



Regular paper

Hybrid metaheuristic approach for multi objective controller placement for Smart City IoT networks

Sheeraz Ali Memon^a, Darius Andriukaitis^{a,*}, Vitas Grimaila^a, Tomyslav Sledevič^b,
Vytautas Markevičius^a, Mindaugas Žilys^a, Algimantas Valinevičius^a, Dangirutis Navikas^a,
Jaromir Konecny^c, Michal Prauzek^c

^a Department of Electronics Engineering, Kaunas University of Technology, Studentu St. 50, 51368 Kaunas, Lithuania

^b Department of Electronic Systems, Vilnius Gediminas Technical University, Saulėtekio Ave. 11, LT-10223 Vilnius, Lithuania

^c Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, 17. listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic

ARTICLE INFO

Keywords:

Internet of Things (IoT) Smart City Networks
Multi-objective Optimization Controller
Placement Problem (CPP)
Particle Swarm Optimization and Genetic
Algorithm

ABSTRACT

The rapid growth of the Internet of Things (IoT) in smart city environments has introduced significant challenges in controller placement. In Software Defined Networking (SDN) based IoT architectures, controller placement plays a critical role in overall network performance, as it directly influences scalability, latency, reliability, energy efficiency, and load balancing, making optimal controller deployment an important research problem. This paper proposes a realistic, multi-objective controller placement framework for large-scale smart city IoT networks. To improve practical relevance, the model incorporates real geographic data from Kaunas city (Lithuania), including city boundaries, building footprints, and water regions, enabling realistic urban deployment analysis. The proposed hybrid PSO-GA algorithm combines the global search capability of genetic techniques with the fast convergence characteristics of swarm intelligence to identify efficient controller placements. The performance of the proposed method is compared with random placement, K-Means, PSO, GA, and APDQN (Affinity Propagation Deep Q-network) based placement methods. The results demonstrate that the proposed method achieves a better balance among coverage efficiency, reliability, energy utilization, load balancing, and network stability. These findings indicate that the proposed method provides a scalable, and robust solution for large-scale smart city IoT controller placement optimization.

1. Introduction

The IoT has emerged as a key enabling technology for the development of modern smart cities. Through the large-scale deployment of interconnected sensors, devices, and communication systems, IoT supports a wide range of urban applications, including transportation, environmental monitoring, healthcare services, industrial automation, smart infrastructure, and public safety. Despite these benefits, the continuous growth in connected devices introduces major challenges in network management, operational efficiency, scalability, and communication reliability. One of the most critical challenges in IoT networks is effective controller placement [1,2], since controller deployment directly influences the overall performance, resilience, and reliability of the network. SDN has emerged as an effective networking paradigm to

address the limitations of traditional network architectures by separating the control plane from the data forwarding plane [3]. Beyond controller placement, centralized SDN control supports efficient D2D multicast communication by improving clustering efficiency and communication management [4]. SDN controller architectures can be broadly classified into centralized and distributed models [5] [6]. In the centralized model, a single controller manages the entire network infrastructure, which simplifies control management but introduces challenges related to scalability, controller overload, and a single point of failure. In contrast, a distributed SDN architecture employs multiple controllers across the network, improving scalability, increasing fault tolerance, enhancing reliability, and enabling better distribution of control responsibilities. This architecture is particularly suitable for large-scale IoT environments, where traffic demand, node density, and

* Corresponding author.

E-mail addresses: sheeraz.memon@ktu.edu (S.A. Memon), darius.andriukaitis@ktu.lt (D. Andriukaitis), vitas.grimaila@ktu.lt (V. Grimaila), tomyslav.sledevic@vilniustech.lt (T. Sledevič), vytautas.markevicius@ktu.lt (V. Markevičius), mindaugas.zilys@ktu.lt (M. Žilys), algimantas.valinevicius@ktu.lt (A. Valinevičius), dangirutis.navikas@ktu.lt (D. Navikas), jaromir.konecny@vsb.cz (J. Konecny), michal.prauzek@vsb.cz (M. Prauzek).

<https://doi.org/10.1016/j.aeue.2026.156522>

Received 17 May 2026; Accepted 15 July 2026

Available online 16 July 2026

1434-8411/© 2026 The Author(s).

Published by Elsevier GmbH. This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

communication requirements vary spatially. Highly scalable network topologies are less susceptible to bottlenecks, which enhance network efficiency, and support stable communication [7]. Controller placement in SDN based IoT networks has a direct impact on several important network performance metrics, including latency, packet loss, load balancing, redundancy, fault tolerance, scalability, and energy utilization. IoT communication is supported by multiple wireless technologies, such as Bluetooth, ZigBee, IEEE 802.15.4, LTE, 6LoWPAN, NB IoT, and 5G, each designed to support reliable data exchange under different operational requirements [8]. LPWAN technologies, including SigFox, NB IoT, and LoRaWAN, are widely adopted because they offer low power consumption, low deployment cost, and long communication range for IoT applications [9]. Among these technologies, NB IoT is particularly suitable for smart city applications because of its low energy consumption, long communication range, and ability to connect a massive number of low power IoT devices [10]. However, inefficient controller placement can lead to increased communication delay, controller overloading, poor backup coverage, uneven traffic distribution, and reduced network adaptability under dynamic operating conditions. Determining the optimal number and placement of controllers remains a major challenge in IoT network design [2]. Controller placement decisions are difficult to modify after deployment due to infrastructure constraints, operational complexity, and high costs. Therefore, selecting effective controller locations during the design stage is essential for long term network efficiency. Several approaches have been proposed to address this problem, including random deployment, clustering based methods such as K-Means, swarm-based optimization methods such as PSO, evolutionary algorithms such as GA, and learning based placement techniques using reinforcement learning models. Although these approaches have demonstrated performance improvements, many existing studies rely on simplified network topologies and limited optimization objectives [11]. Another important limitation in current research is the lack of realistic urban modeling. Most controller placement studies do not adequately account for geographic constraints such as administrative boundaries, lakes and rivers, spatial building of distribution, and heterogeneous traffic concentrations across urban areas. In practical smart city deployments, building coverage is a major consideration because IoT services are heavily concentrated in residential, commercial, industrial, healthcare, and educational zones. Ignoring these factors reduces the practical applicability of optimization results. The credibility of network simulations depends on how closely the simulation environment represents real world deployment conditions. Realistic simulation models improve the reliability, reproducibility, and practical applicability of the obtained results, making them more suitable for real world smart city IoT deployments [12]. Moreover, several existing works validate their approaches using only limited baseline method comparisons, which restricts broader performance evaluation. To overcome these limitations, this study presents a realistic and multi-objective framework for controller placement optimization in a smart city IoT network using a hybrid PSO-GA method. The proposed framework utilizes real geographic data from Kaunas city (Lithuania), including city boundaries, building footprints, and water regions [13], to construct a practical deployment model. A large-scale network consisting of 2000 IoT nodes was generated using both clustered and random spatial distributions to represent dense and sparse urban conditions. The proposed hybrid optimization approach combines the global exploration capability of GA with the fast convergence capability of PSO [14] to identify efficient controller placements. The fitness function used in the optimization process integrates multiple network performance metrics, including node coverage, packet loss, latency, energy consumption, controller load distribution, fault tolerance, backup node connectivity, redundancy, and scalability, to ensure efficient and reliable network performance. To validate its effectiveness, the proposed method is evaluated against five controller placement methods, including random placement, K-Means clustering, PSO, GA, and APDQN placement. In addition, building coverage analysis is

included in the evaluation process to provide a more realistic assessment of urban deployment. The proposed method achieves a balanced trade-off among coverage efficiency, reliability, scalability, and stability under realistic deployment conditions, making it a practical solution for controller placement in large-scale smart city IoT networks. Unlike existing controller placement methods that optimize only a limited number of performance metrics, the proposed framework simultaneously optimizes nine key network performance metrics to achieve a balanced and reliable solution. In addition, the method uses real geographic and building footprint data from Kaunas city (Lithuania) for realistic urban deployment analysis and validates performance through comparison with five other methods. This work substantially extends a preliminary study [15], which evaluated controller placement in the IoT environment without considering building coverage analysis and with comparisons limited to only two methods, namely Random and K-Means. The present study introduces three key advancements. First, building coverage and city boundaries are integrated to provide a more realistic assessment of deployment effectiveness in smart city environments. Second, three additional approaches, namely PSO, GA, and APDQN, are included to enable a broader and more rigorous comparative analysis. Third, nine performance metrics are evaluated, whereas only seven metrics were considered in the preliminary study. Unlike previous studies that primarily focus on optimizing a limited number of performance metrics and rely on synthetic network topologies, this work develops a realistic controller placement framework for large scale smart city IoT networks. The novelty of the proposed framework lies in the integration of real geographic information, a comprehensive multi objective fitness function, building coverage analysis, and extensive comparative evaluation within a unified framework, thereby improving the practical applicability of controller placement for real world smart city IoT deployments. The main contributions of this study are summarized as follows:

- A realistic smart city IoT deployment framework is developed using real geographic data from Kaunas city (Lithuania), including city boundaries, building footprints, and water regions, to model practical urban deployment constraints and improve the realism of controller placement analysis rather than synthetic network topologies.
- A multi-objective fitness function is introduced that jointly optimizes node coverage, packet loss, latency, energy consumption, controller load balancing, fault tolerance, redundancy, scalability, and spatial distribution, instead of optimizing only one or two objectives as in many existing studies. Thereby providing a balanced and reliable controller placement strategy for smart city IoT environments.
- Building coverage analysis is incorporated as an additional evaluation metric, enabling the proposed method to assess not only network-level performance but also practical urban service coverage across residential, commercial, industrial, and educational areas, which has largely been neglected in existing studies.
- A comprehensive comparative evaluation is conducted against five controller placement methods, namely random placement, K-Means clustering, PSO, GA, and APDQN, demonstrating that the proposed hybrid method achieves a better trade-off among coverage efficiency, reliability, load balancing, and overall network stability under realistic deployment conditions.
- The results demonstrate the applicability of the proposed method for large-scale smart city IoT networks, showing its effectiveness as a practical and scalable solution for urban IoT infrastructure.

2. Related work

The Controller Placement Problem (CPP) was first introduced by B. Heller et al. [16], who analyzed how the number and placement of controllers affect overall network performance. Since then, numerous studies have explored various aspects of controller placement, including

latency minimization, reliability, scalability, and fault tolerance. This section reviews key contributions in the literature and provides an overview of research in this area.

2.1. Hybrid DEWO based controller placement

Keshari et al. [17] investigated optimal controller placement in software defined IoT networks for smart city environments using a hybrid Differential Evolution and Whale Optimization (DEWO) algorithm. The study utilized three network topologies, namely TataNld, Deutsche, and Forthnet, derived from the Internet Topology Zoo. Node locations and inter-node distances were derived from geographical coordinates, longitude and latitude, based on methods reported in Bagaa et al. [18] and Hu et al. [19]. The optimization process was initialized with a population size of 100, while the crossover rate and scaling factor were carefully selected to enhance performance. The algorithm iteratively evaluated the fitness function over 200 iterations, using mutation and crossover operations to maintain population diversity and avoid premature convergence. The global best solution was continuously updated until the termination criteria were met. The proposed DEWO approach was compared with PSO and Firefly Algorithm (FFA) for performance evaluation.

The results shown in Table 1 demonstrate that DEWO significantly reduced network latency across all considered topologies. DEWO achieved improvements of 7.82%, 20.25%, and 9.27% over PSO for TataNld, Deutsche, and Forthnet topologies, respectively. Similarly, compared with FFA, DEWO improved performance by 2.35%, 3.55%, and 6.62% for the corresponding topologies. The experiments considered network sizes ranging from 50 to 150 switch nodes, with 1 to 40 controllers. The optimization objective is mainly centered on minimizing latency, while effective controller placement in smart city IoT networks requires simultaneous consideration of multi objective factors, including load balancing, resilience, deployment cost, and controller coordination overhead.

2.2. Controller placement using the Louvain algorithm and Betweenness centrality

Abderrahmane and Drid [20] proposed a controller placement approach for SDN based IoT networks using the Louvain Algorithm and Betweenness Centrality. The Louvain algorithm was adopted due to its scalability and efficiency in analyzing large-scale networks [21], making it suitable for complex and dynamic SDN-IoT environments. It enables rapid detection of community structures, supporting near real time decision making. By identifying densely connected communities, controllers can be placed closer to nodes, reducing communication distance and minimizing latency. Community detection also facilitates effective load balancing by distributing control load across multiple controllers while enhancing reliability and fault tolerance by localizing failures within specific regions. Furthermore, this approach improves overall network performance through better resource allocation, traffic management, and quality of service optimization. Betweenness centrality also helps to identify key nodes along major communication paths. Placing controllers near these nodes improves information flow, reduces communication delay, and enhances routing efficiency [22]. This strengthens network resilience, as high-centrality nodes play a key role in maintaining connectivity during failures [23]. Moreover, analyzing betweenness centrality enables the identification of vulnerable network

Table 1
Latency comparison of DEWO, PSO, and FFA.

Topology	Improvement over PSO (%)	Improvement over FFA (%)
TataNld	7.82%	2.35%
Deutsche	20.25%	3.55%
Forthnet	9.27%	6.62%

regions, supporting fault-tolerant controller placement. Together, these methods address key challenges in SDN based IoT controller placement, including latency minimization, routing optimization, load balancing, scalability, and fault tolerance. Table 2 presents an execution time comparison between the proposed method and the optimal solution across different network sizes.

The results show that the proposed method significantly reduces execution time compared to the optimal solution. For example, at 400 nodes, the optimal solution required 183,168,000 ms, while the proposed method completed execution in only 1720 ms. In addition, the optimal solution does not produce results beyond 450 nodes, whereas the proposed method continues to perform efficiently for networks of up to 1000 nodes. This highlights the scalability and computational efficiency of the proposed approach for large-scale SDN based IoT networks. In terms of the number of controllers, both approaches perform similarly in small-scale networks with fewer than 350 nodes. However, as the network size increases beyond 600 nodes, the proposed method identifies an optimal configuration consisting of 10 controllers, demonstrating improved scalability and operational stability in larger environments. Latency analysis further shows that the proposed method consistently outperforms the optimal solution across different topologies. Even in large-scale scenarios where the optimal solution fails to produce results, the proposed method maintains low latency, reaching approximately 5 ms at 1000 nodes. Reliability analysis shows that the proposed method achieves reliability values close to 1 across all network sizes, indicating strong fault tolerance and stable performance in both small and large-scale environments. Although computationally efficient, the method assumes that community structure and node centrality alone are sufficient indicators for controller placement, which may lead to suboptimal decisions in heterogeneous and highly dynamic IoT environments where traffic behavior and controller capacities continuously change.

2.3. ANP based clustering approach for controller placement

Ali and Roh [1] addressed the CPP in Software-Defined Internet of Things (SD-IoT) by proposing a clustering-based approach that utilizes the Analytical Network Process (ANP), a multi criteria decision making (MCDM). In SDN, network intelligence is centralized at controllers, which are distributed to avoid a single point of failure and improve load balancing. However, improper controller placement increases end-to-end (E2E) latency between switches and controllers. To overcome the limitations of evaluating controller placement across an entire network, the authors proposed segmenting the network into clusters, enabling more efficient assignment of switches to controllers. The proposed method uses ANP as a multi-criteria decision-making technique to optimize controller placement based on several factors, where propagation latency is one of the main factors. Although propagation latency is considered one of the main performance parameters, its modeling assumptions and integration with other dynamic network conditions are not extensively elaborated. For example, K-Means and K-Center based algorithms [16,24] define latency using Euclidean distance (ED). The study shows that latency between switches and controllers is a key factor

Table 2
Execution time comparison across network sizes.

Number of Nodes	Proposed Method (ms)	Optimal Solution (ms)
50	219	1210
100	312	6880
150	469	68,400
200	578	882,000
250	766	1,947,600
300	984	8,928,000
350	1540	58,464,000
400	1720	183,168,000
450	2100	No solution

affecting network responsiveness [25], as higher delays can reduce a controller's ability to handle network events efficiently. For evaluation, the analysis was conducted on real world network topologies, including Abilene Network, US_Net, OS3E, and Interoute, which vary in the number of nodes and edges. The characteristics of these selected network topologies are presented in Table 3. The topologies vary in size and complexity, ranging from small-scale networks such as Abilene with 11 nodes and 14 edges to larger networks such as Interoute with 110 nodes and 149 edges. US_Net and OS3E represent medium-sized networks with 24 and 34 nodes, respectively. This variation in network size enables a comprehensive analysis of the proposed approach under different conditions, from simple to complex network structures.

