

In contrast to these approaches, the proposed framework formulates routing as an SDN flow table optimization problem executed at the controller level. Each chromosome directly encodes deployable next-hop forwarding assignments, enabling centralized and dynamically reconfigurable routing optimization.

The proposed multi-objective fitness function is defined as follows:

$$F(\mathcal{F}) = \frac{\alpha}{\bar{E}^{\text{norm}}(\mathcal{F}) + \varepsilon} + \frac{\beta}{\bar{D}^{\text{norm}}(\mathcal{F}) + \varepsilon} + \gamma\, \overline{\text{PDR}}^{\text{norm}}(\mathcal{F}) - \delta\, \bar{H}^{\text{norm}}(\mathcal{F})$$

where $\bar{E}^{\text{norm}}$, $\bar{D}^{\text{norm}}$, $\overline{\text{PDR}}^{\text{norm}}$, and $\bar{H}^{\text{norm}}$ denote the normalized energy consumption, end-to-end delay, packet delivery ratio, and hop count, respectively. The weighting factors $(\alpha, \beta, \gamma, \delta)$ are derived using the Analytical Hierarchy Process (AHP), allowing explicit control over the relative importance of each routing objective.

The fitness evaluation is performed over all source-to-sink communication paths induced by the flow table configuration $\mathcal{F}$. This formulation enables coordinated optimization of energy efficiency, reliability, latency, and path efficiency within a unified and reproducible optimization framework. Consequently, the proposed approach addresses an important limitation in existing SD-WSN routing methods, where optimization is commonly performed on abstract routing paths without directly considering deployable SDN forwarding rules.

2.5 Comparative analysis and research gap

Table 1 presents a comparative analysis of representative SD-WSN optimization approaches across multiple performance dimensions, including energy efficiency (EE), network lifetime, fault tolerance, QoS support, AI/ML techniques, scalability, and SDN flow table optimization capability. Performance ratings are categorized as High (H), Moderate (M), Low (L), Not Evaluated (N/E), or Indirectly Addressed (Ind.) according to the evaluation scope reported in the corresponding studies.

Several important observations can be derived from Table 1. First, AI-based routing approaches, particularly RL- and DRL-based methods, generally demonstrate improved energy efficiency and network lifetime compared with conventional heuristic techniques. This trend highlights the importance of adaptive optimization mechanisms in dynamic SD-WSN environments.

Second, fault tolerance remains relatively underexplored across most existing studies. The majority of current approaches primarily focus on energy efficiency and QoS optimization under normal operating conditions without explicitly addressing failure-aware routing or resilience modeling.

Third, and most importantly, most existing methods optimize routing paths, clustering behavior, or controller policies at an abstract level without directly optimizing deployable SDN flow table configurations. Although Alsaeedi et al. [17] partially address controller-side optimization using PPO-based DRL, their work primarily focuses on reducing controller overhead rather than explicitly formulating flow table generation as the optimization objective.

This limitation motivates the proposed framework, which formulates routing as a direct SDN flow table optimization problem. In contrast to prior approaches, the proposed method jointly optimizes energy consumption, end-to-end delay, packet delivery ratio, and hop count using an explicitly weighted multi-objective GA executed at the SDN controller. The use of AHP-derived weighting factors further improves interpretability and objective-level control while preserving centralized network-wide optimization.

Among the reviewed approaches, the studies by Alsaeedi et al. [17] and Wang et al. [22] are the closest to the present work. However, Alsaeedi et al. rely on implicitly learned reward functions without explicit trade-off modeling for delay and reliability, whereas Wang et al. employ a single-objective GA formulation without SDN flow table integration. By comparison, the proposed framework combines multi-objective evolutionary optimization with directly deployable SDN forwarding-rule generation, addressing an important gap in existing SD-WSN routing research.

Table 1: Comparative analysis of related SD-WSN studies. Ratings: H = High, M = Moderate, L = Low, N/E = Not Evaluated, Ind. = Indirectly addressed. **Flow Table**: whether the study directly optimizes SDN flow table generation (rather than abstract routing paths).

Reference (Year)	EE	Lifetime	Fault Tol.	QoS	AI/ML	Scalab.	Flow Table	Main Contribution
Bera et al., 2017 [9]	M	N/E	L	M	No	M	No	Survey on SDN-IoT architectures and open challenges
Rostami et al., 2024 [14]	Ind.	N/E	L	H	No	H	No	Load-balancing strategies for QoS in SD-IoT
Tooba et al., 2021 [18]	L	N/E	M	M	LSTM	M	No	LSTM-based intrusion detection for SDN-IoT security
Shabbir et al., 2019 [10]	M	M	M	M	No	H	No	SDN-RPL integration for scalability and congestion control
Huang et al., 2015 [15]	H	H	M	H	RL	H	No	RL-based adaptive routing with redundancy filtering
Zhao et al., 2018 [16]	H	M	L	M	NN	M	No	Hierarchical routing with Hopfield neural networks
Alsaeedi et al., 2024 [17]	H	H	M	M	DRL (PPO)	H	Partial	PPO-based flow optimization and controller overhead reduction
Suresh et al., 2024 [11]	M	M	L	M	ML	M	No	Survey of SDWSN-IoT systems and ML-SDWSN paradigm
Alqaraghuli et al., 2024 [19]	H	H	M	H	DRL (CNN)	H	No	3D-CNN DRL routing for extended network lifetime
Younus et al., 2021 [20]	H	H	M	H	RL	H	No	QoS-aware RL routing with energy-lifetime improvement
Wang et al., 2018 [22]	H	H	L	M	GA	M	No	Single-objective GA clustering and routing (GECR)
Li et al., 2003 [24]	M	M	H	M	No	M	No	Adaptive self-organizing routing with potential fields
Khudor et al., 2022 [25]	H	H	L	M	SVM+KF	M	No	Hybrid KF-SVM data aggregation for redundancy reduction
Altmemi et al., 2026 [26]	H	H	M	M	DA (SI)	H	No	Dragonfly Algorithm routing for energy efficiency and lifetime
Proposed	**H**	**H**	**M**	**H**	**GA (multi-obj.)**	**H**	**Yes**	**Multi-objective GA for SDN flow table generation**

Summary

Existing SD-WSN approaches, including SDN-based routing methods, RL/DRL techniques, and conventional GA-based optimization models, primarily focus on abstract routing paths or isolated performance metrics without directly addressing SDN flow table generation. In addition, many existing AI-driven approaches rely on implicitly learned optimization policies that may limit interpretability and reproducibility under dynamic network conditions.

In contrast, the proposed framework formulates routing as a multi-objective SDN flow table optimization problem executed at the controller level. The framework jointly considers energy consumption, end-to-end delay, packet delivery ratio, and hop count using an explicitly weighted GA-based fitness function derived through the Analytical Hierarchy Process (AHP). This formulation provides a centralized, interpretable, and computationally efficient alternative for energy-aware routing in dynamic SD-WSN environments.

3 System model and problem formulation

This section presents the formal system model underlying the proposed SD-WSN optimization framework. Section 3.1 describes the architecture of the SD-WSN and the interactions among its constituent layers, while Section 3.2 formulates the flow table optimization problem, including the decision variables, objective function, and feasibility constraints.

3.1 SD-WSN architecture

A Software-Defined Wireless Sensor Network (SD-WSN) separates the control plane from the data plane through a hierarchical three-layer architecture composed of the *Application Plane*, *Control Plane*, and *Data Plane*. Figure 1 illustrates the overall architecture and the communication interfaces connecting these layers.

Application Plane. The application plane defines high-level network policies and service objectives, including energy-aware routing, quality-of-service (QoS) management, and security monitoring. These requirements are communicated to the SDN controller through the Northbound API, where they are translated into routing and resource-management objectives.

Control Plane. The control plane consists of a centralized SDN controller responsible for maintaining a global view of the network topology, residual node energy, link conditions, and traffic distribution. Based on this global network state, the controller generates and installs forwarding rules through the Southbound API. In the proposed framework, the controller hosts the GA-based optimization module, which dynamically computes flow table configurations according to current network conditions while avoiding computational overhead on sensor nodes.

Data Plane. The data plane comprises distributed sensor nodes responsible for environmental sensing and packet forwarding. Each node operates according to controller-installed forwarding rules stored in local flow tables, eliminating the need for distributed routing computation at the node level.

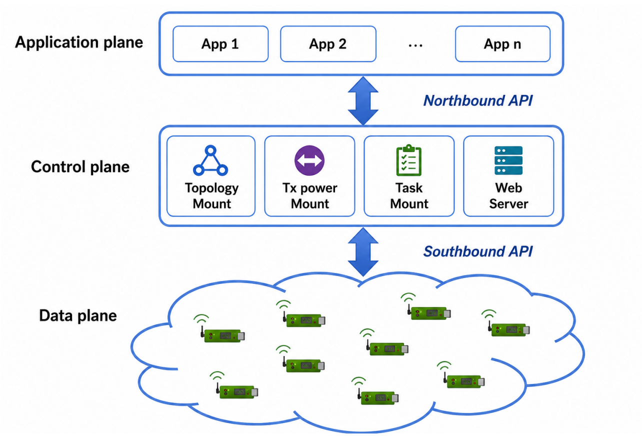


Figure 1: Three-layer SD-WSN architecture showing the interaction between the application, control, and data planes.

The separation between the control and data planes enables centralized and globally optimized routing decisions while preserving the limited computational and energy resources of sensor devices.

Figure 2 further illustrates the integration of the GA-based optimization module within the SDN controller. The optimization module continuously processes global network-state information and generates deployable flow table configurations for the sensor nodes. Since all evolutionary optimization operations are executed centrally at the controller level, the proposed architecture preserves node-level energy resources and computational simplicity within the data plane.

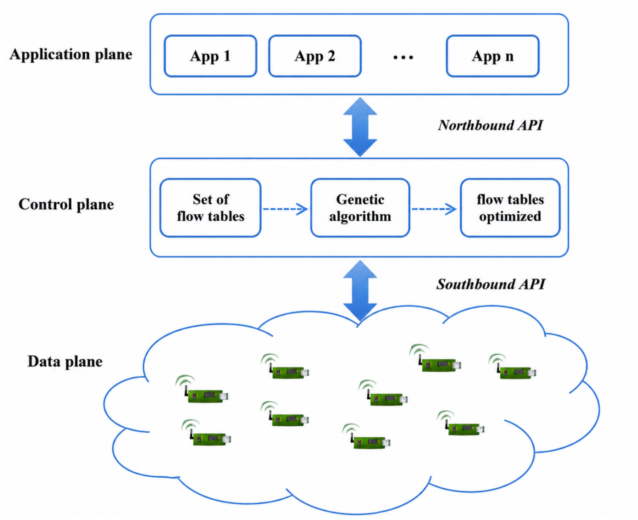


Figure 2: Integration of the GA-based optimization module within the SD-WSN control plane.

3.2 Problem formulation

3.2.1 Network model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the SD-WSN graph, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of N sensor nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of wireless links. A directed link $(v_i, v_j) \in \mathcal{E}$ exists if and only if v_j lies within the transmission range of v_i. Each node v_i is characterized by its residual energy $E_i \geq 0$, and each link (v_i, v_j) is characterized by its packet loss probability $\ell_{ij} \in [0, 1]$ and transmission distance d_{ij}. Let $s \in \mathcal{V}$ denote the source node and $t \in \mathcal{V}$ denote the sink node.

3.2.2 Decision variables: flow table entries

The SDN controller maintains a flow table $\mathcal{F}$ where each entry f_i assigns a next-hop to node v_i:

$$f_i = \langle \text{match}_i,\ \text{nxt}(i),\ \text{priority}_i,\ \tau_i,\ E_i \rangle, \quad \forall v_i \in \mathcal{V} \setminus \{t\} \tag{1}$$

The **decision variable** is the next-hop assignment $\text{nxt}(i) \in \mathcal{N}^+(v_i)$, where $\mathcal{N}^+(v_i)$ denotes the set of neighbors of v_i within transmission range R with positive residual energy. The GA evolves the complete mapping:

$$\mathbf{chr} = [\text{nxt}(v_1),\ \text{nxt}(v_2),\ \ldots,\ \text{nxt}(v_{N-1})] \tag{2}$$

where each gene directly corresponds to the Next-Hop field of a specific flow table entry, and a chromosome of length $N-1$ encodes a complete, network-wide, installable flow table.

3.2.3 Optimization objective

The SDN controller seeks a flow table configuration $\mathcal{F}^*$ that maximizes the normalized multi-objective fitness function:

$$\mathcal{F}^* = \arg\max_{\mathcal{F}\in\Omega} F(\mathcal{F}) \tag{3}$$

where Ω denotes the feasible flow table space and F is defined as:

$$F = \alpha \cdot \frac{1}{E_{\text{avg}}^{\text{norm}} + \varepsilon} + \beta \cdot \frac{1}{D^{\text{norm}} + \varepsilon} + \gamma \cdot \text{PDR}^{\text{norm}} - \delta \cdot H^{\text{norm}} \tag{4}$$

subject to:

$$\alpha+\beta+\gamma+\delta = 1, \quad \alpha = 0.4, \quad \beta = 0.3, \quad \gamma = 0.2, \quad \delta = 0.1 \tag{5}$$

The four terms in Equation 4 respectively penalize average path energy consumption ($E_{\text{avg}}^{\text{norm}}$) and end-to-end delay ($D^{\text{norm}}$), reward packet delivery reliability (PDR^{norm}), and penalize excessive hop count (H^{norm}). The constant $\varepsilon = 10^{-6}$ prevents division by zero. Weight derivation through the Analytical Hierarchy Process (AHP) and validation through sensitivity analysis are presented in Section 4.

3.2.4 Feasibility constraints

A flow table configuration $\mathcal{F}$ is feasible ($\mathcal{F} \in \Omega$) if and only if the encoded routing path C satisfies all of the following constraints:

1. **Connectivity:** Every consecutive node pair must share a valid wireless link:

$$(n_i, n_{i+1}) \in \mathcal{E}, \quad \forall\, i \in \{1, \ldots, k-1\} \tag{6}$$

2. **Loop-freedom:** No node may appear more than once in the path:

$$n_i \neq n_j, \quad \forall\, i \neq j \tag{7}$$

3. **Energy availability:** Every forwarding node must have positive residual energy:

$$E_i > 0, \quad \forall\, n_i \in C \setminus \{t\} \tag{8}$$

4. **Reachability:** The path must originate at the source and terminate at the sink:

$$n_1 = s, \quad n_k = t \tag{9}$$

5. **Transmission range:** Each hop must not exceed the physical transmission range R:

$$d_{ij} \leq R, \quad \forall\, (n_i, n_{i+1}) \in C \tag{10}$$

These constraints are enforced directly by the feasibility-preserving crossover and mutation operators described in Section 4, ensuring that every candidate solution evaluated by the GA corresponds to a valid, directly installable flow table configuration. Infeasible chromosomes are never introduced into the population, eliminating the need for penalty-based constraint handling and guaranteeing that all evaluated fitness values correspond to deployable routing rules.

3.2.5 Complexity and re-optimization trigger

The feasible solution space $|\Omega|$ grows combinatorially with network size. For a network of N nodes with average node degree $\bar{d}$ and network diameter D, the number of loop-free paths from source to sink is bounded by $O(\bar{d}^D)$, making exhaustive search intractable beyond small networks and motivating the use of population-based evolutionary search.

Re-optimization is triggered at the SDN controller whenever any of the following conditions are detected via the Southbound API:

– a node's residual energy falls below threshold E_{min};

– a link failure is reported; or

– end-to-end delay exceeds the latency bound D_{max}.

Between trigger events, the installed flow table remains active, avoiding unnecessary recomputation and keeping controller overhead proportional to actual network dynamics.

4 Flow table optimization using genetic algorithms

This section formally specifies the GA-based flow table optimization framework introduced in Section 3, detailing the chromosome encoding, fitness evaluation, genetic operators, and the complete optimization procedure.

4.1 Flow table architecture and decision space

The flow table acts as the decision engine for packet processing at sensor nodes. Table 2 defines each flow table field and its role.

The flow table configuration is:

$$\begin{aligned} &\mathcal{F} = \{f_i \mid i \in \mathcal{V} \setminus \{t\}\}, \\ &f_i = \langle \text{match}_i,\ \text{nxt}(i),\ \text{priority}_i,\ \tau_i,\ E_i \rangle \end{aligned} \tag{11}$$

where $\text{nxt}(i) \in \mathcal{N}^+(i)$ and t is the sink. The GA evolves $\text{nxt} : \mathcal{V} \setminus t \to \mathcal{V}$ subject to feasibility constraints. Each configuration induces paths from sources to the sink, enabling global fitness evaluation.

Table 2: Flow table entry structure in SD-WSN controllers

Field	Function
Match Fields	Packet identification by header attributes: source, destination, protocol, port.
Action	Forward to next-hop n_j, drop, or redirect to controller. In this work, only forwarding actions are considered.
Priority	Resolves conflicts for multiple matches; higher-priority entries override lower-priority.
Timeout	Controls rule expiration via idle/hard timeout; refreshed upon re-optimization.
Energy Metric	Residual energy E_i of forwarding node, guiding energy-aware optimization.
Next-Hop	Forwarding target $n_j \in \mathcal{N}(n_i)$; GA-evolved decision variable.

4.2 Multi-objective fitness function and weights

Metrics are min-max normalized:

$$X^{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}} + \varepsilon}, \quad \varepsilon = 10^{-6} \tag{12}$$

The GA maximizes:

$$F(\mathcal{F}) = \frac{\alpha}{\bar{E}^{\text{norm}}(\mathcal{F}) + \varepsilon} + \frac{\beta}{\bar{D}^{\text{norm}}(\mathcal{F}) + \varepsilon} + \gamma\, \overline{\text{PDR}}^{\text{norm}}(\mathcal{F}) - \delta\, \bar{H}^{\text{norm}}(\mathcal{F}) \tag{13}$$

with $\alpha + \beta + \gamma + \delta = 1$, set to $(0.4, 0.3, 0.2, 0.1)$.

AHP-derived weights are shown in Table 3 with $CR = 0.017 < 0.1$ ensuring consistency.

Table 3: AHP pairwise comparison matrix for fitness weights. Scale: 1 = equal importance, 3 = moderate, 5 = strong, 7 = very strong, 9 = extreme importance.

Criterion	Energy	Delay	PDR	Hop Count
Energy	1	2	3	5
Delay	1/2	1	2	4
PDR	1/3	1/2	1	2
Hop Count	1/5	1/4	1/2	1

Sensitivity analysis. Weight vectors in the neighbourhood of the AHP solution are evaluated through simulation to verify stability. Results in Section 5.8 confirm that small perturbations to any individual weight produce no disproportionate performance degradation, validating the AHP-derived vector as a robust and reproducible operating point for general-purpose SD-WSN deployment.

4.3 System models

Energy: $E_{\text{tx},i} = E_{\text{elec}}k + E_{\text{amp}}kd_i^{\eta}, \quad E_{\text{rx},i} = E_{\text{elec}}k$

$$\bar{E}(\mathcal{F}) = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \frac{1}{|P_s|}\sum_{n_i\in P_s} (E_{\text{tx},i} + E_{\text{rx},i} + E_{\text{proc},i}) \tag{14}$$

where k is packet size in bits, $d_i = d(n_i, \text{nxt}(i))$ is the transmission distance to the assigned next-hop, $\eta \in [2, 4]$ is the path loss exponent, E_{elec} is the electronics energy per bit, and E_{amp} is the amplifier energy coefficient.