Table 4 presents the E2E delay comparison between the proposed method and the K-Means approach. The results show that the proposed method consistently achieves lower delay across all network topologies. For example, in the Interoute topology, the proposed method demonstrates an approximate delay of 4000 ms compared to about 7500 ms for K-Means. Similar improvements are observed in OS3E and US_Net, while in smaller networks such as Abilene, both methods produce lower delays, but the proposed method still performs better. This indicates that the proposed approach is effective in reducing communication delay, especially in larger and more complex networks.

Table 5 shows the controller-to-controller (C2C) delay comparison. Similarly, the proposed method outperforms the K-Means approach across all topologies. In the Interoute topology, the delay is reduced from around 7000 ms to 5000 ms, and in OS3E and US_Net, noticeable improvements are observed. Even in smaller networks such as Abilene, the proposed method achieves slightly lower delay. These results confirm that the proposed approach improves coordination between controllers and enhances overall network efficiency.

The results indicate that the proposed ANP-based clustering approach consistently outperforms the traditional K-Means algorithm across all evaluated topologies. Significant reductions are observed in both E2E delay and C2C delay. These results demonstrate that clustering-based controller placement using ANP provides a more efficient controller placement strategy compared to conventional approaches, particularly in heterogeneous and large-scale SD-IoT networks. However, the effectiveness of the ANP based placement algorithm depends on fixed evaluation criteria, which may lead to sub-optimal placement decisions in heterogeneous and rapidly changing IoT networks.

2.4. Adaptive controller placement using the ESFO and POCO framework

Hans et al. [26] proposed an adaptive controller placement approach for SD-IoT networks using an enhanced sunflower optimization algorithm (ESFO) integrated with a Pareto-based controller placement framework (POCO). The method employs an adaptive fuzzy mechanism to improve decision making during controller placement, aiming to minimize latency, communication delay, and the number of controllers while maintaining load balance. The ESFO algorithm efficiently explores the search space by considering node density and distance metrics to determine optimal controller locations. The performance of the proposed approach was evaluated against several methods, namely PASIN, Hybrid SD, and PSO. Table 6 presents the simulation parameters used for evaluation.

Table 7 presents the delay comparison for different numbers of

Table 3
Network Topologies.

No	Topology	Nodes	Edges
1	Abilene	11	14
2	US_Net	24	42
3	OS3E	34	41
4	Interoute	110	149

Table 4
E2E delay comparison.

Topology	Proposed (ms)	K-Means (ms)
OS3E	~ 3500	~ 5000
US_Net	~ 3000	~ 4000
Abilene	~ 300	~ 400
Interoute	~ 4000	~ 7500

Table 5
C2C delay comparison.

Topology	Proposed (ms)	K-Means (ms)
OS3E	~ 4100	~ 4800
US_Net	~ 3500	~ 3800
Abilene	~ 300	~ 400
Interoute	~ 5000	~ 7000

Table 6
Simulation parameters.

Parameter	Value
Number of nodes	35
Number of controllers	3 (nodes 3, 19, 23)
Packet size	64 kB
Traffic rate	10–100 pps
Number of sunflower (n)	20%
Pollination rate (p)	0.05%
Mortality rate (m)	0.1%
Problem dimension (d)	2
Lower bound (LB)	[−5, −5]
Upper bound (UB)	[5, 5]
Maximum iterations	100

Table 7
Delay comparison (ms).

Controller	Proposed	PASIN	PSO	Hybrid SD
2	2.8×10^4	3×10^4	3.2×10^4	3.4×10^4
4	2.99×10^4	3.4×10^4	3.8×10^4	4×10^4
6	3.2×10^4	3.8×10^4	4.2×10^4	4.5×10^4
8	3.5×10^4	4.1×10^4	4.8×10^4	5×10^4
10	3.99×10^4	4.5×10^4	5.5×10^4	5.5×10^4
12	4.1×10^4	5×10^4	6.1×10^4	6×10^4
14	4.5×10^4	5.2×10^4	6.5×10^4	6.3×10^4
16	5×10^4	5.5×10^4	6.9×10^4	6.5×10^4
18	5.2×10^4	5.9×10^4	7.1×10^4	6.8×10^4
20	5.8×10^4	6.2×10^4	7.5×10^4	6.8×10^4

controllers. The results indicate that delay increases with the number of controllers for all methods. However, the proposed method consistently achieves lower delay compared to PASIN, PSO, and Hybrid SD across all configurations, demonstrating better performance.

Table 8 presents the average latency comparison for different numbers of controllers. The results indicate that latency decreases as the

Table 8
Average latency comparison (miles).

Controller	Proposed	PASIN	PSO	Hybrid SD
2	400	1100	500	900
4	180	900	210	440
6	100	590	180	390
8	100	400	150	220
10	100	340	150	200
12	80	240	150	170
14	60	220	100	170
16	40	210	80	170
18	40	220	80	150
20	40	200	60	150

number of controllers increases for all methods. However, the proposed method consistently achieves the lowest latency across all configurations. In contrast, PASIN, PSO, and Hybrid SD show higher latency, especially at lower controller, highlighting the effectiveness of the proposed approach. Overall, the results demonstrate that the proposed method significantly outperforms these techniques in terms of latency and communication delay. For instance, with only two controllers, the proposed approach achieves an average latency of 400 miles, which is 22.2% lower than PSO, 76.9% lower than Hybrid SD, and 91.89% lower than PASIN, indicating a substantial improvement in network performance. Although the proposed approach achieves notable improvements, its reliance on heuristic density and distance metrics may lead to suboptimal controller placement in dynamic IoT environments where traffic patterns, controller capacities, and service demands continuously evolve.

2.5. Cluster based controller placement using adaptive Grey wolf optimization (GWOAP)

Keshari et al. [27] proposed a cluster based intelligent method to manage controller load in SDN based IoT network. The authors employed hybrid optimization techniques to improve controller placement and load balancing in smart city networks. The study evaluated several algorithms, including GA, PSO, GA with Affinity Propagation (GAAP), PSO with Affinity Propagation (PSOAP), and Grey Wolf Optimization with Affinity Propagation (GWOAP). The main objective was to minimize communication cost while maintaining efficient controller placement and balanced load distribution across the network. This resulted in improved overall network performance. The performance of the algorithms was evaluated using an objective function over 300 iterations to determine the optimal communication cost. The results showed that the GWOAP approach outperformed the other algorithms in terms of number of controllers versus communication cost and iteration time versus communication cost.

Fig. 1 illustrates that communication cost decreases as the number of controllers increases up to an optimal threshold (approximately 5 to 6 controllers), after which the cost increases because of inter-controller overhead. Among the evaluated algorithms, GWOAP outperformed the other methods, namely GA, PSO, GAAP, and PSOAP, achieving lower communication costs in large-scale IoT networks.

Fig. 2 shows that all methods reduced communication cost rapidly in the early iterations and then gradually stabilized as they converge. Among the evaluated methods, GWOAP achieved the lowest communication cost and converged faster than the other algorithms. GA and PSO exhibited higher communication costs and slower convergence, while GAAP and PSOAP performed better but still did not surpass GWOAP. Overall, GWOAP provided faster convergence and more

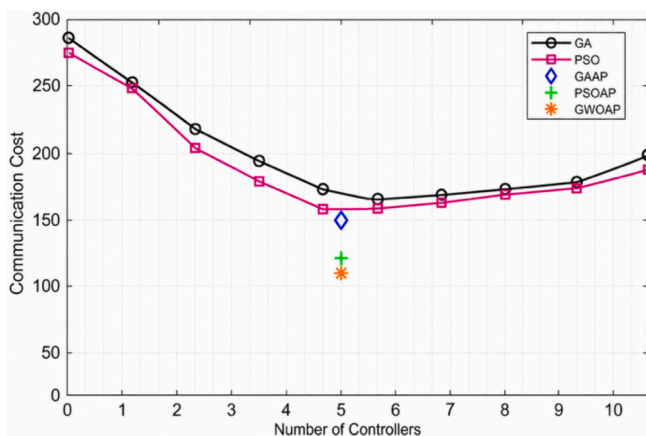


Fig. 1. Total communication cost vs. number of controllers [27].

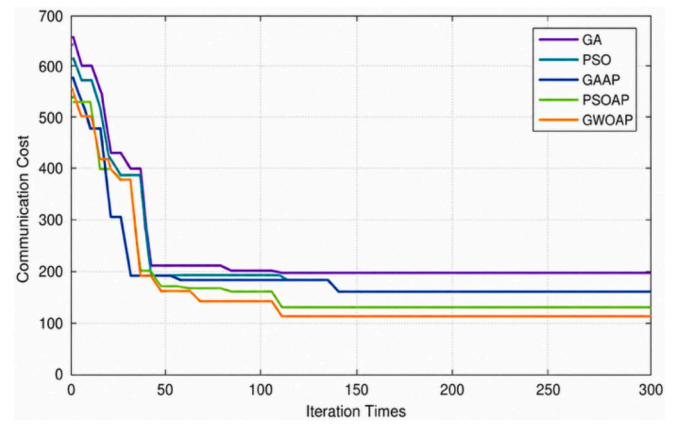


Fig. 2. Iteration times vs. communication cost [27].

efficient solutions for controller placement, significantly reducing communication cost in SDN-IoT networks and making it suitable for large-scale smart city applications. Despite its improved convergence and lower communication cost, the GWOAP algorithm remains largely heuristic driven and topology dependent, which may limit its adaptability to rapidly changing traffic demands, variable controller capacities for real world smart city deployments.

2.6. Submodular optimization based controller placement

B. Heller et al. [16] introduced the CPP in SDN, where the objective was to determine optimal controller locations that can efficiently manage all network nodes while satisfying constraints such as latency, fault tolerance, and resilience. Various optimization methods have been proposed for controller placement in IoT network. Among the early heuristic approaches, some methods selected nodes with the highest number of neighboring nodes to maximize coverage; however, such strategies do not necessarily guarantee minimum latency. To address these limitations, Tran et al. [28] proposed a submodular optimization-based approach, which was evaluated under realistic network conditions. The simulation environment was generated using Zipf's law to model node distribution. A network consisting of 200 nodes was constructed and divided into 10 clusters representing edge networks. The intra-cluster latency was set between 10 and 15 ms, while the inter-cluster latency ranged from 20 to 30 ms. The evaluation compared three approaches: submodular optimization, an optimal solution using the Gurobi solver, and a closest-distance baseline method. Table 9 presents the execution time comparison of three approaches across different network sizes. As the number of nodes increases, the execution time increased for all methods. The closest-distance method consistently showed the lowest execution time due to its simplicity, while the optimal solution required significantly higher computation time and becomes infeasible at 400 nodes. The submodular approach provided balanced performance, achieving lower execution time than the optimal method while maintaining better scalability.

Fig. 3 shows the relationship between average latency and the number of controllers along with a predefined latency threshold. As the number of controllers increases, the average latency decreases for all methods. This occurs because adding more controllers reduces the

Table 9
Execution time comparison.

Nodes	Submodular (s)	Optimal (s)	Closest-distance (s)
100	12	18	7
200	35	65	15
300	80	125	45
400	140	N/A	125

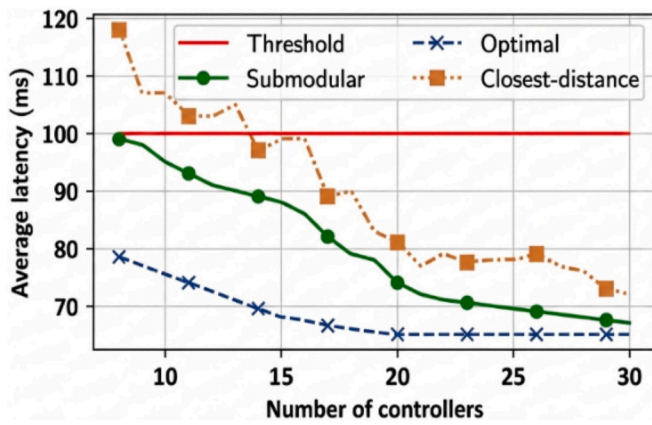


Fig. 3. Impact of the Number of Controllers on Average Network Latency [28].

distance between network nodes and controllers, thereby improving communication efficiency. The optimal method consistently achieved the lowest latency across all, followed by the submodular approach, while the closest-distance method exhibited the highest latency. Overall, the results indicated that the submodular approach provided a good balance between performance and practicality, achieving a balanced trade-off between latency reduction and scalability for larger networks. However, the submodular based placement strategy mainly focuses on reducing latency under predefined constraints, whereas effective controller placement in smart city IoT networks requires adaptive multi objective optimization that jointly considers controller capacity, fault tolerance, and dynamic traffic behavior.

2.7. Survey and comparative analysis of controller placement approaches

B. Heller et al. [16] highlighted that controller placement significantly affects key network performance metrics, particularly delay. Later studies extended this analysis to additional objectives such as load balancing, reliability, and energy efficiency. Abderrahmane et al. [2], in their survey paper, provided an overview of recent research efforts addressing the CPP in SDN based IoT networks. In SDN based IoT environments, controller placement is commonly formulated as a multi-objective optimization problem because optimizing one performance metric often affects others, creating trade-offs among latency, load balancing, reliability, and energy efficiency. The survey classified controller placement research into two subclasses based on the number of performance enhancement: mono-objective and multi-objective optimization. In mono-objective optimization approaches, the focus is on optimizing a single performance metric, typically latency. Among the analyzed studies, Keshari et al. [17] proposed a hybrid optimization method that combines Differential Evolution and Whale Optimization to address the CPP in SDN based IoT networks. The proposed DEWO algorithm was evaluated on TataNld, Deutsche, and Forthnet topologies. The results demonstrated that DEWO achieved lower latency compared to traditional approaches such as GA, PSO, and FFA. For example, in the TataNld topology, latency was reduced by 7.82% compared to PSO and 2.35% compared to FFA. The method also demonstrated scalability and stability as network size increased. However, the hybrid nature of the algorithm introduced higher computational complexity, which may limit its applicability in very large-scale networks. In contrast, multi-objective optimization approaches aim to optimize multiple performance metrics simultaneously, such as latency, load balancing, and controller distribution. Among these, Hans et al. [26] proposed a multi-objective controller placement strategy using an ESFO integrated with the POCO framework. The results showed significant improvements over existing methods such as PASIN, Hybrid SD, and PSO. Specifically, with two controllers, the proposed approach achieved an average latency of 400 miles, which was 22.2% lower than PSO, 76.9% lower than

Hybrid SD, and 91.89% lower than PASIN. Furthermore, the study evaluated performance under various network conditions, including controller imbalance and node failure scenarios, demonstrating the robustness and effectiveness of the proposed approach. However, most approaches have limitations, such as fixed controller counts, limited metric coverage, or scalability issues. This highlights the need for more adaptive and comprehensive optimization methods for efficient controller placement.

2.8. Controller placement in large scale networks using clustering techniques

Boudi et al. [29] investigated controller placement in large scale SDN based IoT networks through a case study for the Algerian network. The proposed topology consisted of 58 nodes (switches) and 140 links, covering a total link distance of approximately 24,307 km across an area of 2,381,741 km². To optimize controller placement, the authors applied the K-Means clustering algorithm to partition the network into multiple domains, where each cluster is managed by a single controller. In addition, they proposed enhanced versions of the K-Center and K-Median algorithms adapted for a weighted topology. In this model, traffic demand was proportional to the population of each state, making the approach population aware. The study also evaluated infrastructure deployment costs based on several factors, including the number of switches, link distances, number of controllers, and labor costs. However, the authors clarify that these costs primarily relate to network extension rather than full deployment, as a baseline infrastructure already exists. Results from simulations indicated that the proposed population aware SDN based IoT architecture achieved improved controller placement, reduced network latency, and enhanced system resilience. This made it particularly effective for applications such as disaster management, where responsiveness and reliability are critical.

Figs. 4 and 5 illustrate the impact of increasing the number of controllers on network latency using two placement strategies, K-Median and K-Center, in an unweighted node topology. In Fig. 4, the K-Median approach shows that average latency decreases consistently as the number of controllers increases from 2 to 5. This occurred because adding more controllers reduced the average distance between switches and their assigned controllers, which improved overall communication efficiency. In Fig. 5, the K-Center method demonstrated a similar decreasing trend, but in terms of maximum latency. As the number of controllers increased, the worst-case delay experienced by any node was reduced, since controllers were more evenly distributed across the network. Overall, the study demonstrated that controller placement in large-scale SD based IoT networks need to consider both infrastructure cost and latency objectives. The comparison between K-median and K-center provided valuable insights into selecting appropriate

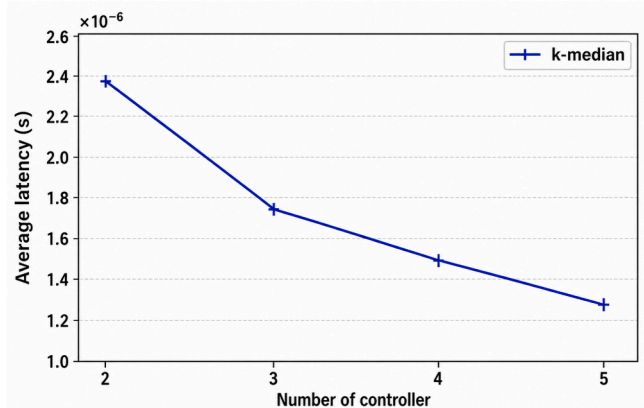


Fig. 4. Effect of the number of controllers on average latency using K-median placement [29].

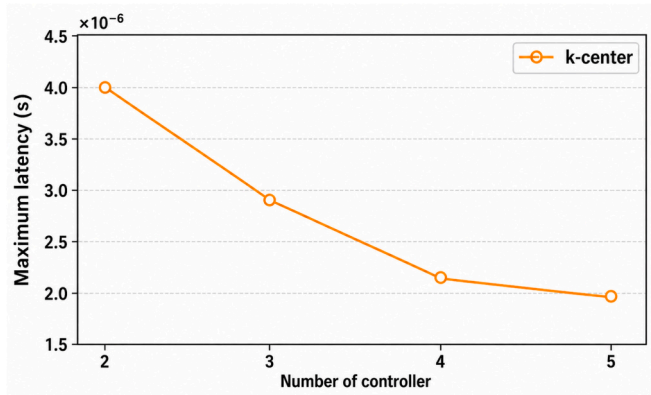


Fig. 5. Effect of the number of controllers on maximum latency using K-center placement [29].

optimization strategies depending on network requirements and performance priorities.

2.9. Energy-efficient controller placement using the EnPlace framework

Maity et al. [30] proposed an approach named EnPlace, which aims to determine optimal controller locations in SD-IoT networks to minimize overall energy consumption while maintaining efficient switch-to-controller latency. To achieve this, the authors used a coalition game [31] that partitions the network into multiple sub-networks, each managed by a single active controller. In addition, selected links were deactivated to further reduce energy usage. Unlike existing controller placement schemes, EnPlace considered cost-efficient in-band control traffic and evaluates the traffic matrix based on different activation patterns of IoT devices. Several prior studies also addressed energy management in wireless and SDN-based networks. Tsiropoulou et al. [32] proposed a non-cooperative game-theoretic approach to optimize uplink transmission power and pricing for users. However, their work did not consider SDN environments or the energy consumption of SDN controllers. Other studies, such as those by Giroire et al. [33], Maaloul et al. [34], and Assefa and Özkasap [35], focused on improving energy efficiency in SDN networks by deactivating links and rerouting traffic. These approaches used optimization models or heuristic algorithms to reduce energy consumption while maintaining network performance. However, they did not address controller placement or IoT-specific traffic patterns. In contrast, EnPlace integrated multiple aspects, including SDN, energy management, controller placement, in-band control, and IoT traffic awareness. The simulation parameters included the ATT topology with 25 switches and 100 to 400 IoT devices. Traffic characteristics included packet sizes ranging from 94 to 234 bytes and data rates from 462 to 11,388 bytes/s. The controller capacity varied between 0.02 and 0.03 mfps, while the link capacity is up to 10 Gbps and traffic rates range from 0.0001 to 0.39 Gbps. Energy-related parameters included baseline power consumption of 30 W, a power coefficient of 10 W, and maximum processor and server power of 86 W and 367 W, respectively. The model also considered delay constraints ranging from 0.001 to 1 s to ensure performance requirements.

Table 10
Effect of traffic load ($\Omega = 0.5$) on energy saving.

Number of IoT Devices	Energy saving			
	EnPlace (%)	Greedy (%)	Kaur et al. (%)	GreCo (%)
100	~ 25	~ 27	~ 24	~ 30
200	~ 30	~ 20	~ 17	~ 25
300	~ 35	~ 15	~ 12	~ 23
400	~ 38	~ 12	~ 10	~ 21

Tables 10 and 11 show the energy-saving performance of different approaches under varying network conditions. In Table 10, the effect of traffic load is analyzed by increasing the number of IoT devices from 100 to 400 while keeping $\Omega = 0.5$. The results show that EnPlace consistently improved energy savings as the number of devices increases, reaching about 38% at 400 devices. In contrast, other methods such as Greedy, Kaur et al., and GreCo show lower savings or a decreasing trend as the network load increases. This indicated that EnPlace is more effective in handling higher traffic loads. Table 11 presents the effect of Ω on energy savings with 100 IoT devices. The results show that energy savings increase significantly as Ω increases from 0.2 to 0.8, rising from about 5% to 45%. This suggested that higher values of Ω , which represent increased traffic or system utilization, led to better energy efficiency in the proposed approach. Therefore, the simulation results demonstrated that EnPlace is more energy efficient. Although EnPlace provides substantial improvements in energy efficiency and QoS awareness, its coalition game-based partitioning primarily emphasizes energy consumption, while limited attention is given to controller synchronization overhead, mobility of IoT devices, and real time adaptive controller reconfiguration in highly dynamic smart city environments.

2.10. Heterogeneous traffic aware controller placement using HTF-CP

In the context of disaster management, SD based IoT networks play a crucial role by utilizing real-time data for effective response. However, managing these networks becomes challenging due to heterogeneous traffic and the varying load generated by switches when communicating with controllers. This heterogeneity makes the CPP more critical. To address this issue, Boudi et al. [36] proposed an optimization method called HTF-CP (Heterogeneous Traffic Flow-Based Controller Placement), which was based on the Lidy+ method. The proposed approach optimized controller placement while effectively handling heterogeneous traffic within the network. The authors evaluated the method using simulation datasets from the Internet Topology Zoo and OS3E. The comparison was performed in terms of latency, number of controllers required, and traffic load distribution. Table 12 presents the results for the three cases, highlighting the impact of controller placement on overall network performance. The results show that both maximum and average latency decrease significantly from Case (A) to Case (C), with maximum latency reducing from 1414.21 to 353.55 and average latency from 604.73 to 110.23. This indicates that improved controller placement leads to lower communication delays. However, the load rate showed an opposite trend. The maximum load rate increased from 0.35 in Case (A) to 1.42 in Case (C), while the average load rate rose from 0.45 to 0.78. This suggested that although latency improved, the controllers experienced higher load. Overall, the results highlighted a trade-off between latency and load distribution, where better controller placement reduces delay but increases controller load.

For the simulation, five network topologies from the Internet Topology Zoo [37] were used, namely GridNet, Abilene, GoodNet, PsiNet, and OS3E. The details of these network topologies are presented in Table 13. The maximum load capacity of each switch was assigned randomly in a heterogeneous manner, ranging from 0.04 to 0.4 MFlows per second. It was assumed that a switch could generate up to 0.4 MFlows per second [24]. For simplicity, the entire network was considered a single domain managed by one controller. In the

Table 11
Effect of traffic load (Ω) with 100 devices on energy saving.

Ω (Omega)	Energy Savings (%)
0.2	~ 5
0.4	~ 17
0.6	~ 39
0.8	~ 45

Table 12

Analytical comparison of the three cases.

Metrics	Case (A)	Case (B)	Case (C)
Maximum Latency	1414.21	707.10	353.55
Average Latency	604.73	282.84	110.23
Maximum Load Rate	0.35	0.71	1.42
Average Load Rate	0.45	0.71	0.78

Table 13

Topology information.

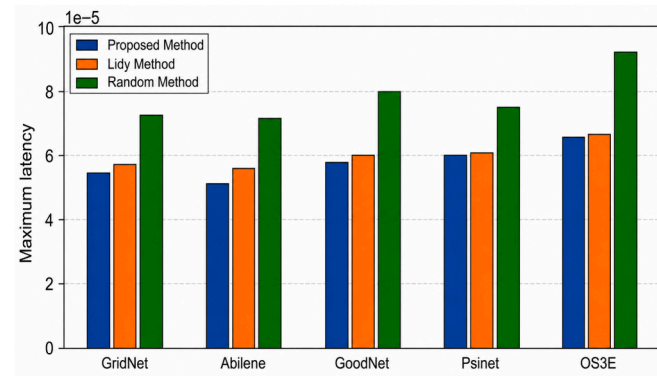
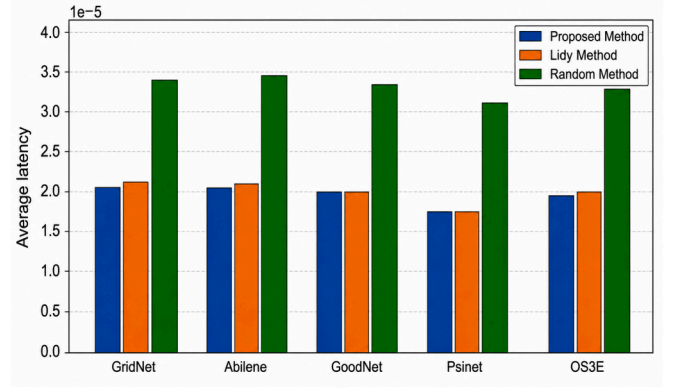
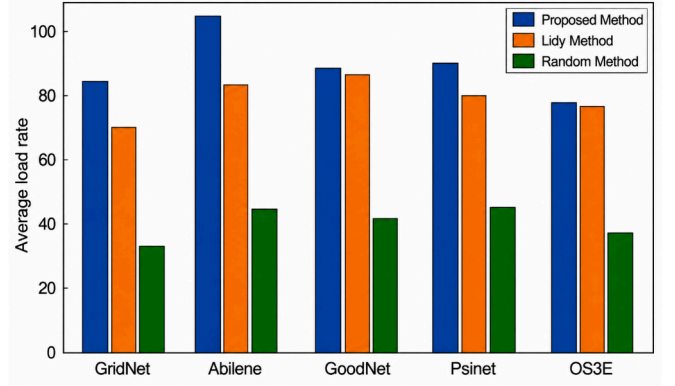
Topology	Number of nodes	Number of edges
GridNet	9	20
Abilene	11	14
GoodNet	17	31
Psinet	24	25
OS3E	34	42

experimental setup, the proposed HTF-CP method was implemented and compared with two other methods these are Lidy and Random method using Python 3.12.0. The simulation was conducted under the assumptions that latency was directly proportional to distance and that the locations of all switches were known.

Fig. 6 illustrates the maximum switch controller latency (WorstSCL), defined as the largest latency between a switch and its assigned controller, computed using the shortest path distance to minimize worst case delay. The results show that the Proposed Method achieves lower maximum latency across all network topologies compared to the Lidy Method and Random Method, indicating more efficient controller placement and reduced worst case communication delay.

Fig. 7 presents the average switch controller latency, defined as the total latency between all switches and their respective controllers divided by the number of switches, reflecting overall communication efficiency. The results indicate that the Proposed Method achieves lower average latency across most network topologies compared to the Lidy Method and Random Method, demonstrating improved controller placement and enhanced network communication efficiency.

Fig. 8 presents the average load rate, analyzed by modeling switches as heterogeneous in terms of traffic load and selecting controller placements to achieve balanced load distribution across the network. The results show that the proposed method provides a more balanced and efficient load distribution across different network topologies compared to the Lidy Method and Random Method, demonstrating improved controller allocation and network performance. Although HTF-CP effectively incorporates heterogeneous switch traffic into controller placement decisions and achieves improved latency performance, its optimization framework remains static, assuming fixed switch weights and controller placement, which may limit adaptability

**Fig. 6.** Switch-controller maximum latency comparison [36].**Fig. 7.** Switch-controller average latency comparison [36].**Fig. 8.** Comparison of average load rate across topologies [36].

in dynamic IoT environments characterized by time varying traffic loads, mobility, and changing network conditions.

2.11. Controller placement using deep Q-learning (DQL)

Ali and Alenazi [38] proposed a dynamic controller placement approach for SD based IoT networks using DQL. In SDN, the controller acts as a central entity responsible for managing network flows, and its placement has a significant impact on communication performance. Suboptimal placement could increase E2E delay due to higher hop counts and greater propagation delays between switches and controllers. To address this challenge, the authors formulated the CPP as a Markov Decision Process (MDP), where network configurations were represented as states and controller placement decisions as actions. A DQL model was employed to approximate the Q-function and learn optimal placement strategies. The reward function was defined as the negative E2E delay, allowing the model to iteratively minimize delay through continuous learning. This dynamic approach enabled the system to adapt to changing network conditions and progressively improve placement decisions. As a result, it reduced communication delays and enhanced overall network performance. For comparison and analysis, the authors validated their approach through experimental evaluation using real-world network topologies, including Interoute, OS3E, USNet, and Abilene. Simulations were conducted in MATLAB and Mininet environments to ensure realistic analysis. The performance of the proposed Deep Q-Network (DQN) based controller placement was compared with the conventional K-Means clustering method, focusing on controller to switch delay. Table 14 indicates that for the 6-cluster configuration, the DQN-based approach consistently outperformed K-Means across all evaluated topologies. In the Abilene topology, the delay was reduced from approximately 300 ms with K-Means to around 200 ms using DQN.

Table 14

DQN vs K-means controller-to-switch delay (6 clusters).

Topology	Proposed DQN (ms)	K-Means (ms)
Abilene	~ 200	~ 300
USNet	~ 2500	~ 3800
OS3E	~ 3000	~ 4800
Interoute	~ 4000	~ 6900

Similarly, in the Interoute network, the delay decreased from about 6900 ms to nearly 4000 ms. These results demonstrated the effectiveness of the proposed method in minimizing communication delay. Moreover, similar performance improvements were observed across other cluster configurations ranging from 6 to 9 clusters, indicating that the model continued to refine controller placement decisions as network partitioning varied. This highlighted the adaptability and robustness of the DQN-based approach in different network scenarios.

Overall, the results confirmed that the proposed method provided more efficient controller placement than traditional methods, resulting in reduced communication delays and enhanced network performance in SD based IoT environments. Despite demonstrating improved adaptability and lower communication delay, the effectiveness of the DQL framework depends heavily on reward design, parameter tuning, and training efficiency, which may limit scalability and convergence stability in highly complex and resource constrained IoT environments.

2.12. Reliability-aware controller placement with backup controllers and error correction

Zangeneh et al. [39] addressed the CPP in IoT-based monitoring systems, particularly in LoRaWAN environments, where reliability challenges arose due to ALOHA-based communication, interference, and channel limitations. The study proposed a dual-solution framework that includes optimal controller placement and optimal backup controller placement to enhance system reliability. In addition, an error control mechanism based on the Kalman filter was integrated to reduce communication errors and improve data accuracy. The first solution focused on optimal controller placement in large-scale IoT monitoring systems, assuming a star-based communication model with sub-network partitioning. This approach ensured efficient communication and minimized disruption in the event of node or link failures. The second solution proposed a backup controller placement strategy, where each network region was assigned a backup controller to maintain system functionality during failures. This was particularly important for real-time applications such as earthquake monitoring. The simulation parameters used for performance analysis are listed in Table 15.

For analysis, the effectiveness of the proposed framework was evaluated through energy consumption and propagation delay analysis under three approaches, namely basic messaging, controller placement messaging, and error correction messaging. As illustrated in Fig. 9, the total energy consumption increased over time for all methods as the number of communication rounds grew. However, the basic messaging approach exhibited the highest energy consumption, reaching nearly 40

Table 15

Simulation parameters.

Parameter	Value
Deployment area	50,000 m
Number of nodes	80–160
Packets transmitted	100–1000
Transmission radius	5000 m
Node distribution	Random
Initial energy	0.01 kJ
Iterations (GWO)	50
Filter size	100
Frequency	868 MHz
Spreading factor	SF12

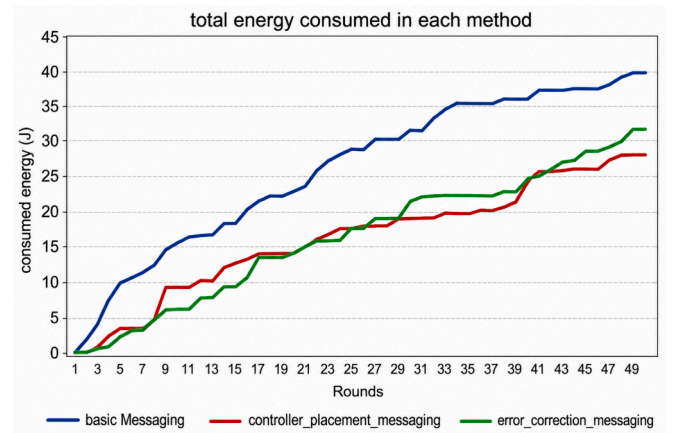


Fig. 9. Comparison of energy consumption across three messaging approaches [39].