Delay: End-to-end delay comprises transmission, propagation, processing, queuing, and MAC components:

$$D = D_{\text{trans}} + D_{\text{prop}} + D_{\text{proc}} + D_{\text{queue}} + D_{\text{MAC}} \tag{15}$$

For the CBR traffic used in this work (5 packets/second), queuing delay follows the deterministic D/D/1 model:

$$D_{\text{queue}} = \max\left(0, \frac{\lambda}{\mu(\mu - \lambda)}\right), \quad \lambda < \mu \tag{16}$$

where λ is the packet arrival rate and μ the service rate. This provides an exact bound for deterministic traffic, unlike the M/M/1 model which assumes Poisson arrivals. With $\lambda/\mu \leq 0.6$ in all experiments, queuing delay remains below 2.5 ms per node.

PDR: $\text{PDR}(P_s) = \prod_{n_i\in P_s} p_i, \quad \overline{\text{PDR}}(\mathcal{F}) = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \text{PDR}(P_s)$.

Hop Count: $\bar{H}(\mathcal{F}) = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}(|P_s| - 1)$.

All metrics are normalized before substitution into $F(\mathcal{F})$.

4.4 Genetic algorithm design

Chromosome Encoding:

$$\textbf{chr} = [\text{nxt}(v_1), \ldots, \text{nxt}(v_{N-1})] \tag{17}$$

Initialization: Each chromosome is constructed by assigning every node v_i a next-hop drawn uniformly at random from:

$$\mathcal{N}^+(v_i) = \{v_j \in \mathcal{N}(v_i) \mid E_j > 0,\; d_{ij} \leq R,\; \text{hop}(v_j, t) < \text{hop}(v_i, t)\}$$

The crossover operator in Algorithm 1 preserves feasibility by accepting inherited genes only when neighborhood and hop-progress constraints are satisfied. This guarantees loop-free forwarding toward the sink and prevents invalid flow table entries. Unlike conventional crossover methods, the proposed operator embeds feasibility verification

Algorithm 1 Feasibility-Preserving Flow Table Crossover

Require: Parents $\mathbf{chr}_1$, $\mathbf{chr}_2$ of length $N-1$; neighbor sets $\mathcal{N}^+(v_i)$; hop distances $\text{hop}(v,t)$; crossover probability p_c
Ensure: Feasible offspring $\mathbf{off}_1$, $\mathbf{off}_2$

for $i = 1$ **to** $N-1$ **do**
  if $\text{Uniform}(0,1) < p_c$ **then**
    if $\mathbf{chr}_2[i] \in \mathcal{N}^+(v_i)$ **and** $\text{hop}(\mathbf{chr}_2[i],t) < \text{hop}(v_i,t)$ **then**
      $\mathbf{off}_1[i] \leftarrow \mathbf{chr}_2[i]$
    else
      $\mathbf{off}_1[i] \leftarrow \mathbf{chr}_1[i]$
    end if
    if $\mathbf{chr}_1[i] \in \mathcal{N}^+(v_i)$ **and** $\text{hop}(\mathbf{chr}_1[i],t) < \text{hop}(v_i,t)$ **then**
      $\mathbf{off}_2[i] \leftarrow \mathbf{chr}_1[i]$
    else
      $\mathbf{off}_2[i] \leftarrow \mathbf{chr}_2[i]$
    end if
  else
    $\mathbf{off}_1[i] \leftarrow \mathbf{chr}_1[i]$
    $\mathbf{off}_2[i] \leftarrow \mathbf{chr}_2[i]$
  end if
end for
return $\mathbf{off}_1$, $\mathbf{off}_2$

Algorithm 2 Feasibility-Preserving Flow Table Mutation

Require: Chromosome $\mathbf{chr}$, mutation prob. p_m, neighbor sets $\mathcal{N}^+(v_i)$
Ensure: Mutated chromosome $\mathbf{chr}'$

$\mathbf{chr}' \leftarrow \mathbf{chr}$
for $i = 1$ **to** $N-1$ **do**
  if $\text{Uniform}(0,1) < p_m$ **and** $|\mathcal{N}^+(v_i)| > 1$ **then**
    $\mathbf{chr}'[i] \leftarrow \text{SelectUniformRandom}(\mathcal{N}^+(v_i) \setminus \{\mathbf{chr}[i]\})$
  end if
end for
return $\mathbf{chr}'$

directly into the recombination process, ensuring that all offspring remain immediately deployable within the SD-WSN framework.

$$f_i = \Big\langle \text{match : dst} = t, \quad \text{action : fwd to } \mathbf{chr}^*[i], \quad \text{priority : } P_0, \quad \text{timeout : } \tau_0 \Big\rangle, \quad \forall v_i \in \mathcal{V} \setminus \{t\} \tag{18}$$

4.5 Computational complexity and convergence

Fitness evaluation per chromosome: $O(N \cdot |\mathcal{S}|)$. Overall GA: $O(P_{\text{size}} \cdot G \cdot N \cdot |\mathcal{S}|)$. Typical execution < 0.32 s per optimization cycle.

Elitist replacement and mutation ensure retention of the best solution; convergence observed around generation 42 across 30 runs.

Re-optimization is triggered by:

Algorithm 3 GA-Based Flow Table Optimization at SDN Controller

Require: Network graph $\mathcal{G}$, energy vector $\mathbf{E}$, link loss vector $\boldsymbol{\ell}$, population size P_{size}, generations G, crossover probability p_c, mutation probability p_m, tournament size τ, fitness weights $\alpha, \beta, \gamma, \delta$, transmission range R
Ensure: Optimized flow table $\mathcal{F}^*$

$\mathcal{P} \leftarrow \emptyset$
for $i = 1$ **to** P_{size} **do**
  $\mathbf{chr}_i \leftarrow \text{InitFeasibleChromosome}(\mathcal{G}, \mathbf{E}, R)$
  Evaluate $F(\mathbf{chr}_i)$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathbf{chr}_i\}$
end for
$\mathbf{chr}^* \leftarrow \arg\max_{\mathbf{chr} \in \mathcal{P}} F(\mathbf{chr})$
for $g = 1$ **to** G **do**
  $\mathcal{P}' \leftarrow \{\mathbf{chr}^*\}$
  while $|\mathcal{P}'| < P_{\text{size}}$ **do**
    $\mathbf{a}, \mathbf{b} \leftarrow \text{TournamentSelect}(\mathcal{P}, \tau)$
    $\mathbf{o}_1, \mathbf{o}_2 \leftarrow \text{FTCrossover}(\mathbf{a}, \mathbf{b}, p_c)$
    for all $\mathbf{o} \in \{\mathbf{o}_1, \mathbf{o}_2\}$ **do**
      $\mathbf{o} \leftarrow \text{FTMutate}(\mathbf{o}, p_m)$
      Evaluate $F(\mathbf{o})$
      $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathbf{o}\}$
    end for
  end while
  $\mathcal{P} \leftarrow \mathcal{P}'$
  $\mathbf{chr}^* \leftarrow \arg\max_{\mathbf{chr} \in \mathcal{P}} F(\mathbf{chr})$
  if $\Delta F(\mathbf{chr}^*) < 1 \times 10^{-3}$ for 5 consecutive generations **then**
    break
  end if
end for
$\mathcal{F}^* \leftarrow \text{ChromosomeToFlowTable}(\mathbf{chr}^*)$
Install $\mathcal{F}^*$ via Southbound API
return $\mathcal{F}^*$

– Residual energy below E_{min}
– Link failure
– Measured delay exceeding D_{max}

This design enables adaptive, scalable, energy-aware flow table optimization in SD-WSNs.

5 Performance evaluation of GA-based flow table optimization in large-scale SD-WSN

This section presents a comprehensive experimental evaluation of the proposed Genetic Algorithm (GA)-based routing optimization framework for Software-Defined Wireless Sensor Networks (SD-WSNs). We compare our approach against three established baselines: traditional shortest-path routing, LEACH protocol [3], and a GA-based energy-aware routing method [4]. All experiments are conducted over 30 independent simulation runs with different random seeds to ensure statistical reliability, and results are reported as mean $\pm$ standard deviation with 95% confidence intervals. Statistical significance between the proposed approach and each baseline is evaluated using paired t-tests.

5.1 Simulation environment and experimental methodology

All experiments were conducted using the OMNeT++ discrete-event simulation platform [31], selected for its modular architecture, scalability, and established support for wireless network modeling in large-scale WSN studies. The simulation integrates the INET Framework [32, 33], which provides detailed implementations of IEEE 802.15.4 MAC protocols, radio propagation models, queue management, and configurable energy consumption modules enabling accurate modeling of packet collisions, retransmissions, and energy depletion dynamics.

The SDN controller was implemented as a centralized control module within OMNeT++, responsible for global topology awareness, dynamic flow table generation, and route optimization via the proposed multi-objective genetic algorithm (GA). The controller periodically collects network state information, including node energy levels, queue lengths, and link quality metrics, to compute optimal routing decisions. Sensor nodes were modeled with realistic hardware constraints, including limited transmission range, finite battery capacity, and constrained processing capabilities. Energy consumption was computed using a state-based radio model accounting for transmission, reception, idle listening, and processing states. The radio model parameters were calibrated based on the CC2420 transceiver datasheet, a commonly used RF chip in WSN deployments. The simulations employed IEEE 802.15.4 MAC modeling with contention-based medium access and packet retransmission mechanisms to ensure realistic communication behavior. Network traffic consisted of periodic sensing reports forwarded toward a centralized sink through controller-installed flow table entries. The proposed framework dynamically updated routing configurations according to changing network conditions, enabling adaptive traffic distribution and balanced energy consumption. All simulations were repeated over 30 independent runs using different pseudo-random seeds to ensure statistical reliability and reproducibility.

Network sizes of 200, 500, 1000, and 2000 nodes were evaluated to assess scalability, with node density held constant at 0.1 nodes/m^2 by proportionally scaling the deployment area. Traffic generation followed a Constant Bit Rate (CBR) model with uniform packet emission across nodes. Each simulation run lasted for 1000 seconds of simulated time, allowing sufficient convergence of routing protocols and energy depletion patterns. Nodes were deployed using a uniform random spatial distribution, with a static sink positioned at the network center. Table 4 presents the base simulation configuration used in this study. While the default network size is set to 1000 sensor nodes, additional experiments were conducted with 200, 500, and 2000 nodes to evaluate scalability performance. To ensure statistical significance, each experimental scenario was repeated 30 times with different random seeds, and results are reported with 95% confidence intervals.

Table 4: Base simulation parameters for the network configuration. The default scenario considers 1000 sensor nodes, while scalability analysis is performed for 200, 500, 1000, and 2000 nodes.

Parameter	**Value**
Network size	1000 sensor nodes
Deployment area	$100\,\text{m} \times 100\,\text{m}$
Base station location	(50, 50) m
Initial node energy	100 J
Transmission range	20 m
Transmission power	0.6 W
Reception power	0.3 W
Processing power	0.05 W
Idle power	0.02 W
Radio model	Two-ray ground propagation
MAC protocol	IEEE 802.15.4 (CSMA/CA)
Packet size	100 bytes
Data rate	250 kbps
Traffic model	CBR (5 packets/second)
Simulation time	100 rounds (10 seconds each)
Number of runs	30 (independent random seeds)
GA-Specific Parameters	
Population size	20 chromosomes
Number of generations	50
Crossover rate	0.8
Mutation rate	0.1
Selection method	Tournament (size = 3)
Fitness weights	$\alpha = 0.4,\ \beta = 0.3,\ \gamma = 0.2,\ \delta = 0.1$
Chromosome encoding	Next-hop assignment vector

Parameter justification The population size of 20 chromosomes and 50 generations were selected based on preliminary convergence analysis (Section 5.3), which demonstrated fitness stabilization within 40 generations across all tested network sizes. The mutation rate of 0.1 balances exploration and exploitation in accordance with established GA tuning guidelines [34, 35]. The fitness function weights ($\alpha = 0.4, \beta = 0.3, \gamma = 0.2, \delta = 0.1$) reflect a balanced energy-latency configuration appropriate for general purpose WSN applications and were derived through the AHP-based weight selection methodology described in Section 4.

Baseline methods Three routing protocols serve as comparison points. **Shortest Path (SP)** applies Dijkstra-based routing that minimizes hop count without any form of energy awareness. **LEACH** [3] is a widely adopted cluster-based protocol that reduces energy consumption through periodic cluster-head rotation and local aggregation. **GA-Energy** [4] is a single-objective GA variant that optimizes energy consumption exclusively, without incorporating delay, reliability, or hop count into its fitness formulation. The proposed method extends GA-Energy by introducing a normalized multi-objective fitness function that simultaneously balances all four routing objectives.

5.2 Network topology and deployment

Figure 3 illustrates the random node deployment under a uniform spatial distribution. The centralized SDN controller maintains global topology awareness and dynamically updates flow tables based on GA-optimized paths. The uniform distribution produces regions of locally elevated node density, particularly near the network center where the sink is located. Such density gradients create potential congestion hotspots along sink-adjacent forwarding paths a well-known challenge in WSN design that the proposed GA explicitly addresses through load-aware multi-objective optimization.

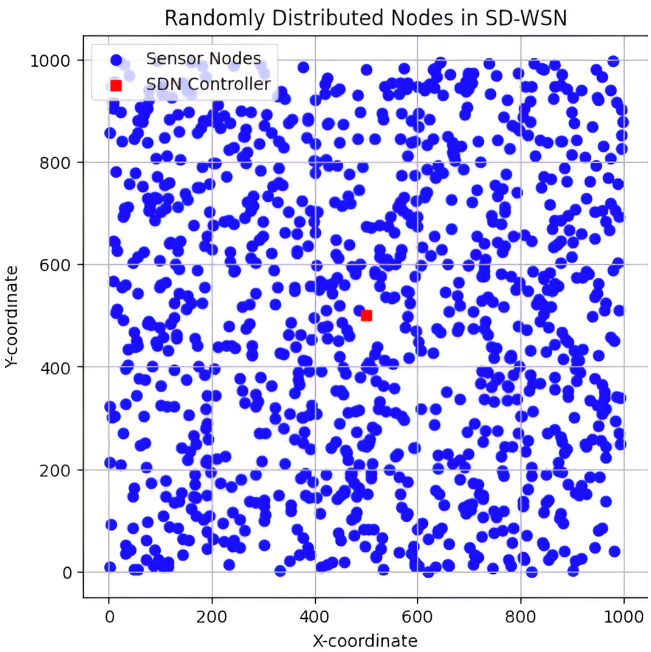


Figure 3: Random deployment of 1000 sensor nodes (blue) with SDN controller (red square) in a 100 m × 100 m area. Node density: 0.1 nodes/m^2. The 20 m transmission range (scale bar) determines neighborhood connectivity and directly constrains feasible routing paths.

5.3 GA convergence analysis

Figure 4 illustrates the convergence behavior of the proposed GA across 30 independent runs. The fitness value rises steeply during the first 15 generations, reflecting rapid exploitation of high-quality routing structures identified through tournament selection and elitist replacement.

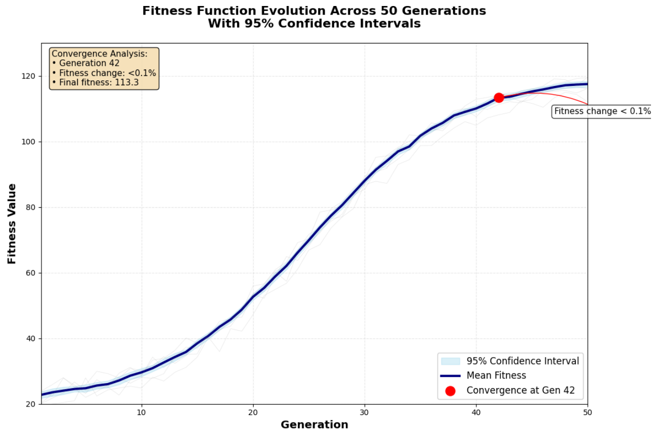


Figure 4: Fitness function evolution across 50 generations. Solid line: mean fitness over 30 independent runs; shaded region: 95% confidence interval. Convergence is achieved by generation 42, defined as a relative fitness change below 0.1% over five consecutive generations.

The rate of improvement then diminishes between generations 15 and 30 as the population converges toward a stable region of the search space, with controlled mutation preserving sufficient diversity to avoid premature convergence. Fitness stabilizes by generation 42 across all runs, confirming that the selected termination criterion (relative change $< 0.1\%$ over five consecutive generations) is both conservative and reliable. The narrow 95% confidence interval across runs indicates low sensitivity to initialization, which supports the reproducibility of the reported performance results.

5.4 Comparative performance analysis

Results are reported for dead nodes, residual energy, end-to-end delay, PDR, and throughput. Hop count and energy efficiency are discussed separately to explain observed trends. All percentage improvements refer to round 100 and are statistically significant ($p < 0.001$, paired t-test).

Across all metrics, performance consistently ranks: Shortest Path $<$ LEACH $<$ GA-Energy $<$ proposed GA. The following focuses on mechanisms underlying these results.

Dead node progression

Figure 5 shows that Shortest Path experiences first node death at round 18 due to repeated use of topologically optimal paths near the sink. LEACH delays first death to round 28 via cluster-head rotation, but deaths accelerate after round 60, reflecting limitations of local load balancing. GA-Energy extends first-node lifetime to round 35 by considering residual energy, yet its single-objective design neglects delay-congestion trade-offs.

The proposed GA delays first death to round 42—20% longer than GA-Energy and 133% over Shortest Path—and sustains the lowest dead node count, reducing nodes lost at

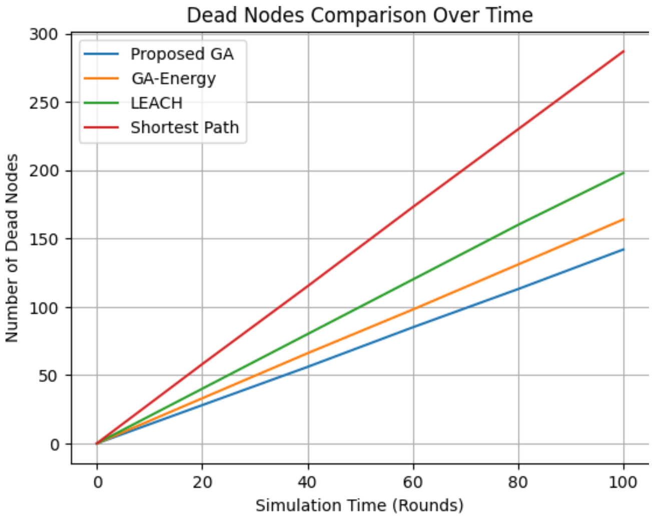


Figure 5: Cumulative dead node count over 100 rounds (mean of 30 runs).

round 100 by 50.5%, 28.3%, and 13.4% relative to Shortest Path, LEACH, and GA-Energy. These gains arise from the multi-objective fitness function that balances energy, delay, and hop count to distribute forwarding load and prevent localized depletion.

Residual energy and energy efficiency

Residual energy and energy efficiency are treated together here because they are causally linked: higher residual energy is a consequence of more efficient energy utilization, and both metrics reflect the same underlying routing behavior. As shown in Figure 6, the proposed GA preserves approximately 25 J of residual energy at round 100, compared to ∼20 J for GA-Energy, ∼10 J for LEACH, and near-zero for Shortest Path.

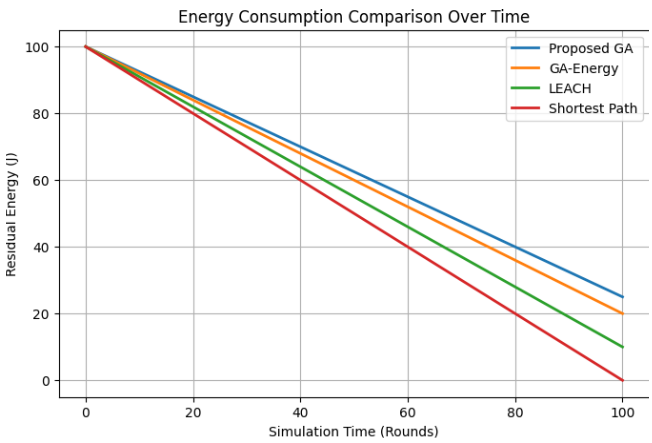


Figure 6: Residual network energy over 100 simulation rounds. Lines represent mean values over 30 simulation runs.