J. In contrast, the controller placement messaging approach showed moderate energy usage, while the error correction messaging method demonstrated relatively efficient energy consumption, remaining lower than or comparable to the controller placement approach in later rounds. This indicated that integrating optimized placement and error control mechanisms helped to reduce unnecessary energy expenditure. Similarly, Fig. 10 presents the comparison of propagation delay over time for the three approaches. The results show that the basic messaging method experienced the highest delay, increasing significantly with each round and approaching 100,000 s. The controller placement messaging approach reduced this delay considerably, while the error correction messaging method achieved the lowest propagation delay among all three. These results highlighted that the proposed framework not only improved reliability through backup controller placement and error correction but also enhanced overall network efficiency. It reduced both energy consumption and propagation delay, making it more suitable for large-scale and real-time IoT monitoring applications.

These findings demonstrated that the proposed framework was efficient and reliable for IoT based monitoring applications. However, the proposed method is mainly reliability centric, with limited consideration of scalability in dense heterogeneous IoT deployments, and multi objective trade-offs among latency, energy, and control overhead required for rational controller placement in smart city IoT environment.

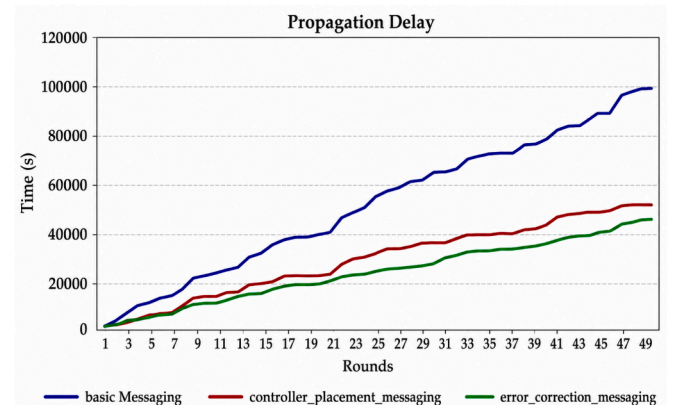


Fig. 10. Comparison of propagation delay across three messaging approaches [39].

2.13. Load balanced controller placement using MDCLB algorithm

Babbar et al. [40] proposed a Multiple Distributed Controller Load Balancing (MDCLB) algorithm to address load imbalance in large-scale SDN based IoT smart city networks. In distributed SDN environments, uneven traffic across controllers led to increased packet loss, higher response time, and degraded performance. The MDCLB algorithm balanced controller load based on CPU utilization, improving overall efficiency. The evaluation was conducted using Mininet and the Ryu controller in a Linux environment and OpenFlow 1.3 for communication. Traffic was generated using the iperf tool, and performance was measured in terms of CPU utilization. As shown in Fig. 11, before applying MDCLB, CPU utilization was unevenly distributed across controllers, ranging from 40% to 60%, indicating load imbalance. After applying the MDCLB algorithm, CPU utilization became distributed across all controllers, reaching 60%, 62%, 61%, 64%, and 63%, respectively, demonstrating effective load redistribution. Overall, the average CPU utilization improved by approximately 5% after load balancing, highlighting a more balanced distribution of load across controllers.

Overall, the study showed that effective load balancing was essential for large-scale SDN based IoT deployments. The MDCLB algorithm provided a practical solution by dynamically redistributing load among controllers, leading to improved resource utilization, reduced response time, and enhanced network performance in smart city environments. Although MDCLB effectively balances controller workload using CPU utilization-based switch migration, controller placement is not jointly optimized with load balancing, and other important factors such as latency, and energy efficiency.

2.14. IoT aware virtual network function placement for Smart City applications

Rafique et al. [41] proposed the IoT aware Virtual Network Function Placement (IVP) problem to address challenges in managing dynamic IoT traffic and limited network resources in smart city environments. The integration of IoT with Virtualized Network Functions (VNFs) created new opportunities for advanced applications but introduced complexity in resource allocation and traffic handling. The authors evaluated different placement algorithms with varying objective functions under both static and dynamic traffic scenarios using realistic IoT traffic data. The study was conducted through a simulation-based case analysis within a smart city environment. The IVP problem was formulated as an optimization task that jointly considered deployment cost, end-to-end delay, and load balancing. The objective function minimized total cost and delay, including propagation, transmission, queuing, and processing delays, while ensuring efficient utilization of

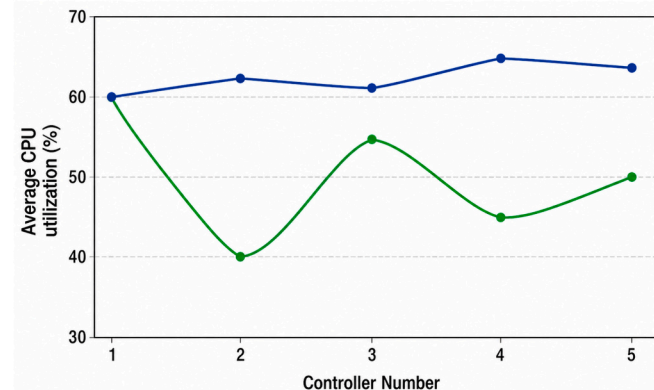


Fig. 11. Average CPU Utilization of Controllers Before and After Load Balancing [40].

edge and cloud resources. The results highlighted the importance of multi-objective optimization approaches, as they effectively balanced conflicting goals and improved overall system performance under dynamic IoT conditions. For analysis and comparison, the authors evaluated multiple placement strategies, including Mixed Integer Programming (MIP), Tabu Search (TS), Round Robin, and a Baseline approach. MIP provided optimal solutions but was computationally complex for large-scale scenarios. Therefore, the TS metaheuristic was used as a practical alternative, offering near-optimal solutions through a local search mechanism with short-term memory. The Round Robin approach was considered a delay-aware method suitable for IoT applications with strict latency requirements, where each service function chain (SFC) was placed on different servers in a sequential manner. To evaluate performance under realistic conditions, a set of smart city use cases was presented in Table 16.

These use cases represented the dynamic, scalable, and heterogeneous nature of IoT traffic across multiple domains, including intelligent transport systems, public safety, smart energy, and smart healthcare. The table included key parameters such as bandwidth, latency, and device density. The applications, such as autonomous traffic, city surveillance, smart grids, and connected ambulance, reflected diverse infrastructure demands and traffic patterns. For example, connected ambulance required very high bandwidth (up to 1000 Mbps) with low latency, while home energy systems operated with minimal bandwidth but tolerated higher delays. This variation allowed comprehensive evaluation of the IVP solution under different network conditions.

Fig. 12 shows the cumulative communication delay for different placement algorithms under static IoT traffic, divided into low data rate and high data rate applications. In Fig. 12(a), for low data rate applications, all algorithms produced similar delays. For road safety, delay was around 12.6 ms across all methods. In city surveillance, delay was the lowest at about 3.6 ms. For home energy, delay was around 13.4 ms, while in smart grid applications, delay increased to about 16 to 18.8 ms. The Baseline approach showed the highest delay of about 18.8 ms, while Round Robin achieved the lowest at about 15.8 ms. In Fig. 12(b), for high data rate applications, the differences were more noticeable. In autonomous traffic, delay was approximately 31 ms for MIP, 32 ms for TS, 39 ms for Baseline, and 26 ms for Round Robin, making Round Robin the most efficient. For structural health, delay ranged from about 19 ms to 24 ms. In connected ambulance, Round Robin achieved the lowest delay at about 13 ms, compared to 25 ms for MIP, 27 ms for TS, and 29 ms for Baseline. For remote monitoring, delays were closer, ranging from about 20 to 22 ms. Overall, Round Robin consistently achieved the lowest communication delay, especially in high data rate and latency-sensitive applications. MIP provided near-optimal performance but was computationally expensive, while TS offered moderate results. The Baseline approach consistently showed the highest delay, indicating lower efficiency. These results highlighted the importance of delay aware placement strategies in optimizing IoT network performance. Despite its effective resource allocation strategy, the IVP framework assumes predefined traffic classes and static optimization objectives, which may limit adaptability to rapidly evolving service demands,

Table 16
Smart city IoT use cases and traffic characteristics.

Application	Bandwidth (Mbps)	Latency (ms)	Device density (/Km ²)
Autonomous Traffic	0.05–10	10	12,000
Road Safety	0.005	10–100	3000
City Surveillance	20–100	10	60
Structural Health	50–100	1–20	>60,000
Home Energy	0.001–0.1	200–300	6000
Smart Grids	0.001–1.5	1–20	6000
Connected Ambulance	1000	10	60
Remote Monitoring	5	250	60,000

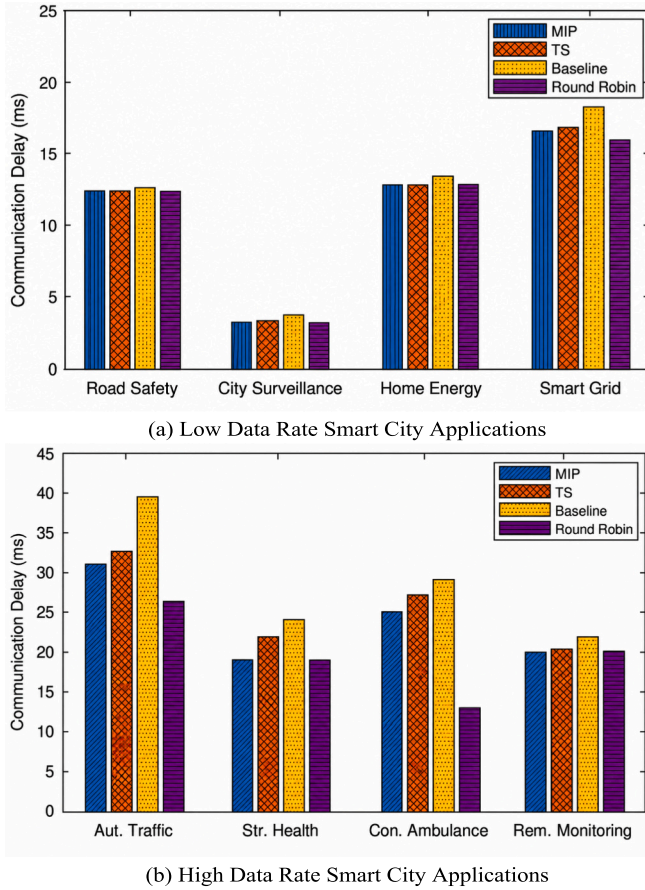


Fig. 12. Communication Delay Analysis for Low and High Data Rate Smart City Applications [41].

mobility patterns, and real time control decisions in heterogeneous smart city IoT networks.

2.15. Controller and sink placement in industrial IoT using PACSA-MSCP

Faragardi et al. [42] proposed the Parallel Ant Colony leveraged by Simulated Annealing for the multiple-sink/controller problem (PACSA-MSCP) algorithm for optimal placement of controllers and sink nodes in industrial IoT systems. With the increasing adoption of IoT in industrial environments, ensuring reliability and timeliness is critical for time-sensitive applications. SDN enables separation of the control and data planes, where controllers and sink nodes must be carefully placed to handle dynamic network conditions while minimizing deployment cost. The proposed approach formulated the placement problem as an optimization task, ensuring that each sensor node is covered by an appropriate number of controllers and sinks to satisfy reliability and delay requirements. PACSA-MSCP combines a parallel max-min ant system with simulated annealing to efficiently explore the solution space. The authors formulated the sink and CPP as an optimization task using an Integer Linear Programming (ILP) model. This formulation enabled the use of standard solvers such as CPLEX and also supported heuristic and metaheuristic approaches for large-scale networks. To efficiently evaluate solutions, the authors defined a fitness function that combined the objective and constraints into a single expression. The fitness function is expressed as follows:

$$\text{Fitness}(X, X') = \text{TotalCost}(X, X') + \beta \left(\sum_{i=1}^9 P_i(X, X') \right)$$
 where TotalCost represents the overall deployment cost, β is a penalty coefficient that guided the search toward valid solutions, and P_i represents penalty functions. The fitness function ensured that the optimization process

minimized both deployment cost and infeasibility. As a result, the algorithm was guided toward solutions that were not only cost-effective but also met reliability and timeliness requirements in industrial IoT systems.

The effectiveness of the proposed approach was analyzed by comparing execution time across different algorithms, as presented in Tables 17 and 18. Table 17, representing small-scale scenarios, GRASP-MSP achieved the lowest execution time at 0.8 min, followed by MSCP at 4.2 min. GA and PMAS showed moderate performance with 9.2 min and 10.3 min, respectively, while SA required 16.5 min. CPLEX had a high computational cost, exceeding 120 min, which limited its practicality even for smaller networks. Table 18, for large-scale scenarios, execution times increased for all algorithms. GRASP-MSCP and MSCP remained the most efficient, with 37.5 min and 39.2 min, respectively. GA and SA required more time at 72.3 min and 66.7 min, while PMAS performed the worst among heuristic methods at 94.3 min. CPLEX again showed extremely high complexity, exceeding 600 min, making it impractical for large-scale deployment. It is worth noting that within 10 runs, the PACSA-MSCP algorithm was able to achieve the same optimal result as CPLEX at least once. Even when executed 10 times, the total computation time was 42 min, calculated as 10×4.2 min, which remained significantly lower than the execution time of CPLEX, exceeding 120 min. This showed that PACSA-MSCP not only provided optimal solutions but also achieved these results with significantly lower computational cost. Therefore, it was a practical and efficient approach for large-scale industrial IoT deployment. Although PACSA-MSCP provides optimal solutions, its hybrid metaheuristic design increases model complexity and depends on predefined candidate locations and parameter tuning, which may reduce scalability and practical deployment flexibility in highly dynamic large scale IoT systems.

3. Comparison with previous model

An earlier study [15] evaluated controller placement in an IoT environment without incorporating building coverage. In that study, the proposed hybrid GA-PSO method was compared with two methods, random placement and K-Means. For a communication range of 1000 m, the optimized method attained a latency of 1459.67 m, which was significantly lower than random placement at 2417.74 m and slightly better than K-Means at 1461.49 m. Similarly, load variance (load balance) was considerably reduced to 45.89 Mbps², compared to 773.61 Mbps² for random placement and 78.63 Mbps² for K-Means, indicating effective load distribution. Packet loss was recorded at 74.25%, which was lower than random placement at 82.50% and K-Means at 76.10%. Energy consumption for the optimized method was 1265.52×10^6 Mbps·m², which was significantly lower than random placement at 4053.33×10^6 Mbps·m², although slightly higher than K-Means at 1201.89×10^6 Mbps·m². Scalability was highest for the optimized approach at 0.000155, compared to 0.000139 for random placement and 0.000152 for K-Means, while redundancy for the optimized method was 2937.16 m, compared to 3406.09 m for random placement and 2984.81 m for K-Means. Fault tolerance at this range was zero for both the proposed and K-Means methods, indicating the absence of backup coverage. When the communication range was increased to 3000 m, packet loss for the optimized method dropped significantly to 2.35%,

Table 17
Small-scale execution time comparison.

Algorithms	Execution Time (min)
MSCP	4.2
GRASP-MSP	0.8
SA	16.5
PMAS	10.3
CPLEX	> 2 h
GA	9.2

Table 18
Large-scale execution time comparison.

Algorithms	Execution time (min)
MSCP	39.2
GRASP-MSP	37.5
SA	66.7
PMAS	94.3
CPLEX	> 10 h
GA	72.3

compared to 28.40% for random placement and 1.40% for K-Means. Meanwhile, fault tolerance improved to 0.60 for the optimized method, compared to 0.48 for random placement and 0.58 for K-Means, demonstrating enhanced network reliability. Despite these findings, the earlier study did not address building coverage in its analysis, even though this was an important factor in urban environments. By integrating building data into the updated model, a more realistic analysis of

controller placement was achieved. Furthermore, in the updated model, building data and three additional methods, namely PSO, GA, and APDQN, were utilized for analysis, and their results were compared to evaluate and validate the effectiveness of the proposed method. The updated model expanded the comparison to five methods, providing broader and more comprehensive validation of the proposed approach. The updated model, including building data and the three additional methods, was presented in [Figs. 13 and 14](#).

For a communication range of 1000 m, the optimized method covered 12,043 buildings out of a total of 42,099, resulting in 28.61% building coverage. This was higher than random placement, which covered 7689 buildings, representing 18.26% coverage. The optimized method also outperformed PSO, GA, K-Means, and APDQN at this communication range. PSO covered 11,190 buildings (26.58%), GA covered 10,834 (25.73%), K-Means covered 11,903 (28.27%), and APDQN covered 10,370 (24.63%). These results indicated that for a communication range of 1000 m, the optimized method maintained

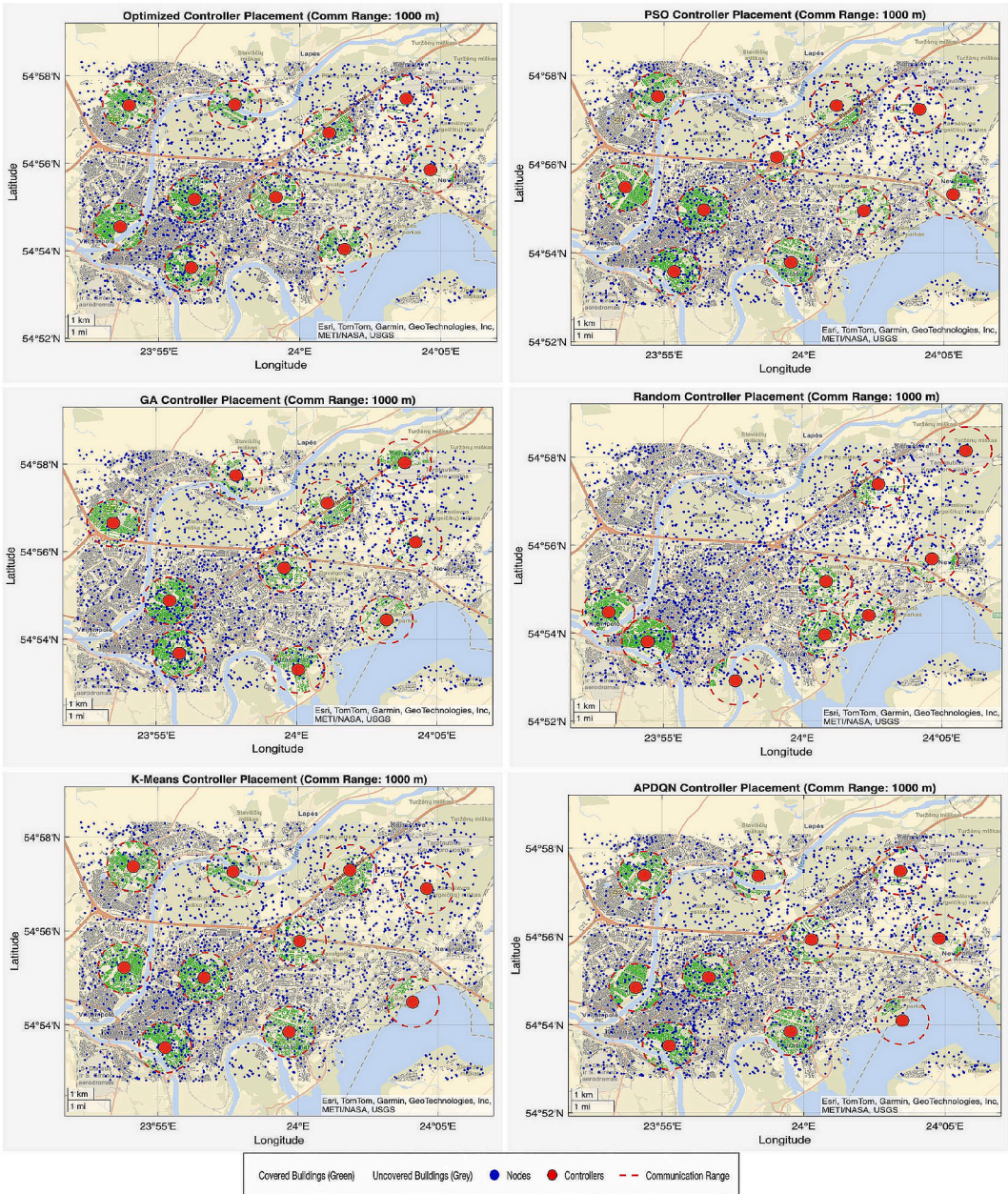


Fig. 13. Visualization of controller placement methods at 1000 m.

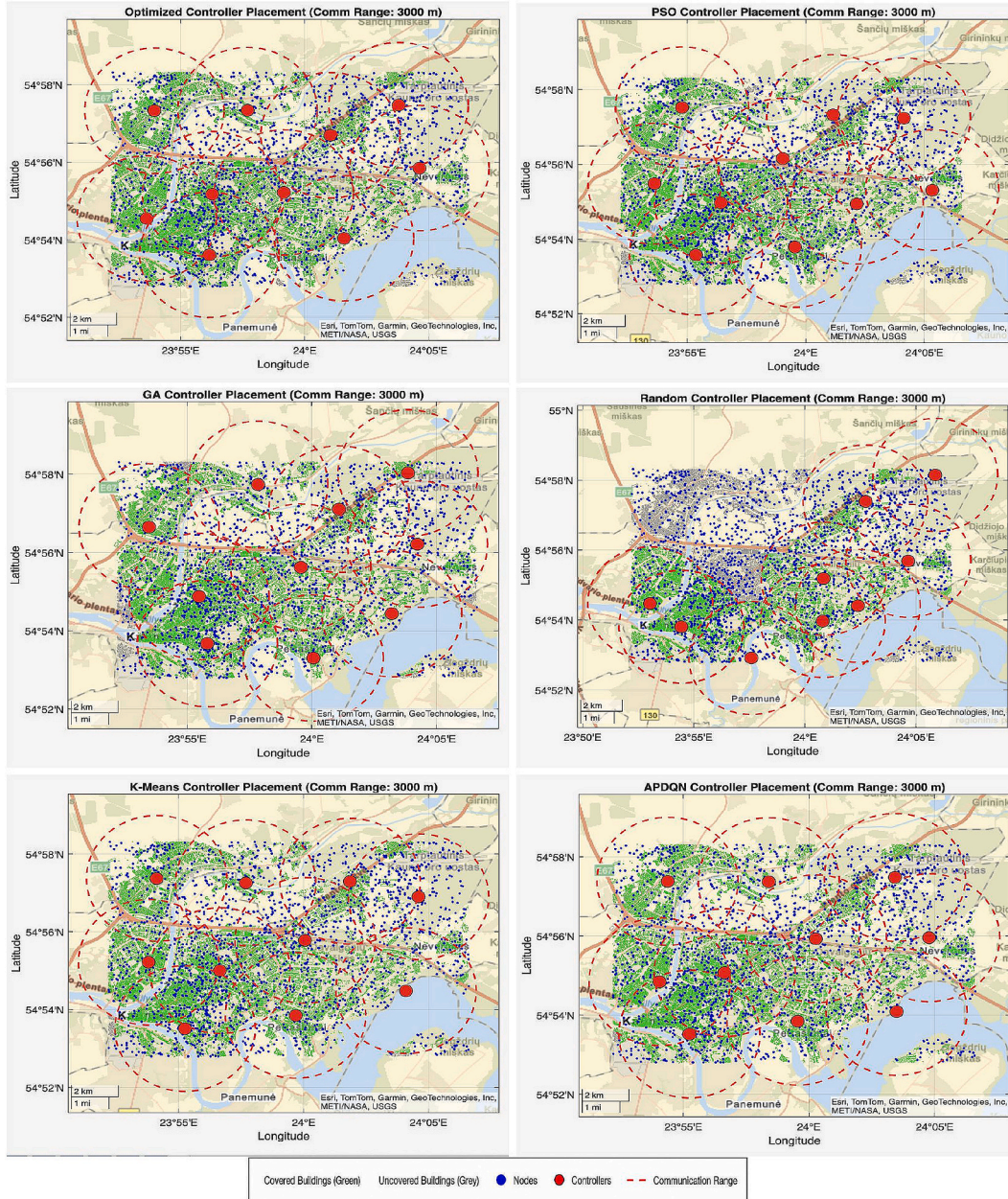


Fig. 14. Visualization of controller placement methods at 1000 m.

effective coverage across a high proportion of both nodes and buildings. When the communication range increased to 3000 m, building coverage improved significantly for all methods because of the wider communication radius. The optimized method covered 41,459 buildings, achieving 98.48% coverage. PSO covered 41,022 buildings, representing 97.44% coverage, while GA covered 39,725 buildings, corresponding to 94.36% coverage. Random placement performed poorly compared to other placement methods, covering only 27,868 buildings with 66.20% coverage. K-Means achieved coverage of 41,605 buildings, corresponding to 98.83%, and APDQN attained the highest coverage with 41,929 buildings, representing 99.60%. The optimized method still performed competitively, with only a marginal difference. Although K-Means and APDQN slightly outperformed the optimized method in terms of maximum building coverage at a 3000 m communication range, the optimized method demonstrated balanced overall performance across multiple metrics. It maintained a more uniform distribution of resources, reduced the risk of congestion, and improved overall

network stability. Therefore, integrating building coverage not only enhanced the realism of the model but also confirmed that the proposed method was more suitable for practical smart city deployments, where both coverage efficiency and resource balance were critical considerations.

4. Proposed work

This study proposes a comprehensive and realistic framework for optimal controller placement in IoT networks by integrating real geographic modeling, traffic-aware node behavior, and a hybrid PSO-GA approach. GA has been widely used for a variety of optimal placement problems across complex network systems [43]. It introduces diversity into the population through crossover and mutation operations [44], which improve exploration of the search space, but it typically suffers from slow convergence. In contrast, PSO is recognized for its ability to adapt to changing scenarios without significantly degrading system

performance [45]. Therefore, the hybrid PSO-GA approach was adopted because it combines the diversity generation capability of GA with the adaptive and efficient optimization characteristics of PSO, resulting in more effective optimization performance. The simulation framework is divided into four main parts: network initialization and deployment, controller placement optimization, performance evaluation, and metric computation. However, metric computation is not treated as an independent part of the simulation; instead, it is integrated with other stages to enable continuous evaluation of network performance. These parts are discussed in detail in the following sections.

Part 1: Network Initialization and Deployment

The first part focuses on constructing a realistic IoT network environment using real-world geographic data from Kaunas city, Lithuania. Shapefiles representing city boundaries, water bodies (lakes and rivers), and building footprints [13] are transformed into a meter-based coordinate system to enable accurate spatial analysis. Node deployment follows a multi-Objective deployment strategy [46]. Approximately 20% of nodes are placed in clustered regions within a radius of 2000 m to simulate dense urban centers, while the remaining 80% are randomly distributed across valid land areas. A validation mechanism ensures that nodes are placed only within valid land regions inside the city boundary. Building footprints are classified into categories such as residential, commercial, industrial, and educational. This classification is used for later evaluation of building-level coverage. To model realistic communication behavior, NB-IoT communication constraints are incorporated. A maximum uplink capacity of 0.16 Mbps [47] is considered. Each node is assigned a baseline traffic load randomly. Additionally, event-driven traffic bursts are introduced for approximately 10% of nodes, while ensuring that the total traffic does not exceed the maximum uplink capacity. All generated data, including node locations, traffic loads, and environmental constraints, are stored for subsequent processing.

Part 2: Optimized Controller Placement Using a Hybrid PSO-GA Algorithm

The second stage performs controller placement optimization using a hybrid PSO-GA algorithm. Each candidate solution represents controller positions in a two-dimensional continuous space, where each controller is defined by its x and y coordinates. The population is initialized using a combination of K-Means clustering (40%), random selection (20%), and additional random candidates (40%) to maintain diversity. The performance of metaheuristic algorithms depends heavily on parameter selection, which is often determined through empirical tuning and trial and error [48]. In the proposed hybrid PSO, GA method, a population size of 50 is used over 1000 generations. The PSO parameters are set to an inertia weight of $w = 0.7$ and acceleration coefficients of $c_1 = 1.5$ and $c_2 = 1.5$. During each iteration, controller positions are updated using velocity and position equations, with velocity limits defined based on city dimensions. Additional logic brings controllers closer to high traffic node clusters, improving load-aware placement. This strategy improves controller placement in high traffic regions. GA operators are applied to improve exploration and avoid local optima. Crossover is performed with a probability of 0.8 by exchanging subsets of controller positions between solutions, while mutation (probability 0.2) [44] randomly reassigns selected controllers to new candidate locations. A diversity reset strategy is also applied every 10 generations, where a portion of the population is reinitialized to avoid premature convergence. To ensure feasibility, all controller positions are snapped to valid outdoor node locations using a snapping mechanism that enforces uniqueness and avoids overlap. This is achieved through distance-based assignment and matching techniques, ensuring that each controller is placed at a distinct and valid position. Additional constraints ensure that controllers remain within the city boundary. The best solutions were retained in each generation, and performance metrics were recorded throughout the optimization process. The optimization process is guided by a multi-objective fitness function that integrates several performance metrics to evaluate the effectiveness of controller placement. The objective function is designed to capture multiple critical aspects of

network performance, including node coverage, packet loss, latency, energy, controller load balance, fault tolerance, scalability, redundancy, and spatial distribution. Each metric is normalized within the range 0 to 1 to ensure consistency and fair contribution to the overall objective. The weighting coefficients used in the multi objective fitness function were empirically selected through iterative experimentation [48]. Multiple weight combinations were evaluated during the development of the proposed framework, and the final weighting coefficients were selected based on the overall optimization performance. The selected weights reflect the relative importance of each performance metric [14]. Node coverage is given the highest priority with a weight of 0.30, followed by packet loss with 0.15. Energy consumption, controller load variance, and fault tolerance are each assigned a weight of 0.10, while redundancy and scalability are assigned weights of 0.09 and 0.08, respectively. Latency and spatial distribution are assigned smaller weights of 0.05 and 0.03. Based on these weights, the fitness function is formulated as:

$$F = 0.30 \times \text{node_coverage} + 0.15 \times \text{packet_loss} + 0.05 \times \text{latency} + 0.10 \times \text{energy} + 0.10 \times \text{controller_load_variance} + 0.10 \times \text{fault_tolerance} + 0.08 \times \text{scalability} + 0.09 \times \text{redundancy} + 0.03 \times \text{spacing}$$

This fitness function ensures a balanced trade-off between coverage, efficiency, reliability, and robustness. The computational complexity of the proposed hybrid PSO-GA algorithm depends on the population size (P), the number of generations (G), the number of IoT nodes (N), and the number of controllers (C). During each generation, every candidate solution is evaluated using the proposed multi objective fitness function, where IoT nodes are assigned to controllers within the communication range and the corresponding network performance metrics are computed. The overall computational complexity of the proposed optimization algorithm is $O(G \times P \times N \times C)$. The computational efficiency of the proposed model was evaluated using a MATLAB implementation. The optimization process required approximately 4 min to complete 1000 generations with a population size of 50 on a MacBook Pro (M2 processor and 8 GB RAM). This runtime demonstrates that the proposed framework is computationally feasible for large scale IoT networks. Finally, the optimized results are visualized on a geographic map, showing nodes, controllers, and communication ranges.

Part 3: Performance Evaluation and Comparison

The third stage evaluates the proposed method against multiple approaches, including PSO, GA, random placement, K-Means, and an APDQN. The proposed algorithm is compared with these methods across multiple metrics, including node coverage, packet loss, latency, energy consumption, load balancing, fault tolerance, backup node availability, redundancy, and scalability. Controller-level analysis is also performed to determine the number of nodes and buildings served by each controller. Building coverage was evaluated by representing buildings as polygon based geographic objects [49] and using centroid locations as representative points for spatial service assessment, enabling realistic coverage estimation across different building types. Results are presented through tables, comparative graphs, convergence curves over 1000 generations, and geographic visualizations showing node distribution, controller placement, buildings and communication ranges. This part ensures a comprehensive and fair comparison of all placement strategies. Overall, the proposed framework presents a robust, scalable, and realistic solution for controller placement in smart city IoT networks by combining hybrid optimization with real-world constraints and comprehensive performance evaluation.

Part 4: Metric Computation and Fitness Evaluation

The final part defines the evaluation framework through a metric computation function that calculates all performance metrics used in the optimization process. Node coverage is defined as the percentage of nodes served within controller communication range and is expressed as percentage (%). It is calculated as $\text{Node Coverage (\%)} = \frac{N_{\text{covered}}}{N_{\text{total}}} \times 100$, here N_{covered} denotes the number of nodes covered by at least one controller and N_{total} represents the total number of deployed nodes.