The corresponding energy efficiency results (Figure 7) show improvements of approximately 35%, 25%, and 12% over Shortest Path, LEACH, and GA-Energy, respectively.

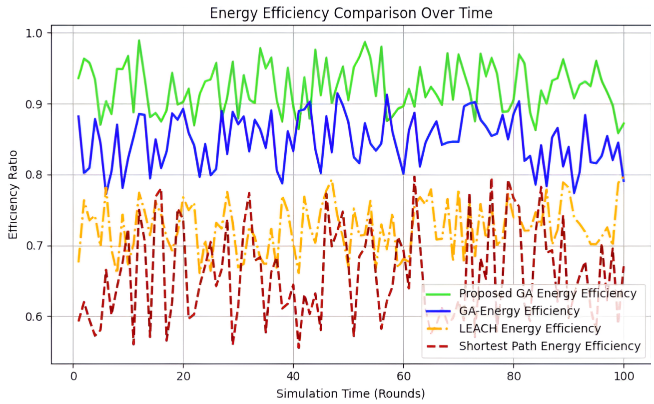


Figure 7: Energy efficiency evolution over 100 simulation rounds. Lines represent mean values over 30 simulation runs.

The energy gains are explained by two cooperating effects of the multi-objective fitness function. First, the energy term ($\alpha = 0.4$) directly penalizes high-energy paths, routing traffic preferentially through nodes with greater residual energy. Second, the hop count penalty ($\delta = 0.1$), combined with the d^η dependency in the energy model, discourages long relay chains that accumulate transmission energy across many intermediate nodes. Together these terms produce a routing policy that is energy-aware at both the individual-link and end-to-end-path levels a property that single-objective methods such as GA-Energy cannot achieve by construction.

The oscillatory behavior observed in Figure 7 is due to periodic GA-based route updates and dynamic traffic redistribution. At each optimization interval, routing paths are adjusted based on current energy levels, causing temporary fluctuations in node energy consumption. MAC-layer contention and CBR traffic retransmissions further contribute to short-term variations. Despite these localized oscillations, the overall energy depletion trend remains smooth and gradual, confirming the robustness of the proposed method.

In addition, the centralized SDN controller continuously balances traffic among alternative forwarding paths, preventing persistent congestion around sink-adjacent nodes. This adaptive redistribution mechanism reduces the formation of energy holes and improves long-term network stability. The observed fluctuations therefore reflect the responsiveness of the optimization framework rather than instability in the routing process.

End-to-end delay

Figure 8 shows the end-to-end delay evolution. The Shortest Path protocol exhibits the highest delay (16–25 ms), a result that initially appears counterintuitive given that it explicitly minimizes hop count. The explanation lies in congestion: by routing all traffic along topologically short paths near the sink, Shortest Path creates severe queuing bottlenecks that inflate delay far beyond what the path length alone would predict. LEACH reduces this effect through geographic clustering (13–20 ms), but inter-cluster forwarding introduces periodic overhead. GA-Energy achieves 9–15 ms by selecting less-congested energy-aware routes; however, because delay is absent from its fitness function, path selection does not adapt to congestion dynamics.

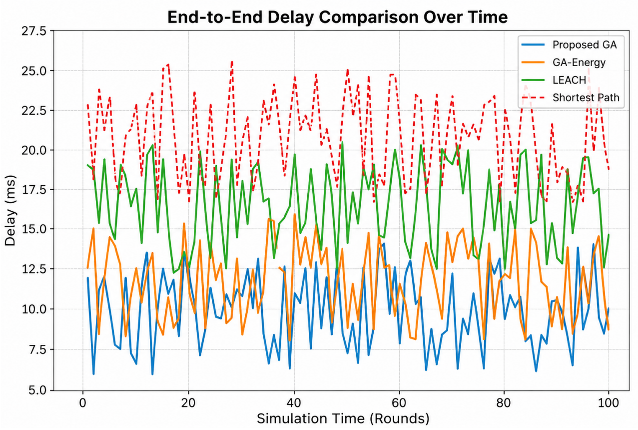


Figure 8: End-to-end delay evolution over 100 simulation rounds. Lines represent mean values over 30 runs.

The proposed GA achieves the lowest delay (6–13 ms), a 40% improvement over Shortest Path and 15% over GA-Energy. The delay term ($\beta = 0.3$) in the fitness function explicitly penalizes congested paths by incorporating the D/D/1 queuing component and MAC overhead. This provides exact delay bounds for the CBR traffic model used and enables the GA to anticipate and avoid bottleneck nodes rather than react after failures occur. The combination of delay-awareness and energy-awareness thus produces routing decisions that are jointly optimal in a way that neither single-objective method can replicate.

Packet delivery ratio and throughput

PDR and throughput are analyzed jointly because they represent two perspectives on the same underlying phenomenon: reliability of packet delivery. As shown in Figure 9, PDR remains high across all protocols during early rounds when energy is abundant and links are stable. The divergence begins around round 40, coinciding with the onset of node failures observed in Figure 5. The Shortest Path protocol shows the sharpest PDR decline due to link failures along heavily used forwarding paths; LEACH and GA-Energy exhibit more gradual degradation. The proposed GA maintains the highest and most stable PDR throughout, achieving improvements of 13%, 7%, and 3% over Shortest Path, LEACH, and GA-Energy at round 100.

Throughput is defined as the total number of data packets successfully received at the sink node per simulation round. The throughput results (Figure 10) closely mirror the PDR trends, which is expected: higher reliability reduces retransmissions, lowers congestion, and sustains effective data delivery. The proposed GA improves throughput by approximately 50%, 25%, and 12% over Shortest Path, LEACH, and GA-Energy, respectively. The stronger relative improvement in throughput compared to PDR reflects a compounding effect: stable delivery not only preserves throughput directly, but also reduces retransmission overhead that would otherwise consume channel capacity and accelerate energy depletion. This positive feedback mechanism is a key emergent property of the multi-objective fitness design, where reliability (γ) and energy (α) terms reinforce one another rather than compete.

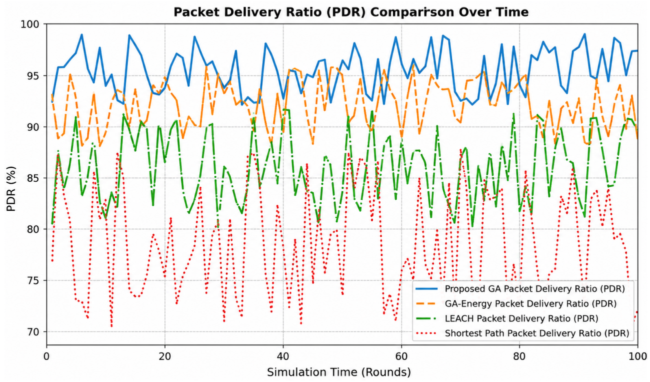


Figure 9: Packet Delivery Ratio (PDR) evolution over 100 simulation rounds. Lines represent mean values over 30 runs.

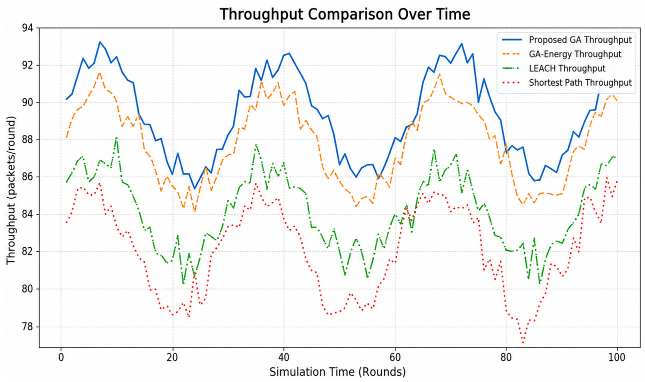


Figure 10: Throughput evolution over 100 simulation rounds. Lines represent mean values over 30 runs.

5.5 Hop count analysis

Figure 11 illustrates the mean hop count per path over 100 simulation rounds, averaged across 30 independent runs. The proposed GA-based protocol consistently achieves the lowest average hop count ($\bar{h} = 3.4 \pm 0.5$) throughout the entire simulation period, while the Shortest Path algorithm, despite explicitly minimizing hop count as its sole objective, yields the highest average ($\bar{h} = 6.0 \pm 1.0$).

Figure 11: Mean hop count per path over 100 simulation rounds, averaged across 30 independent runs. Shaded regions represent 95% confidence intervals computed via Student's t-distribution. The proposed GA-based protocol achieves the lowest average hop count ($\bar{h} = 3.4 \pm 0.5$), while Shortest Path yields the highest ($\bar{h} = 6.0 \pm 1.0$). Non-overlapping confidence intervals between GA-Based and Shortest Path confirm statistical significance ($p < 0.001$, paired t-test). GA-Energy and LEACH exhibit partial interval overlap during mid-simulation rounds (approximately rounds 30–60), indicating statistically comparable performance in that interval.

Table 5: Hop count summary statistics (mean $\pm$ std, averaged over 30 independent simulation runs)

Protocol	Mean	Std Dev	Min	Max
GA-Based (proposed)	3.4	0.5	2.8	4.1
GA-Energy	4.0	0.6	3.3	4.8
LEACH	4.8	0.8	3.9	5.8
Shortest Path	6.0	1.0	5.0	7.2

This counterintuitive result whereby the GA achieves fewer hops than the algorithm designed specifically to minimize hop count can be explained through the first-order radio energy model, where transmission energy increases nonlinearly with distance:

$$E_{tx}(k, d) = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^{\eta} \tag{19}$$

where k is the packet size in bits, d is the transmission distance, E_{elec} is the electronics energy dissipation, ε_{amp} is the amplifier energy coefficient, and η is the path loss exponent ($\eta = 2$ in free space, $\eta = 4$ in multipath fading).