Similarly, packet loss is computed as the ratio of unserved traffic to total traffic and is expressed as a percentage (%). It is calculated as $\text{Packet loss (\%)} = \frac{T_{\text{unserved}}}{T_{\text{total}}} \times 100$, here T_{unserved} denotes the amount of traffic that is not successfully served, and T_{total} represents the total generated traffic. Latency is computed using the traffic weighted average distance between nodes and their assigned controllers and is expressed in meters (m). It is calculated as $\text{Latency} = \frac{\sum_{i=1}^N d_i \cdot t_i}{\sum_{i=1}^N t_i}$, where d_i denotes the distance between node i and its assigned controller, measured in meters (m), and t_i represents the traffic load associated with node i , measured in Mbps. Energy consumption is modeled as a function of traffic load and the squared transmission distance and is expressed in Mbps-m². It is calculated as $\sum_{i=1}^N t_i \cdot d_i^2$ where t_i denotes the traffic load associated with node i , measured in Mbps, and d_i represents the transmission distance between node i and its assigned controller, measured in meters (m). Moreover, load balancing is measured as the variance of traffic distributed across controllers and is expressed in kbps². It is calculated as $\sigma^2 = \text{Var}(L_j)$, where L_j denotes the traffic handled by controller j , measured in Kbps. Lower variance indicates a more balanced distribution of traffic among

controllers. Fault tolerance is evaluated based on the availability of backup controllers within the communication range of covered nodes and is expressed as a unitless ratio. It is calculated as $\text{Fault Tolerance} = \frac{N_{\text{nodes} \geq 2 \text{ controllers}}}{N_{\text{total covered nodes}}}$, where $N_{\text{nodes} \geq 2 \text{ controllers}}$ denotes the number of covered nodes that are connected to at least two controllers, and $N_{\text{total covered nodes}}$ represents the total number of covered nodes. Higher values indicate greater resilience against controller failures. Redundancy is measured as the average distance to the second nearest controller within the communication range and is expressed in meters (m). It is calculated as $\text{Redundancy} = \frac{1}{|R|} \sum_{i \in R} d_{i,2}$, where R denotes the set of nodes connected to at least two controllers, $|R|$ represents the total number of such nodes, and $d_{i,2}$ is the distance between node i and its second nearest controller, measured in meters (m). Lower values indicate better backup connectivity and improved redundancy. Scalability is evaluated as the product of the spare capacity ratio and the node coverage ratio and is expressed as a unitless ratio. It is calculated as $\text{Scalability} = \left(\frac{\sum H_j}{T_{\text{served}}}\right) \cdot \left(\frac{N_{\text{covered}}}{N_{\text{total}}}\right)$, where H_j denotes the available headroom of controller j , measured in Mbps, T_{served} represents the total served

```

===== Part 1: Network Generation =====
1. INPUT numNodes, numControllers, commRange
2. LOAD cityBoundary, waterPolygons, buildings
3. CONVERT coordinates to meters; classify buildings
4. DEFINE clusterCenters; relocate invalid centers
5. GENERATE clustered + random nodes (valid area only)
6. IDENTIFY outdoorNodes
7. ASSIGN baseTraffic + eventTraffic (10% nodes)
8. COMPUTE buildingCentroids;
8. SAVE networkData;

===== Part 2: PSO-GA Optimization =====
9. LOAD networkData;
10. INITIALIZE population (K-Means and random);
11. INITIALIZE velocity, pBest, gBest
12. FOR generation = 1 to maxGen DO
    EVALUATE fitness; UPDATE pBest, gBest
    APPLY PSO update (velocity and position)
    APPLY GA (crossover and mutation); RESET population
13. Run APDQN
14. save optimized controllers and APDQN controllers

===== Part 3: Evaluation =====
15. Run baseline methods (PSO, GA, Random, K-Means)
16. COMPUTE metrics (node coverage, packet loss, latency, energy, load balance, fault
    tolerance, redundancy, backup node count scalability)
17. COMPUTE building coverage statistics
18. DISPLAY tables and plots
19. VISUALIZE and compare results
20. OUTPUT final metrics results

===== Part 4: Metrics =====
21. COMPUTE distances (nodes ↔ controllers)
22. ASSIGN nodes; IDENTIFY nodes within range
End

```

Fig. 15. Pseudo-code.

traffic in the network, measured in Mbps, N_{covered} is the number of nodes within the communication range of at least one controller, and N_{total} is the total number of nodes in the network. Higher values indicate greater capacity for accommodating additional traffic demand while maintaining network coverage. Backup Nodes Count is defined as the number of nodes that have at least two controllers within the communication range and is expressed as an integer count. It is calculated as Backup Nodes Count = $\sum_{i=1}^N 1(\sum_{j=1}^C 1(d_{ij} \leq R) \geq 2)$, where N denotes the total number of nodes, C represents the total number of controllers, d_{ij} is the distance between node i and controller j , measured in meters (m), and R is the controller communication range, measured in meters (m). The term $\sum_{j=1}^C 1(d_{ij} \leq R) \geq 2$ indicates whether node i has at least two controllers within its communication range, thereby representing the availability of backup connectivity. Higher values indicate greater network resilience and redundancy. The pseudo-code in Fig. 15 provides an overview of the main computational steps followed in the MATLAB implementation.

Fig. 16 illustrates the workflow of the proposed hybrid PSO-GA method for optimal controller placement in an IoT based smart city network. The process begins with parameter initialization, network data loading, and generation of an initial population of candidate solutions. To ensure reproducibility and maintain consistent experimental conditions across all evaluated controller placement methods, a fixed random seed (rng(42)) was used throughout the optimization process. In the PSO stage, the fitness of all particles is first evaluated using the defined objective function, which measures the quality of each controller placement based on network performance metrics. Each particle then compares its current fitness value with its personal best solution (pBest), and if the new solution is better, the personal best is updated. The algorithm also updates the global best solution (gBest), which represents the best controller placement found among all particles. Based on these updates, particle velocity and position are adjusted to search for improved controller locations. In the GA stage, weaker solutions are selected for crossover and mutation to generate better offspring, while elitism preserves the best solution, and a diversity reset mechanism helps avoid premature convergence and local optima. A constraint handling step ensures that controllers are placed only on valid outdoor

nodes. This iterative optimization continues until the termination condition is met, resulting in the optimal controller placement for the smart city IoT network.

5. Results and discussion

The first phase, network initialization and deployment, focuses on constructing the IoT environment. Fig. 17 illustrates the deployment of an NB-IoT network over the geographical area of Kaunas city (Lithuania), where a total of 2000 IoT nodes are distributed within the city boundary. The black polygon represents the administrative boundary of Kaunas city (Lithuania), while the blue points indicate the spatial locations of the deployed IoT nodes. The deployment of nodes follows a hybrid strategy that combines clustered and random distributions to reflect realistic urban scenarios. Approximately 20% of the nodes are placed in 5 predefined clusters with a radius of 2000 m, representing high-density regions such as commercial centers and other busy zones. The remaining 80% of nodes are randomly distributed across the city to ensure uniform coverage. Nodes are placed in both outdoor and indoor environments following spatial constraints derived from geographic data, where deployment is limited to the city boundary, not allowed in water areas, and permitted within building footprints for indoor placement. The simulation incorporates NB-IoT communication characteristics, where each node generates low data rate traffic consistent with typical sensor behavior. Baseline traffic is modeled as low-rate periodic transmissions, while 10% of the nodes are randomly selected as event driven devices that generate high traffic. However, all traffic loads are constrained by the maximum uplink capacity of NB-IoT, ensuring realistic network conditions. Communication ranges of 1000 m and 3000 m are considered, which align with typical urban NB-IoT coverage capabilities and serve as key parameters for subsequent controller placement and connectivity analysis. The geographic environment is constructed using real shapefiles, including city boundaries, water bodies, and building footprints. This enables accurate spatial validation during node generation and enhances the realism of the deployment. Additionally, building data are classified according to functional categories such as residential, commercial, industrial, educational institutes, hospitals, and other facilities, and their centroids are computed for use in later stages of network optimization. Overall, this simulation framework provides a realistic representation of large-scale IoT deployment in an urban environment, and establishes a foundation for evaluating controller placement strategies, coverage performance, and network efficiency in later stages of analysis.

The second phase of the simulation focuses on the optimal placement of controllers within the generated IoT network using the hybrid PSO-

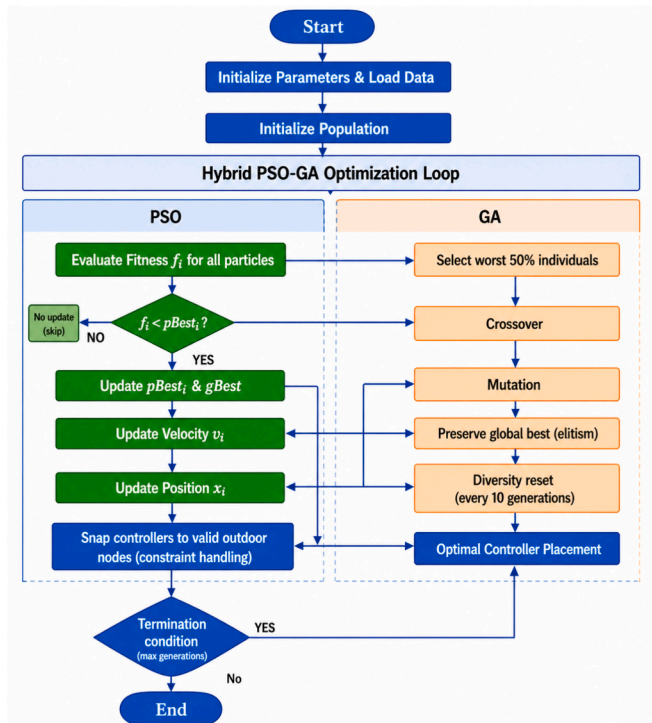


Fig. 16. Flowchart of the Hybrid PSO-GA algorithm.

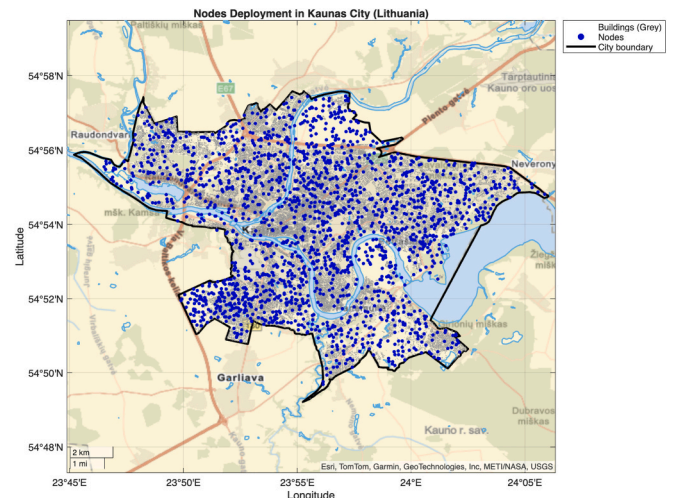


Fig. 17. Node deployment in Kaunas city (Lithuania).

GA framework. This part of the simulation begins by loading the network data generated in Part 1, including node positions, traffic loads, communication range, and geographic constraints such as city boundaries, water bodies, and building layouts. Only outdoor nodes are considered candidate locations for controller placement to ensure practical deployment feasibility. Additionally, building data are analyzed to determine the distribution of building classes, which provides contextual insight into the urban environment and supports realistic interpretation of the deployment scenario. The proposed algorithm aims to determine the optimal controller positions such that multiple performance objectives are simultaneously satisfied. The controller placement search is initialized using PSO, where particles update their positions based on personal and global best experiences. To enhance exploration and avoid premature convergence, GA operators, including crossover, mutation, and periodic diversity reset, are applied. Furthermore, a snapping mechanism ensures that all controller positions remain mapped to valid outdoor node locations, preserving deployment realism. The fitness function is designed as a weighted multi-objective model that integrates several key network performance metrics. These include node coverage, packet loss, latency, energy consumption, controller load balancing, fault tolerance, scalability, backup node count, and redundancy. Each metric is normalized and empirically assigned a specific weight, allowing the algorithm to achieve a balanced trade-off across multiple performance objectives. During the optimization process, these metrics are recorded at each iteration, enabling detailed performance and convergence analysis. The convergence behavior of the proposed PSO-GA algorithm is illustrated in Fig. 18 for communication ranges of 1000 m and 3000 m, respectively. In both cases, the fitness value shows a rapid decline during the initial iterations, followed by gradual stabilization. Specifically, the fitness decreases sharply in the early generations, indicating effective global exploration, and then gradually converges toward a near optimal value as the search space is refined. This behavior confirms the stability and effectiveness of the proposed hybrid optimization approach across different communication ranges.

Fig. 19 presents the evolution of the performance metrics across iterations for a communication range of 1000 m. Node coverage improves steadily, indicating enhanced connectivity across the network, while packet loss decreases, reflecting improved communication reliability. Latency remains relatively stable with minor fluctuations, suggesting that delay performance is maintained within acceptable limits. Energy consumption shows a decreasing trend, highlighting improved

efficiency in controller placement. Load balancing fluctuates moderately as traffic is redistributed among controllers, while fault tolerance slightly decreases, indicating a trade-off between redundancy and efficiency. Backup node count stabilizes after initial fluctuations, and redundancy increases gradually, demonstrating improved spatial distribution of controllers. Scalability also shows a consistent upward trend, indicating improved adaptability of the network.

Fig. 20 presents the evolution of the performance metrics over iterations for a communication range of 3000 m. Node coverage remains relatively stable with minor fluctuations, indicating consistent connectivity across the network, while packet loss decreases, reflecting improved communication reliability. Latency fluctuates over iterations but shows overall improvement, suggesting that delay performance is maintained within acceptable limits. Energy consumption exhibits a decreasing trend, highlighting improved efficiency in controller placement. Load balancing varies significantly during the initial iterations due to uneven traffic distribution among controllers, and then stabilizes as traffic is gradually redistributed. Fault tolerance remains relatively stable with slight improvements, indicating a modest enhancement in network reliability. Backup node count stabilizes after the initial iterations, while redundancy decreases gradually, demonstrating a more efficient spatial distribution of controllers. Scalability remains nearly constant with minor variations, indicating limited adaptability of the network.

Figs. 21 and 22 illustrate the normalized metrics for communication ranges of 1000 m and 3000 m, respectively, where all performance metrics are scaled between 0 and 1 for fair comparison. To ensure a fair evaluation, the metrics are normalized and analyzed collectively, and the optimal iteration is selected based on a balance between high overall performance and minimal variation among the metrics.

Figs. 23 and 24 provide a combined visualization of all normalized metrics for communication ranges of 1000 m and 3000 m, respectively, clearly showing convergence trends across multiple objectives. Although some fluctuations occur during the early stages due to exploration mechanisms, the metrics stabilize in later iterations, confirming that the algorithm converges to a balanced and efficient solution.

The optimized controller placement is illustrated in Figs. 25 and 26, where the spatial distribution of controllers across Kaunas city (Lithuania) is shown along with their coverage regions for communication ranges of 1000 m and 3000 m, respectively. The controllers, labeled C1 to C10, are strategically positioned to cover dense clusters of IoT nodes while maintaining balanced spacing across the city. The

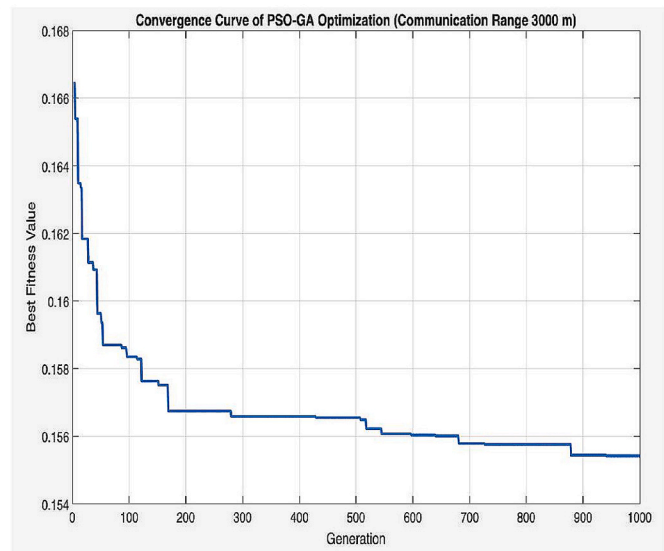
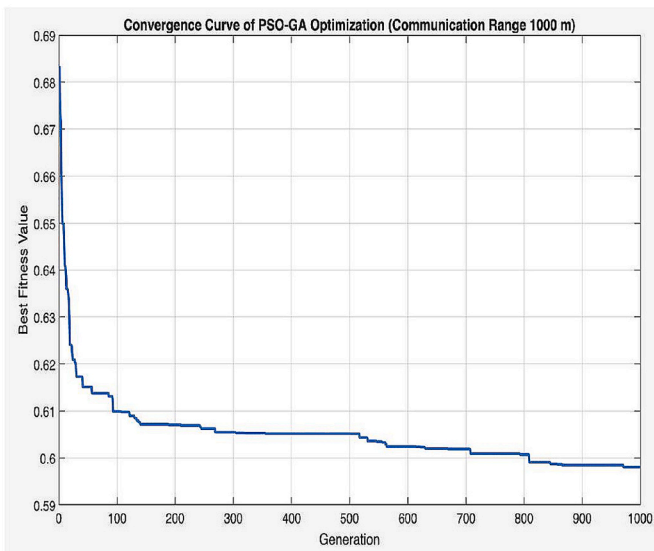


Fig. 18. Convergence behavior of the Proposed PSO-GA algorithm for a communication ranges of 1000 and 3000 m.

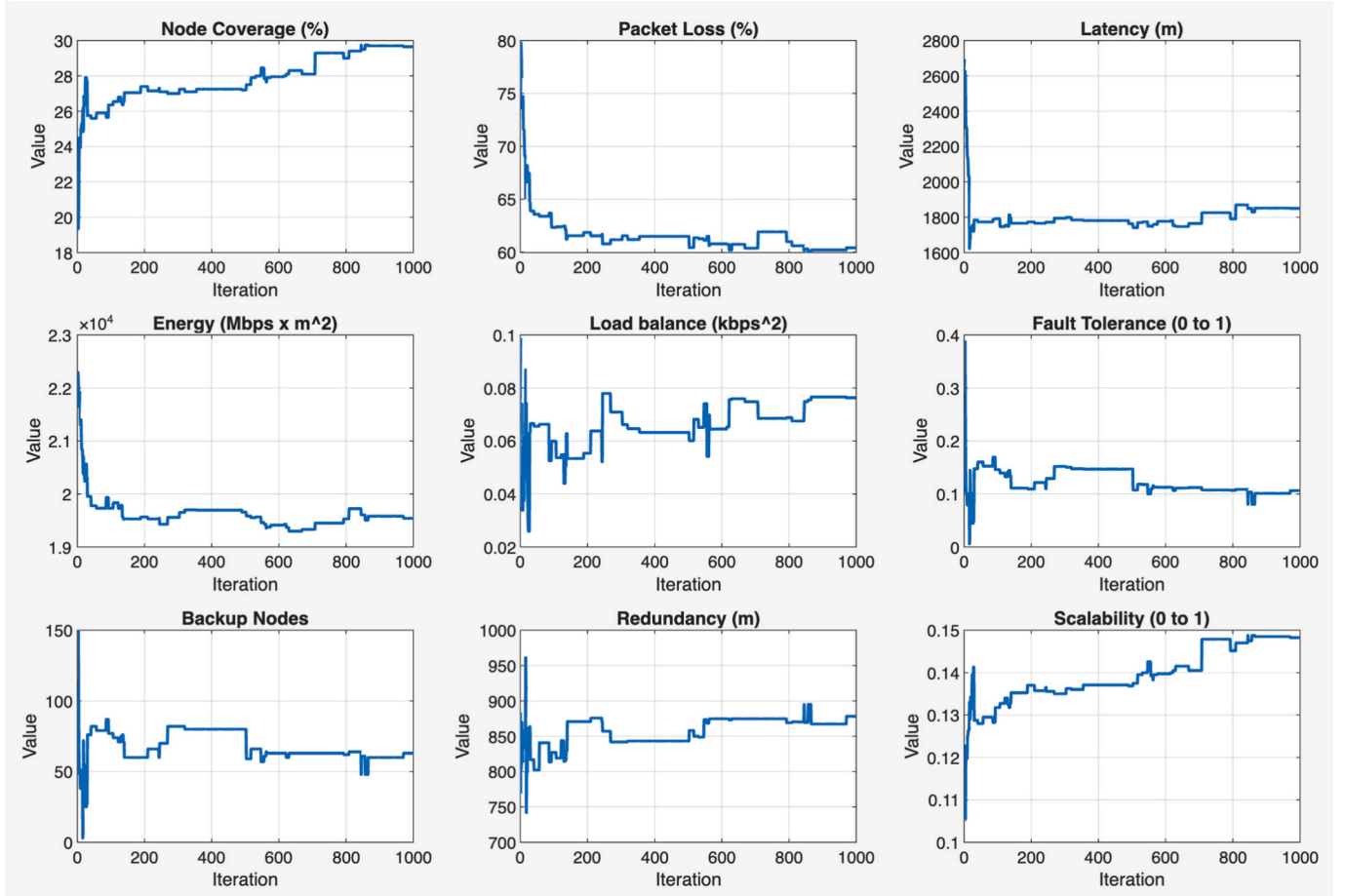


Fig. 19. Evolution of performance metrics for 1000 m communication range.

dashed circular regions represent the communication ranges, demonstrating effective coverage overlap in high-density areas and reduced coverage gaps compared to lower range scenarios. In the 1000 m range, controllers are more localized, requiring careful placement to ensure maximum coverage, whereas in the 3000 m range, wider coverage areas allow fewer coverage gaps and improved connectivity across distant regions. It is observed that controllers are concentrated in regions with higher node density while still ensuring coverage of peripheral areas. This spatial arrangement reduces uncovered nodes and enhances overall network performance. The visualization indicates that the proposed PSO-GA optimization results in well distributed controller placement across different communication ranges while maintaining practical deployment constraints.

In the third phase of the simulation, the proposed method is compared with other controller placement approaches, including PSO, GA, random placement, K-Means, and APDQN. This phase utilizes the network data generated in Part 1 and the optimized controller locations obtained in Part 2. To ensure a fair and consistent comparison, all controller sets are restricted to valid outdoor node locations under identical conditions. For each method, a comprehensive set of performance metrics is computed using a unified evaluation framework. These metrics include node coverage, packet loss, latency, energy consumption, load balancing, fault tolerance, scalability, backup node count, and redundancy. The fitness value for each $method$ is computed using the same weighted multi-objective formulation applied in the proposed method, ensuring consistency in the evaluation process. As shown in Table 19, the performance metrics for a communication range of 1000 m indicate that the optimized method provides one of the most balanced overall performances compared to PSO, GA, random, K-Means, and APDQN

methods. The optimized method achieves the highest node coverage of 29.650%, the lowest packet loss of 60.412%, achieving the lowest energy consumption value of $19,540.655 \text{ Mbps} \times \text{m}^2$, the highest fault tolerance of 0.106, the highest number of backup nodes of 63, and the highest scalability ratio of 0.148, demonstrating better overall performance. In terms of latency, the K-Means method performs best with the lowest value of 1472.991 m, followed closely by PSO with 1492.909 m. For load balancing, the GA method shows the best performance with the lowest value of 0.020. In terms of redundancy backup distance, the GA method also achieves the lowest value of 848.684 m. Overall, while methods such as PSO, GA, random, K-Means, and APDQN perform well in specific metrics, the proposed optimized method provides the most balanced performance across multiple evaluation criteria. Similarly, as shown in Table 20, the performance metrics for a communication range of 3000 m indicate that the optimized method provides one of the most balanced overall performances compared to PSO, GA, random, K-Means, and APDQN methods. In terms of node coverage, the K-Means method performs best with the highest value of 99.550%, followed by the optimized method with 98.550%. The optimized method achieves the lowest latency of 1411.863 m and lower energy consumption of $61,032.320 \text{ Mbps} \times \text{m}^2$ compared to other methods. In terms of packet loss, the K-Means method performs best with the lowest value of 0.098%, followed by the optimized method with 0.710%. For load balancing, the optimized method shows better performance with a lower value of 0.086 kbps^2 compared to other methods. In fault tolerance, the GA method performs best with the highest value of 0.651, followed by PSO with 0.604 and the optimized method with 0.595. The GA method also achieves the highest number of backup nodes with 1253, while random and APDQN perform poorly in this metric, with values of 753

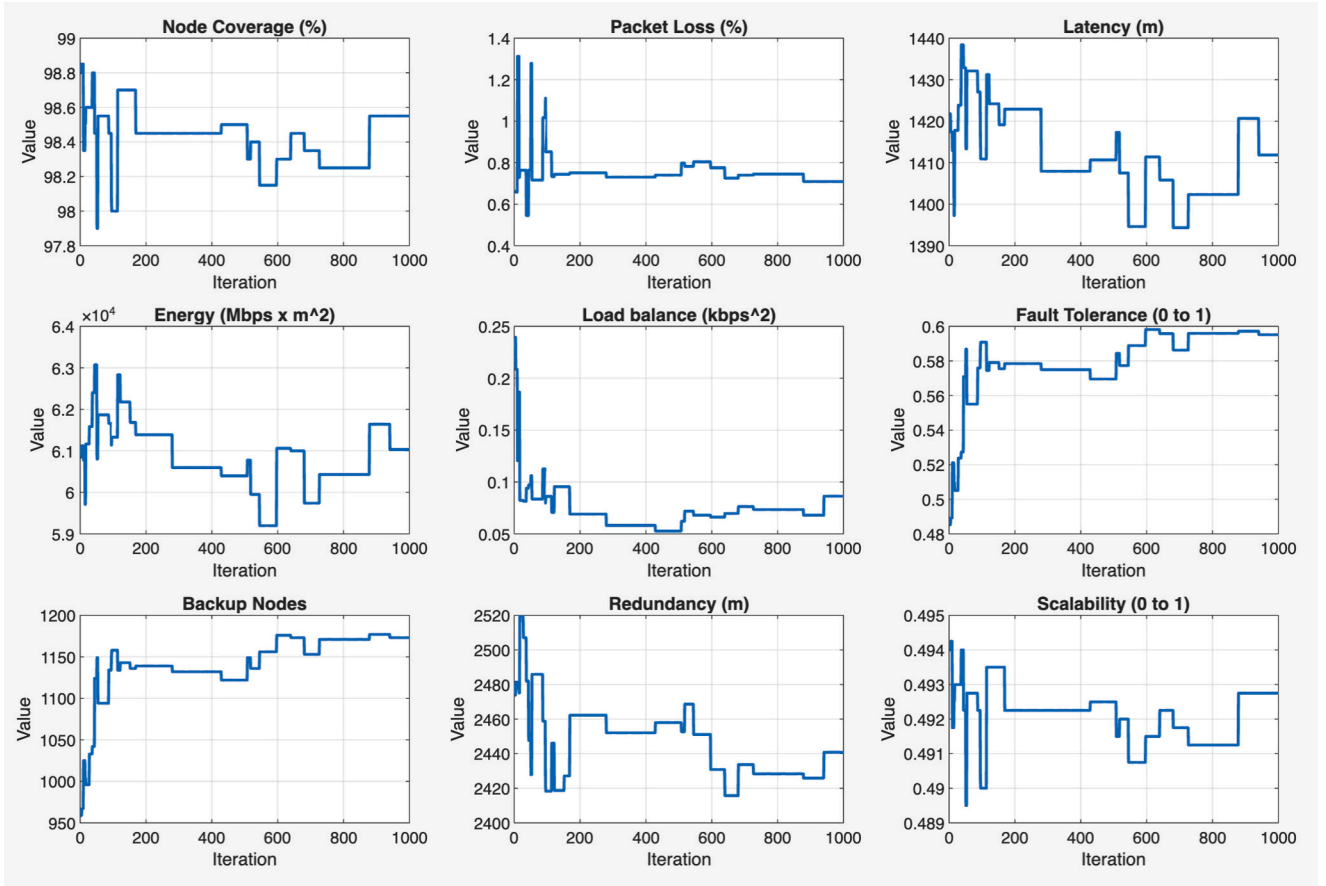


Fig. 20. Evolution of performance metrics for 3000 m communication range.

and 702, respectively. For redundancy backup distance, the random method shows the lowest value of 2116.165 m, indicating closer backup placement. For scalability, the APDQN method performs best with the highest value of 0.511, followed by K-Means with 0.498 and the optimized method with 0.493. Overall, while methods such as PSO, GA, random, K-Means, and APDQN perform better in specific metrics, the proposed optimized method provides a balanced and competitive performance across multiple evaluation criteria. Figs. 27 and 28 further illustrate these comparisons through bar chart visualizations for communication ranges of 1000 m and 3000 m, respectively, providing a clear comparative analysis of the performance of all methods across the evaluated metrics, where the best performing method for each metric is highlighted in green.

In addition to the overall performance metrics, a detailed analysis of per-controller node coverage is presented in Tables 21 and 22, and Figs. 29 and 30 for communication ranges of 1000 m and 3000 m, respectively. Here, C represents the controllers (C1 to C10). For the 1000 m range, the optimized method shows a balanced distribution of covered nodes across controllers, with values of 41 nodes at C1, 61 nodes at C2, 78 nodes at C3, and so on up to 67 nodes at C9 and 63 nodes at C10. In comparison, PSO shows a higher concentration of covered nodes at certain controllers, such as 104 nodes at C2, while GA reaches 75 nodes at C4. The APDQN method shows significant variation, with 104 nodes at C5 and 20 nodes at C1. Random and K-Means also exhibit uneven distributions, with lower values in several controllers, such as 15 nodes at C5 for random and 32 nodes at C1 for K-Means. For the 3000 m range, overall coverage per controller increases significantly due to the increase in communication range. The optimized method maintains consistent coverage, with 168 nodes at C1, 233 nodes at C2, 217 nodes at C5, and 229 nodes at C9. PSO reaches higher values at some controllers, such as 210 nodes at C1 and 229 nodes at C2, while GA achieves 222

nodes at C5 and 218 nodes at C10. The APDQN method shows high coverage at certain controllers, with 313 nodes at C7 and 291 nodes at C10, along with lower values such as 67 nodes at C2, indicating an imbalance in load distribution. Similarly, random and K-Means exhibit large variations, with random showing 278 nodes at C10 and 16 nodes at C5, and K-Means reaching 269 nodes at C2 but lower values at other controllers. These results indicate that while some methods achieve higher node coverage at specific controllers, the proposed optimized method maintains a more uniform distribution of nodes across all controllers in both 1000 m and 3000 m scenarios, which results in better load balancing and improved overall network stability.

To further evaluate the effectiveness of the proposed approach, building coverage is assessed using real geographic data from Kaunas city (Lithuania). The results are presented in Tables 23 and 24 for communication ranges of 1000 m and 3000 m. For the 1000 m range, the optimized method covers 13,720 buildings out of a total of 54,847, resulting in a building coverage of 25.01%. This is slightly lower than the GA method, which achieves the highest coverage of 25.21% with 13,830 covered buildings. PSO and K-Means show moderate coverage with 10,883 and 10,589 covered buildings, respectively, while APDQN covers 11,603 buildings. The Random method performs the worst, covering only 7831 buildings with a coverage of 14.27%. Across individual controllers, the optimized method maintains a relatively balanced distribution, with 769 buildings at C1, 1786 at C3, 2252 at C5, and 2505 at C7. In contrast, PSO covers more buildings at specific controllers, with 2298 at C2 and 2105 at C3. The GA method achieves coverage of 1983 buildings at C4 and 2240 at C7. K-Means demonstrates high variation, with 2255 buildings at C2 and only 36 at C5. Similarly, APDQN shows uneven per-controller coverage, with 2298 buildings at C5 and only 60 at C1. For the 3000 m range, building coverage increases significantly due to the larger communication range. The K-Means

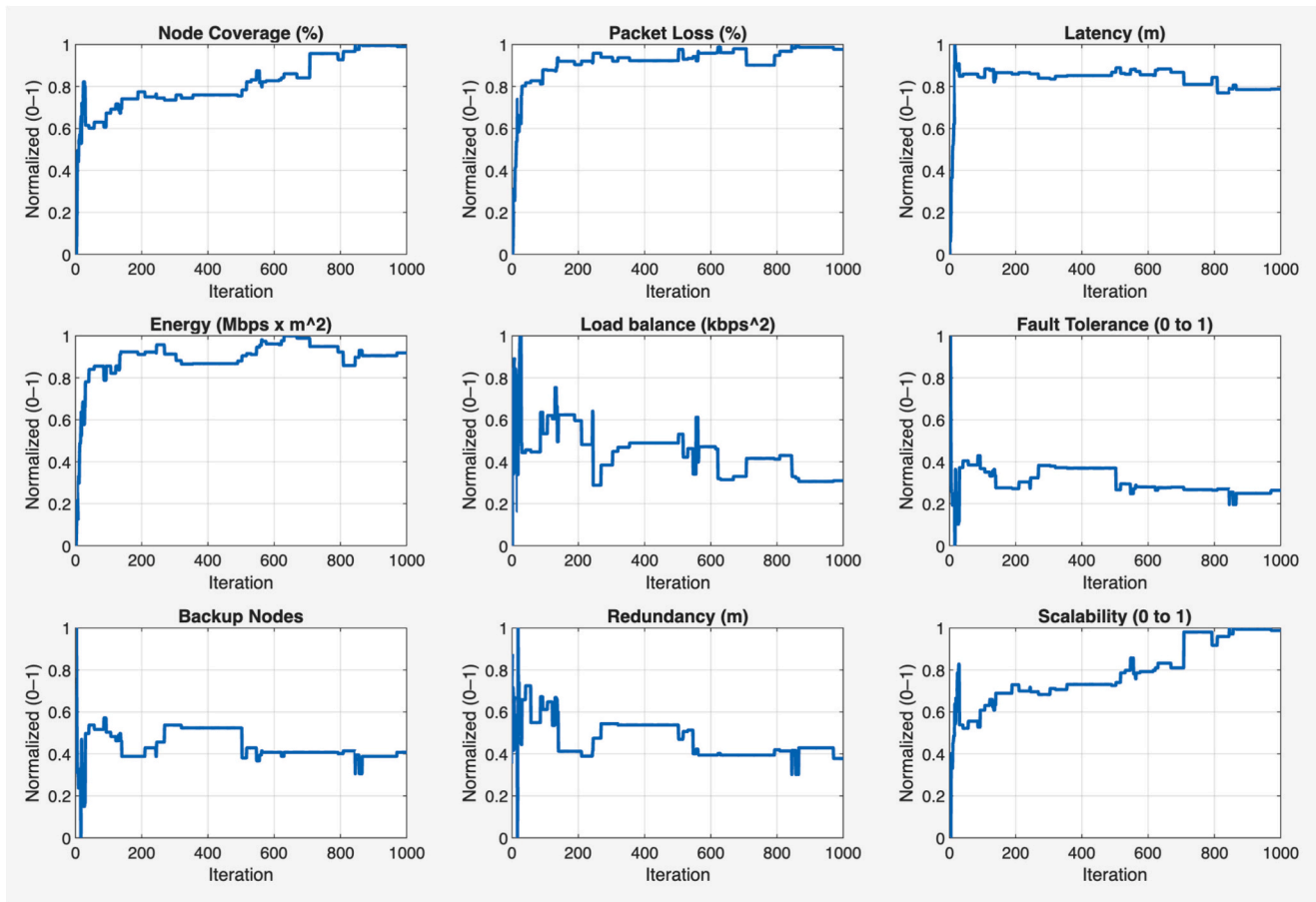


Fig. 21. Normalized performance metrics across iterations for a 1000 m communication range.

method achieves the highest building coverage of 99.31%, covering 54,472 buildings, followed by the optimized method with 53,640 covered buildings and 97.79% coverage. PSO and GA cover 53,586 and 53,233 buildings, respectively, while APDQN covers 52,967 buildings. The Random method again shows the lowest performance, covering 39,713 buildings with 72.40% coverage. Across controllers, the optimized method maintains a steady distribution, with 5364 buildings at C1, 7315 at C3, 9278 at C5, and 4793 at C7. The PSO method shows higher values at specific controllers, with 10,591 buildings at C3 and 5852 at C6, while GA records 7362 at C3 and 8990 at C6. The APDQN method shows high variation, with 10,196 buildings at C7 compared to 1867 at C6. K-Means achieves high building coverage, with 8416 at C3 and 6721 at C7. The Random method shows an uneven distribution, with 7653 buildings at C3 and 1217 at C5. Tables 25 and 26 further present the classification of building types across different methods and communication ranges. It shows that the optimized method maintains a more balanced and consistent distribution of building coverage among controllers in both 1000 m and 3000 m scenarios, supporting improved network performance.