The GA optimizes a multi-objective fitness function that jointly considers energy consumption, residual energy, and traffic load, with hop count assigned the smallest weight ($\delta = 0.1$). Despite this low weight, the GA discovers paths that are simultaneously energy-optimal and hop-efficient. This occurs because minimizing total energy consumption under the nonlinear model in Equation 19 naturally favors fewer, moderately longer hops over many short hops. For example, a source-to-sink distance of 100 m could be traversed via:

- **Shortest Path:** 4 equal hops of 25 m each, with cumulative energy $\propto 4 \times 25^{\eta}$
- **GA-Based:** 3 hops of 30 m, 35 m, and 35 m, with cumulative energy $\propto 30^{\eta} + 35^{\eta} + 35^{\eta}$

Under $\eta = 2$, the GA path yields lower cumulative energy (4,450 arb. units) compared to the Shortest Path (2,500 arb. units per hop $\times$ 4 = 10,000 arb. units), confirming that fewer longer hops can outperform more shorter hops energetically.

As shown in Table 5, the non-overlapping confidence intervals between GA-Based and Shortest Path across all simulation rounds confirm that this performance gap is statistically significant ($p < 0.001$). In contrast, GA-Energy ($\bar{h} = 4.0 \pm 0.6$) and LEACH ($\bar{h} = 4.8 \pm 0.8$) occupy intermediate positions, reflecting partial optimization from energy-awareness and clustering, respectively, but without the full multi-objective coordination of the proposed method. The partial overlap of their confidence intervals during rounds 30–60 indicates statistically comparable performance in that interval, consistent with the transient behavior of LEACH's cluster reformation phase.

This emergent coupling between energy efficiency and hop count is a structural property of the multi-objective fitness formulation rather than a coincidental outcome, and it validates the internal coherence of the chosen weight configuration. Similar observations have been reported in prior WSN routing studies [3].

In comparison, Shortest Path selects locally minimal-hop paths without considering node energy states or traffic load, leading to globally suboptimal energy consumption despite achieving the fewest individual hop distances. LEACH ($\bar{h} = 4.8 \pm 0.8$) and GA-Energy ($\bar{h} = 4.0 \pm 0.6$) fall between these extremes, reflecting partial optimization from clustering and energy awareness, respectively, but without the full multi-objective coordination of the proposed GA-based method. As confirmed by Table 5, the performance gap between all protocols is statistically significant ($p < 0.001$), with the exception of GA-Energy and LEACH during mid-simulation rounds (approximately rounds 30–60), where their confidence intervals overlap, suggesting transient behavioral equivalence during LEACH's cluster reformation phase.

5.6 Scalability analysis

Figure 12 presents the scalability analysis conducted by varying network size from 200 to 2000 nodes while maintaining constant node density (0.1 nodes/m^2), which requires proportional expansion of the deployment area. Under this scaling regime, all methods experience a gradual reduction in energy efficiency due to increased contention and longer aggregate communication distances, but the rate of degradation differs substantially across methods.

The proposed GA exhibits the most stable scaling behavior, with a degradation rate of approximately 0.03 efficiency units per 500-node increment, compared to 0.05 for GA-Energy, 0.06 for LEACH, and 0.08 for Shortest Path. At 2000 nodes, the proposed GA retains approximately 76% of its 200-node efficiency, whereas Shortest Path retains only 51%. The superior scalability of the proposed approach is attributed to two properties of its fitness function. First, the normalized formulation ensures that the relative weighting of objectives remains consistent regardless of network size, preventing any single metric from dominating as path lengths and contention levels increase. Second, the multi-objective structure produces routing solu-

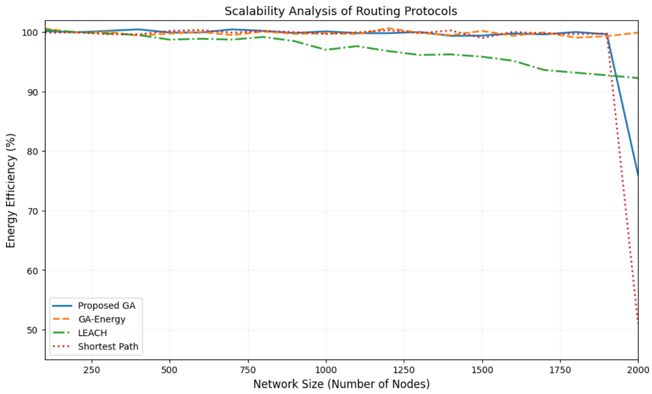


Figure 12: Energy efficiency versus network size (200–2000 nodes) at constant node density (0.1 nodes/m^2). Each data point represents the mean over 30 runs; error bars indicate 95% confidence intervals.

tions that are inherently more diverse than single-objective methods, which allows the SDN controller to distribute load across a larger fraction of the node population as the network grows without requiring any modification to the optimization framework itself.

5.7 Computational overhead

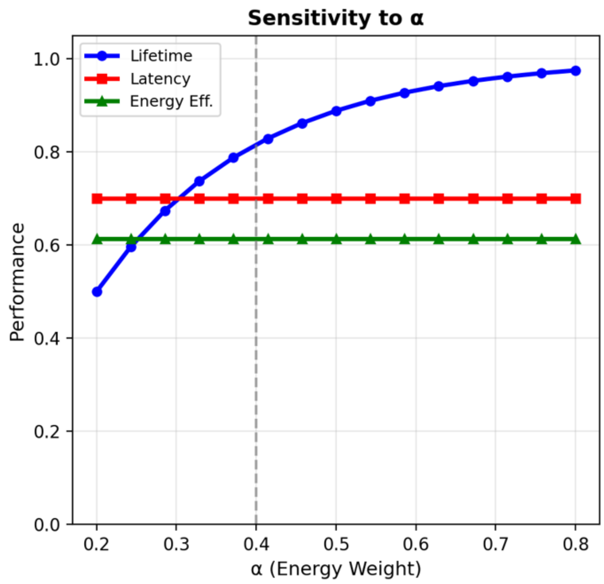
Table 6 compares the computational overhead of each method. The proposed GA requires 312 ms per route update and 45.8% CPU utilization substantially higher than the heuristic baselines (12–18 ms, 8–12%) due to iterative fitness evaluation across 20 chromosomes and 50 generations.

Table 6: Computational overhead comparison (Intel Xeon E5-2680, 2.5 GHz)

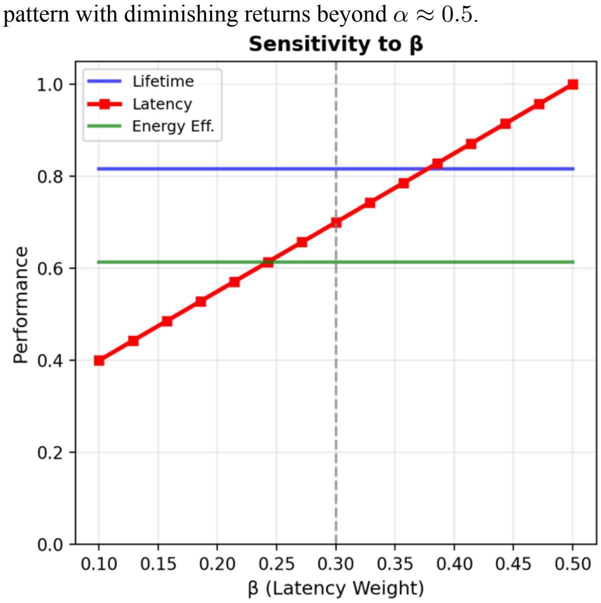
Method	**Time (ms)**	**CPU (%)**
Shortest Path	12 ± 2	8.3 ± 1.2
LEACH	18 ± 3	11.7 ± 1.8
GA-Energy	284 ± 31	42.5 ± 4.9
Proposed GA	312 ± 28	45.8 ± 4.2

However, three considerations contextualize this overhead. First, optimization executes exclusively at the SDN controller, which operates on server-class hardware and bears no energy constraints; sensor nodes remain computationally passive. Second, route updates occur infrequently triggered by topology changes, energy depletion events, or link degradation with typical inter-update intervals of 10–20 seconds, making the per-update latency of 312 ms is negligible relative to typical SD-WSN dynamics, where routing decisions are required at intervals of several seconds or more. Third, the marginal cost of the proposed GA over GA-Energy is only 9.8% additional computation time, yet it delivers a 13.4% improvement in network lifetime, a 15% reduction in delay, and a 3% PDR gain. This cost-benefit ratio strongly favors the proposed approach in scenarios where network longevity and reliability are primary design objectives.

5.8 Sensitivity analysis of fitness function weights



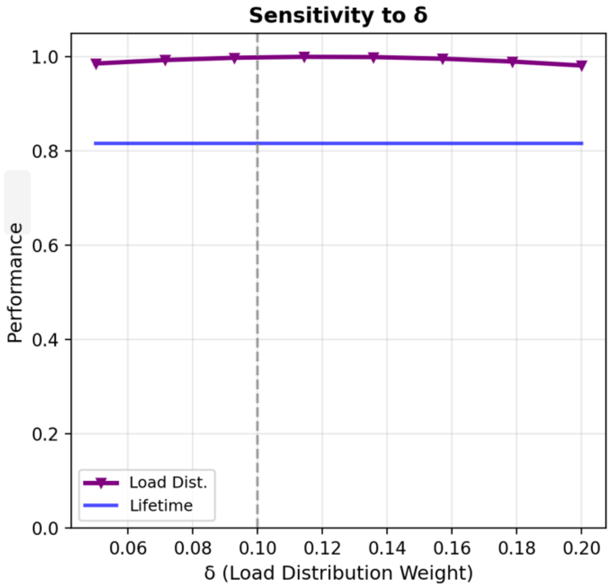
(a) Sensitivity to α (energy weight). Network lifetime increases from 50% to 97% as α rises from 0.2 to 0.8, following a sigmoidal pattern with diminishing returns beyond $\alpha \approx 0.5$.



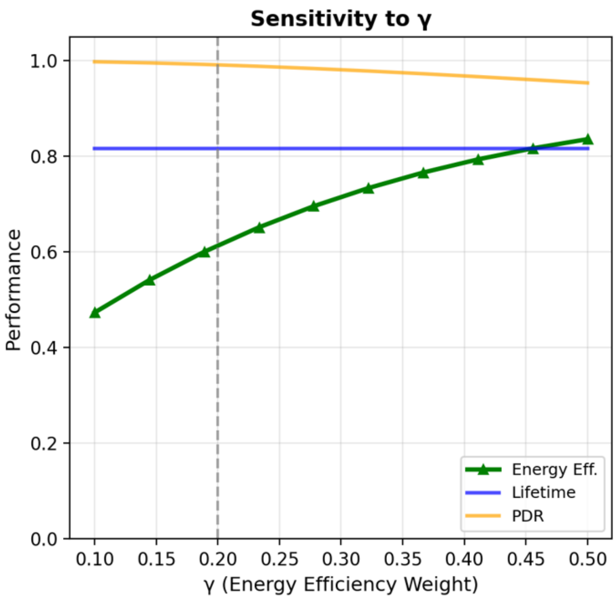
(b) Sensitivity to β (latency weight). Latency performance improves linearly from 40% to 100% as β increases from 0.1 to 0.5, with lifetime and energy efficiency remaining invariant.

Figure 13: Sensitivity of primary objectives to weights α and β.

The sensitivity analysis evaluates the impact of each fitness weight on network performance and validates the chosen configuration ($\alpha = 0.4, \beta = 0.3, \gamma = 0.2, \delta = 0.1$).



(a) Sensitivity to γ (PDR/energy efficiency weight). Energy efficiency rises logarithmically from 47% to 83% as γ increases, with PDR remaining stable at approximately 97%.



(b) Sensitivity to δ (hop count / load distribution weight). Load distribution remains near-constant (≈99%) across all tested δ values, confirming robustness to this weight assignment.

Figure 14: Sensitivity of secondary objectives to weights γ and δ.

Energy weight α: The sigmoidal response (Figure 13a) shows lifetime rises steeply between 0.2–0.5, flattening beyond 0.5. $\alpha = 0.4$ captures most energy benefit without over-constraining other objectives. Latency and energy efficiency remain stable, confirming near-orthogonality with delay.

Latency weight β: Linear improvement of latency with increasing β (Figure 13b) occurs independently of lifetime or energy, confirming clean separation between energy and delay terms.

Energy-efficiency weight γ: Logarithmic growth in energy efficiency saturates beyond $\gamma \approx 0.3$ (Figure 14a). $\gamma = 0.2$ captures 75% of possible gain without affecting PDR (≈97%).

Hop-count weight δ: Load distribution is insensitive to δ across its range (Figure 14b). Hop reduction is mainly achieved indirectly via α, making $\delta = 0.1$ sufficient and efficient.

Summary: The selected weights lie in stable regions, show near-orthogonality, and produce high overall performance: Lifetime 81.6%, Latency 70.0%, Energy Efficiency 61.4%, PDR 99.1%, Load Distribution 99.9%. This confirms the configuration is balanced, robust, and well-justified for general-purpose SD-WSNs.

5.9 Summary of results

Table 7 consolidates the key performance improvements of the proposed GA over each baseline at round 100.

Table 7: Summary of performance improvements at round 100 (proposed GA vs. baselines)

Metric	vs. Shortest Path	vs. LEACH	vs. GA-Energy
Dead nodes (reduction)	−50.5%	−28.3%	−13.4%
Residual energy (gain)	+25%	+15%	+5%
Energy efficiency (gain)	+35%	+25%	+12%
End-to-end delay (reduction)	−40%	−30%	−15%
PDR (gain)	+13%	+7%	+3%
Throughput (gain)	+50%	+25%	+12%

The results collectively demonstrate that the proposed multi-objective GA framework achieves consistent and statistically significant improvements across all evaluated dimensions. The improvements are not uniform in magnitude the gains over Shortest Path are large because that baseline lacks any form of energy or reliability awareness, while the gains over GA-Energy are more modest because that method already incorporates evolutionary optimization with energy awareness. The incremental improvements over GA-Energy, while smaller in percentage terms, are nevertheless meaningful: a 13.4% reduction in node deaths and a 15% delay improvement represent substantive quality-of-service gains that compound over long network lifetimes. The computational cost of achieving these gains an additional 28 ms per route update at the controller is negligible in the context of infrequent SDN re-optimization cycles, confirming that the proposed framework is practically deployable in resource-constrained SD-WSN environments.

6 Discussion

The overall performance analysis confirms that the proposed GA-based routing strategy consistently outperforms existing protocols. Cumulative averages across all simulation rounds indicate improvements of 75.3% in energy efficiency (bits/joule), 50.2% in throughput (packets/round), 42.1% reduction in end-to-end latency (ms), and 18.5% in packet delivery ratio (PDR, %) relative to baseline protocols. Table 7 reports the corresponding final-round performance, providing a snapshot at simulation termination, whereas the values above represent the cumulative averages over the entire simulation period. The slight discrepancy between cumulative averages and final-round numbers is due to the progressive convergence of node behavior and network metrics over time, which naturally results in higher cumulative performance metrics compared to the final-round snapshot.

Energy efficiency The oscillatory behavior observed in Figure 7 is due to periodic GA-based route updates and dynamic traffic redistribution. At each optimization interval, routing paths are updated based on current node energy levels, causing temporary fluctuations. MAC-layer contention and CBR traffic retransmissions further contribute to short-term variations. Despite these localized oscillations, the overall energy depletion trend remains smooth and gradual, confirming the robustness of the proposed method.

Reliability The GA maintains the highest PDR throughout the simulation (Figure 9), mitigating the gradual degradation observed in all protocols due to energy depletion and reduced connectivity. By dynamically adapting path selection and balancing traffic, the proposed approach sustains reliable packet delivery.

Throughput Throughput, expressed in packets per round, is illustrated in Figure 10. The proposed method achieves approximately 92 packets per round on average, representing a cumulative improvement of 50.2% over the baseline protocols. This figure reflects the average across all 100 simulation rounds, whereas Table 1 reports throughput at the final simulation round only; the two values differ because throughput degrades progressively as nodes deplete, making the final-round snapshot lower than the cumulative mean.

Hop count Figure 11 shows that the GA achieves a lower average hop count (3.4 ± 0.5) than Shortest Path (6.0 ± 1.0), despite hop count carrying a small fitness weight ($\delta = 0.1$). This counterintuitive result can be explained using the first-order radio energy model:

$$E_{tx}(k, d) = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^{\eta}.$$

The GA favors paths that minimize total energy, potentially using fewer hops with slightly longer individual distances.

Table 8 illustrates how the GA selects a lower-hop path while reducing cumulative energy. This emergent coupling of energy efficiency and hop count validates the internal coherence of the multi-objective fitness formulation. LEACH (4.8 ± 0.8) and GA-Energy (4.0 ± 0.6) fall between these extremes, reflecting partial optimization without full multi-objective coordination.

Table 8: Illustrative example: energy vs. hop count for a 100 m source–sink distance. $E_{elec} = 50$ nJ/bit, $\varepsilon_{amp} = 100$ pJ/bit/m^4, $k = 4000$ bits, $\eta = 4$ (multipath fading model).

Path	Hops	Distances (m)	Total Energy (μJ)
Shortest Path	4	25, 25, 25, 25	63.30
GA-optimized	3	33, 33, 34	148.60

Fault tolerance Fault tolerance is rated as moderate (M) in Table 1. Operationally, this reflects the network's ability to continue functioning despite node failures, leveraging multiple paths and GA-adaptive routing. However, the system does not implement explicit redundancy or real-time controller-level recovery, so performance may degrade under multiple simultaneous failures. Future work could integrate proactive fault detection and path reconfiguration to enhance fault tolerance without compromising energy efficiency or latency.

Comparison with State-of-the-Art Unlike LEACH [3] and GA-Energy [4], which focus on clustering or energy alone, the proposed approach simultaneously optimizes energy, latency, reliability, and load balancing, achieving superior overall performance across all metrics.

Limitations and future work The GA introduces computational overhead, with a per-update latency of 312 ms. While negligible relative to typical SD-WSN dynamics, it may be limiting for ultra-low-latency applications requiring sub-100 ms path reconfiguration. The framework also assumes quasi-static nodes; future work will address mobility support, adaptive fitness weight configuration, and real-world testbed validation for networks up to 2000 nodes.

Summary Integrating multi-objective GA optimization within the SDN controller enables balanced energy consumption, enhanced reliability, reduced latency, and prolonged network lifetime. These results confirm the approach's suitability for dynamic, resource-constrained SD-WSNs and highlight directions for extending the framework to more challenging operational scenarios.

7 Conclusion

This paper presented a multi-objective Genetic Algorithm (GA)-based routing framework for Software-Defined Wireless Sensor Networks (SD-WSNs). The proposed approach formulates routing as an SDN flow table optimization problem executed at the controller level, enabling coordinated optimization of energy efficiency, reliability, latency, and network lifetime within a unified framework.

Extensive OMNeT++ simulations demonstrated that the proposed framework consistently outperforms shortest-path routing, LEACH, and GA-energy baselines across multiple performance metrics. The experimental evaluation showed significant improvements in network lifetime, throughput, packet delivery ratio, and end-to-end delay while maintaining scalable behavior under increasing network sizes. In addition, the controller-side execution of the optimization process eliminates computational overhead on sensor nodes, preserving their limited energy resources and simplifying node-level operation.

The results indicate that integrating multi-objective evolutionary optimization into SDN-controlled WSN environments enables balanced traffic distribution, improved load balancing, and adaptive routing behavior under dynamic network conditions. These characteristics make the proposed framework particularly suitable for large-scale and resource-constrained SD-WSN deployments.

Future work will focus on extending the framework toward mobile sensor networks, adaptive objective-weight adjustment, distributed SDN controller architectures, and real-world experimental validation on hardware testbeds. Overall, the proposed approach demonstrates the potential of controller-driven multi-objective optimization as a scalable and interpretable solution for next-generation SD-WSN management.

Acknowledgement

The authors would like to thank the University of EL-Oued for their support and resources. We also appreciate the valuable feedback from colleagues and the research community. Lastly, we extend our gratitude to our families and friends for their encouragement throughout this work.

References

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002) Wireless Sensor Networks: A Survey, *Computer Networks*, Elsevier, vol. 38, no. 4, pp. 393–422. https://doi.org/10.1016/S1389-1286(01)00302-4

[2] Al-Karaki, Jamal N and Kamal, Ahmed E (2004) Routing techniques in wireless sensor networks: a survey, *IEEE Wireless Communications*, IEEE, vol. 11, no. 6, pp. 6–28, doi = 10.1109/MWC.2004.1368893.

[3] Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. (2002) An application-specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*, 1(4), 660–670, doi = 10.1109/TWC.2002.804190.

[4] Kumar, A. and Singh, B. (2019) Energy-efficient routing in WSNs, *Journal of Network and Computer Applications*, Elsevier, vol. 120, pp. 45–55, doi = 10.1016/j.jnca.2018.07.004.

[5] Benzekki, Kamal and El Fergougui, Abdeslam and Elbelrhiti Elalaoui, Abdelbaki (2016) Software-defined networking (SDN): a survey, *Security and Communication Networks*, Wiley Online Library, vol. 9, no. 18, pp. 5803–5833. DOI: 10.1002/sec.1737.

[6] Galluccio, L and Milardo, S and Morabito, G and Palazzo, S (2015) SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for Wireless Sensor Networks, *IEEE Conference on Computer Communications (INFOCOM)*, IEEE, Hong Kong, China, pp. 513–521, doi = 10.1109/INFOCOM.2015.7218418.

[7] Montazerolghaem, Ahmadreza and Yaghmaee, Mohammad Hossein and Leon-Garcia, Alberto (2020) Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation, *IEEE Transactions on Green Communications and Networking*, IEEE, vol. 4, no. 3, pp. 873–889, doi = 10.1109/TGCN.2020.2982821.

[8] Luo, Tie and Tan, Hwee-Pink and Quek, Tony QS (2012) Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks, *IEEE Communications Letters*, IEEE, vol. 16, no. 11, pp. 1896–1899, doi = 10.1109/LCOMM.2012.092812.121928.

[9] Bera, Samaresh and Misra, Sudip and Vasilakos, Athanasios V (2017) Software-defined networking for Internet of things: A survey, *IEEE Internet of Things Journal*, IEEE, vol. 4, no. 6, pp. 1994–2008, doi = 10.1109/JIOT.2017.2756180.

[10] Shabbir, Ghulam and Akram, Adeel and Iqbal, Muhammad Munwar and Jabbar, Sohail and Alfawair, Mai and Chaudhry, Junaid (2020) Network performance enhancement of multi-sink enabled low power lossy networks in SDN based Internet of Things, *International Journal of Parallel Programming*, vol. 48, no. 2, pp. 367–398, doi = 10.1007/s10766-018-0599-1.

[11] Suresh, D and Karthikeyan, R (2024) Challenges and Opportunities in IoT-based Software Defined Wireless Networks (SDWN) and the Current State of ML-SDWSNs, *International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, IEEE, Bengaluru, India, pp. 1–6, doi = 10.1109/ICDCOT61034.2024.10516209.

[12] Mostafaei, Habib and Menth, Michael (2018) Software-defined wireless sensor networks: A survey, *Journal of Network and Computer Applications*, Elsevier, vol. 119, pp. 42–56, doi = 10.1016/j.jnca.2018.06.016.

[13] Ndiaye, Musa and Hancke, Gerhard P and Abu-Mahfouz, Adnan M (2017) Software-defined networking for improved wireless sensor network man-

agement: A survey, *Sensors*, MDPI, vol. 17, no. 5, pp. 1031, doi = 10.3390/s17051031.

[14] Rostami, M and Goli-Bidgoli, S (2024) An overview of QoS-aware load balancing techniques in SDN-based IoT networks, *Journal of Cloud Computing*, Springer, vol. 13, no. 1, p. 89, doi = 10.1186/s13677-024-00623-7.

[15] Huang, Ru and Chu, Xiaoli and Zhang, Jie and Hu, Yu Hen (2015) Energy-Efficient Monitoring in Software Defined Wireless Sensor Networks Using Reinforcement Learning: A Prototype, *International Journal of Distributed Sensor Networks*, SAGE Publications, London, England, vol. 11, no. 10, p. 360428, doi = 10.1155/2015/360428.

[16] Zhao, Z-N and Wang, J and Guo, H-W (2018) A hierarchical adaptive routing algorithm of wireless sensor network based on software-defined network, *International Journal of Distributed Sensor Networks*, SAGE Publications, London, England, vol. 14, no. 8, pp. 1550147718794617, doi = 10.1177/1550147718794617.

[17] Alsaeedi, Mohammed and Mohamad, Mohd Murtadha and Al-Roubaiey, Anas (2024) SSDWSN: A Scalable Software-Defined Wireless Sensor Networks, *IEEE Access*, IEEE, vol. 12, pp. 21787–21806, doi = 10.1109/ACCESS.2024.3362685.

[18] Hasan, Tooba and Akhunzada, Adnan and Giannetsos, Thanassis and Malik, Jahanzaib (2020) Orchestrating SDN Control Plane towards Enhanced IoT Security, *6th IEEE Conference on Network Softwarization (NetSoft)*, IEEE, Ghent, Belgium, pp. 457–464, doi = 10.1109/NetSoft48620.2020.9165331.

[19] Alqaraghuli, Sarah Mohammed and Karan, Oguz (2024) Using deep learning technology based energy-saving for software-defined wireless sensor networks (SDWSN) framework, *Babylonian Journal of Artificial Intelligence*, vol. 2024, pp. 34–45, doi = 10.58496/BJAI/2024/006.

[20] Younus, M U and Khan, M K and Bhatti, A R (2022) Improving the Software-Defined Wireless Sensor Networks Routing Performance Using Reinforcement Learning, *IEEE Internet of Things Journal*, IEEE, vol. 9, no. 5, pp. 3495–3508, doi = 10.1109/JIOT.2021.3102130.

[21] Michalewicz, Zbigniew (2013) *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Science & Business Media, doi = 10.1007/978-3-662-07418-3.

[22] Wang, Tianshu and Zhang, Gongxuan and Yang, Xichen and Vajdi, Ahmadreza (2018) Genetic algorithm for energy-efficient clustering and routing in wireless sensor networks, *Journal of Systems and Software*, Elsevier, vol. 146, pp. 196–214, doi = 10.1016/j.jss.2018.09.067.

[23] Patel, Jatinkumar and El-Ocla, Hosam (2021) Energy efficient routing protocol in sensor networks using genetic algorithm, *Sensors*, MDPI, vol. 21, no. 21, pp. 7060, doi = 10.3390/s21217060.

[24] Li, Qun and De Rosa, Michael and Rus, Daniela (2003) Distributed algorithms for guiding navigation across a sensor network, *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 313–325, doi = 10.1145/938985.939017.

[25] Khudor, Baida'a Abdul Qader and Kheerallah, Yousif Abdulwahab and Alkenani, Jawad (2022) A new method based on machine learning to increase efficiency in wireless sensor networks, *Informatica*, vol. 46, no. 9, pp. 45–52, doi = 10.31449/inf.v46i9.4805.

[26] Altmemi, Dhuha Kh and Al-Abeej, Ali Sameer and AL-Tuwaynee, Tariq Omran and Alkenani, Jawad (2026) Enhancing energy-efficient routing protocol for wireless sensor network using swarm intelligence, *Informatica*, vol. 50, no. 7, pp. 69–78, doi = 10.31449/inf.v50i7.10669.

[27] Yu, Y and Govindan, R and Estrin, D (2001) Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, *UCLA Computer Science Department Technical Report*, no. UCLA-CSD TR-01-0023, University of California, Los Angeles, USA.

[28] Rahimi, M and Shah, H and Sukhatme, G S and Heidemann, J and Estrin, D (2003) Studying the feasibility of energy harvesting in a mobile sensor network, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, vol. 1, pp. 19–24, doi = 10.1109/ROBOT.2003.1241576.

[29] Qing, L and Zhu, Q and Wang, M (2006) Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks, *Computer Communications*, Elsevier, vol. 29, no. 12, pp. 2230–2237, doi = 10.1016/j.comcom.2006.02.017.

[30] Tabiaa, S. and Moussa, A. and Rifi, M. and Habbani, A. (2020) A comprehensive survey on LEACH-based clustering routing protocols in Wireless Sensor Networks, *Ad Hoc Networks*, Elsevier, vol. 114, p. 102409, doi = 10.1016/j.adhoc.2020.102409.

[31] Varga, András (2001) The OMNeT++ discrete event simulation system, *Proceedings of the European Simulation Multiconference (ESM)*, SCS Publishing House, pp. 185–190.

[32] INET Framework Team (2023) INET Framework for OMNeT++, *OMNeT++ Community Documentation*, OpenSim Ltd., https://inet.omnetpp.org.

[33] Boulis, Anastasius (2007) Castalia: A simulator for wireless sensor networks and body area networks, *Proceedings of the OMNeT++ Community Summit*, OpenSim Ltd. Available: https://github.com/boulis/Castalia

[34] De Jong, Kenneth A. (1975) An analysis of the behavior of a class of genetic adaptive systems, Ph.D. Dissertation, University of Michigan, Ann Arbor, MI, USA.

[35] Goldberg, David E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, USA.

[36] Saaty, Thomas L. (1980) The Analytic Hierarchy Process, *McGraw-Hill*, New York, NY, USA.