Overall, the results demonstrate that the proposed method achieves one of the most balanced performances among the evaluated placement approaches by effectively balancing coverage, efficiency, robustness, and scalability, making it well suited for large-scale IoT network deployment in smart city environments. To further validate these findings, Figs. 31 and 32 provide a detailed visual comparison of controller placement strategies under two communication ranges, 1000 m and 3000 m, including the proposed optimized method, PSO, GA, random placement, K-Means, and APDQN. In each figure, blue dots represent IoT nodes, green areas indicate covered buildings, grey areas indicate uncovered buildings, red circles denote the communication range of each

controller, squares represent controllers labeled C1 to C10, and the black boundary outlines the Kaunas city (Lithuania) area. For a communication range of 1000 m, the coverage of nodes and buildings remains limited due to the large geographical area, resulting in fragmented coverage and large uncovered regions. The optimized approach achieves more balanced spatial distribution, covering 29.65% of nodes, which is higher than PSO, GA, random placement, K-Means, and APDQN, which provide moderate coverage, while random placement performs the worst with only 16.35% coverage. When the communication range increases to 3000 m, overall coverage improves substantially across all methods, with most nodes becoming covered and only minimal uncovered regions remaining. Consequently, differences between algorithms in terms of coverage diminish, as even random placement achieves 75.5% node coverage. However, it does not perform well in other performance metrics, and overall, the proposed method continues to provide balanced and competitive performance across multiple metrics by maintaining better efficiency and balanced performance parameters. The increased communication range also leads to considerable overlap among controller coverage areas. Despite this, the optimized method maintains a more balanced and efficient distribution across the Kaunas city area. These observations demonstrate that controller placement remains a critical factor across different communication ranges, because effective controller placement significantly enhances overall network performance.

Compared with related work, the proposed hybrid PSO-GA controller placement method demonstrates significant improvements across multiple performance metrics at both 1000 m and 3000 m communication ranges, showing effectiveness in both limited and extended coverage scenarios. Keshari et al. [17] reported that the proposed DEWO method achieved latency improvements of 7.82%, 20.25%, and 9.27% over PSO

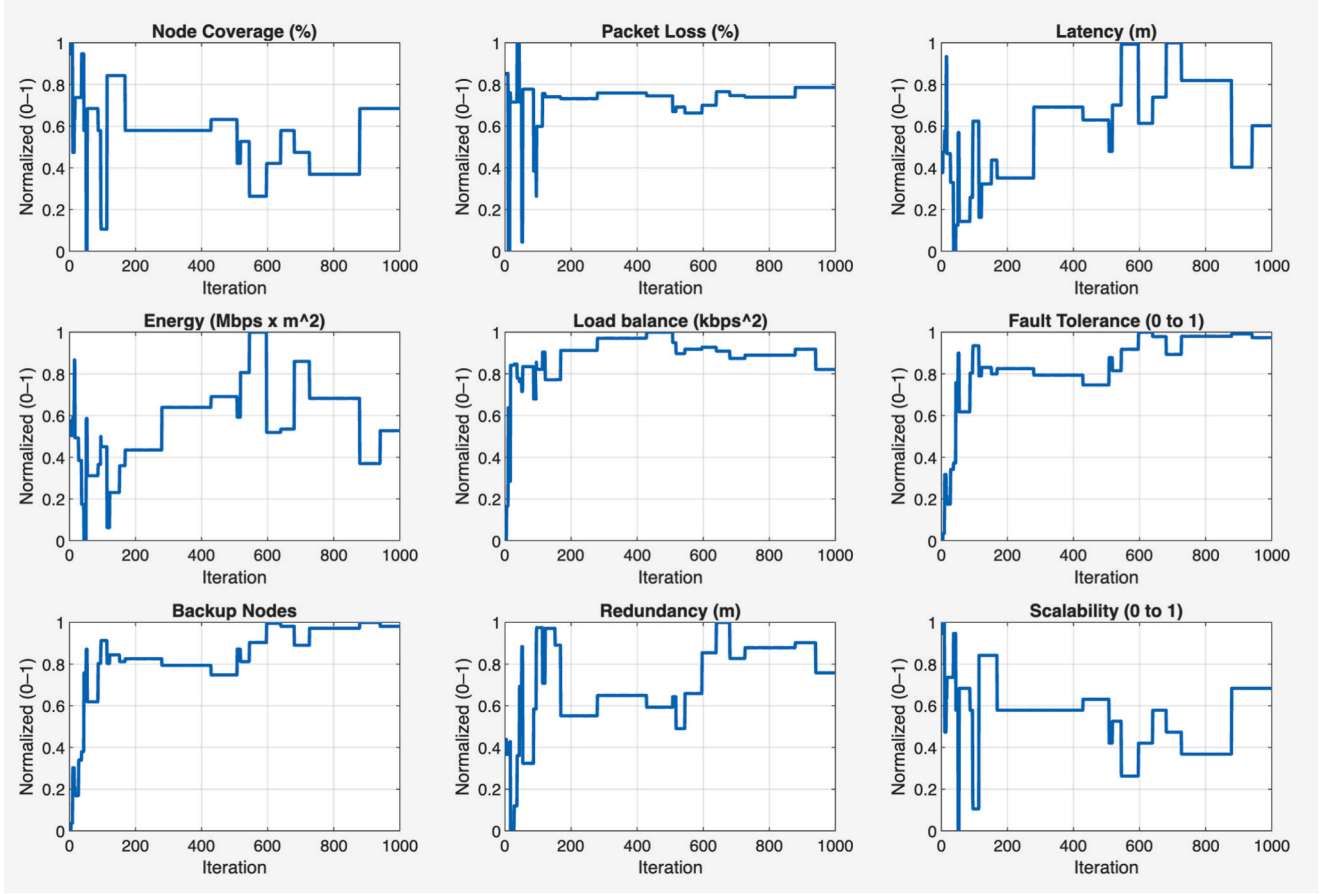


Fig. 22. Normalized performance metrics across iterations for a 3000 m communication range.

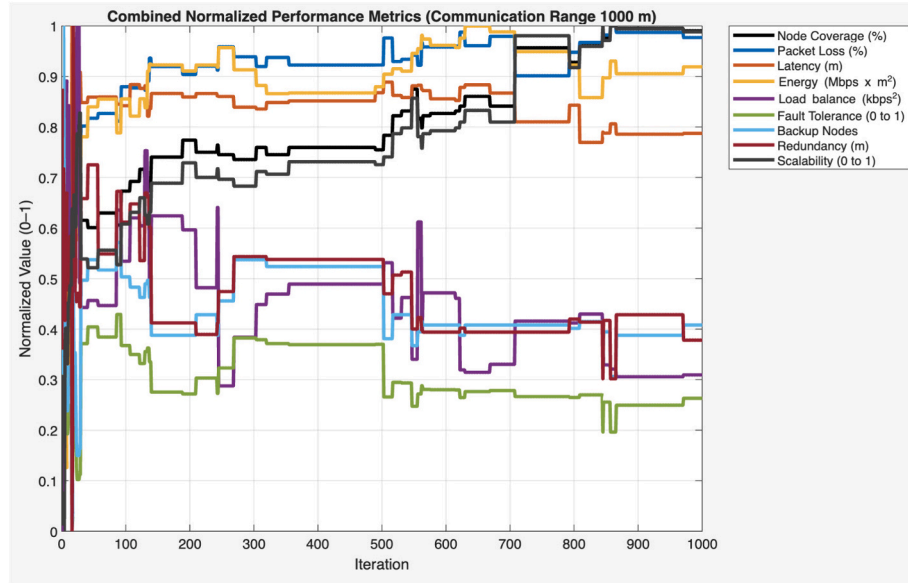


Fig. 23. Comparative visualization of all normalized metrics for 1000 m communication range.

for TataNId, Deutsche, and Forthnet topologies. Similarly, the method proposed by Hans et al. [26] achieved 20% lower latency than PSO using the ESFO-POCO framework, while Ali and Alenazi [38] reported that their proposed method reduced delay by 33.33% to 42.00% compared with K-Means across multiple topologies. In comparison, the proposed method reduced latency for a 1000 m communication range by 19.96%

compared with Random placement. For a 3000 m communication range, latency improved by 38.92% compared with random placement, 10.16% compared with APDQN, 4.15% compared with GA, 4.15% compared with K-Means, and 2.77% compared with PSO. In terms of energy efficiency, Maity et al. [30] proposed the EnPlace method, which achieved 22.74% higher energy savings than GreCo at 200 devices,

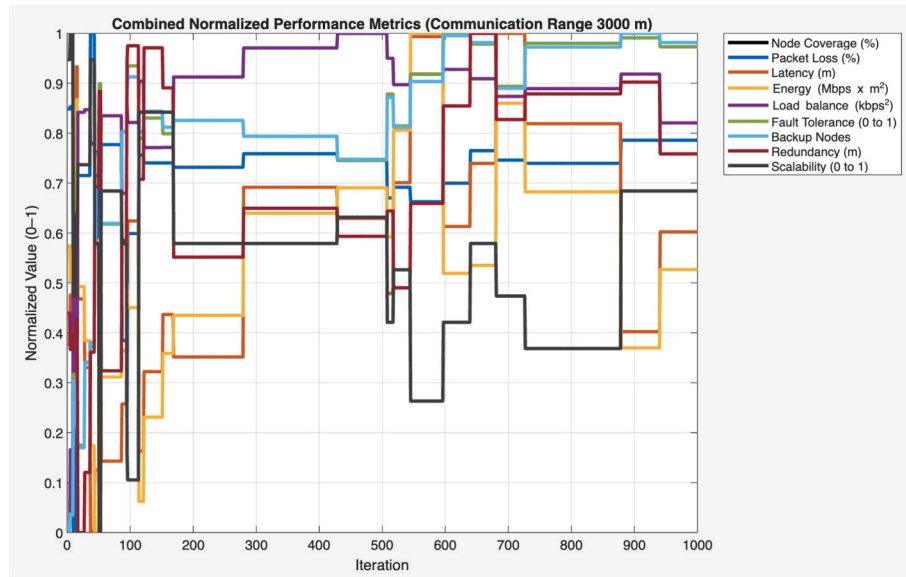


Fig. 24. Comparative visualization of all normalized metrics for 3000 m communication range.

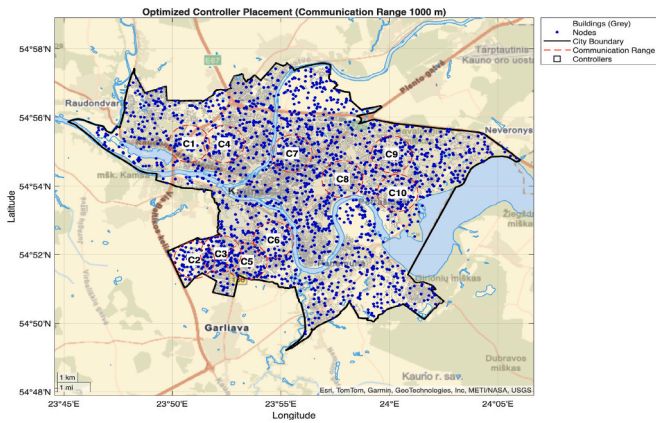


Fig. 25. Optimized controller placement for 1000 m communication range.

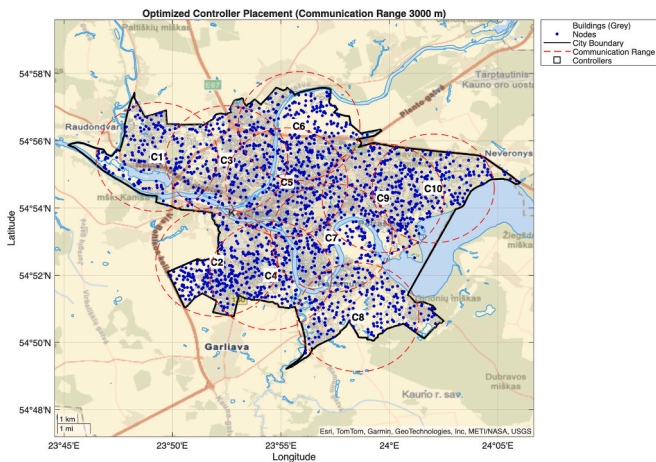


Fig. 26. Optimized controller placement for 3000 m communication range.

increasing to about 38% at 400 devices. In comparison, the proposed method reduced energy consumption at 1000 m communication range by 14.45% compared with Random placement, 15.09% compared with

APDQN, 12.06% compared with K-Means, 3.60% compared with PSO, and 3.46% compared with GA. For a 3000 m communication range, energy efficiency improved by 50.09% over Random placement, 16.53% over APDQN, 9.10% over GA, 4.20% over K-Means, and 4.15% over PSO, showing better energy optimization across both communication ranges compared with other methods. For load balancing, Babbar et al. [40] proposed the MDCLB method, which improved controller load balancing and resulted in approximately 5% improvement in CPU utilization. In comparison, the proposed method improved load balancing at 1000 m communication range by 29.63% compared with APDQN. At 3000 m, load balancing improved further by 92.53% compared with Random placement, 93.89% compared with APDQN, 70.95% compared with K-Means, 66.54% compared with GA, and 36.30% over PSO, substantially outperforming these methods. In terms of reliability and fault tolerance, Abderrahmane and Drid [20] reported reliability values close to 1.0, while Zangeneh et al. [39] improved network reliability using backup controller placement. In comparison, at 1000 m communication range, the proposed method achieved fault tolerance improvements of 23.26% compared with Random placement and 65.63% compared with GA, while K-Means fault tolerance remained zero. At 3000 m, the proposed framework maintained 19.24% higher fault tolerance than Random placement, 27.96% higher than K-Means, and 64.82% higher than APDQN, confirming competitive reliability at both communication ranges. Regarding scalability, Abderrahmane and Drid [20] proposed the Louvain Algorithm and Betweenness Centrality method, which demonstrated scalability up to 1000 nodes, while Tran et al. [28] proposed a submodular optimization method that showed scalability for networks up to 400 nodes. In comparison, the proposed framework was evaluated using 2000 nodes. At 1000 m communication range, the proposed method achieved 51.02% higher scalability compared with Random placement, 20.33% over APDQN, 14.73% over K-Means, 11.28% over PSO, and 4.96% over GA. At 3000 m, the proposed method achieved 17.94% higher scalability compared with Random placement, 1.23% over PSO, and 2.49% over GA, demonstrating competitive network scalability across both deployment ranges.

6. Conclusion

This paper presented a realistic, multi-objective approach for optimal controller placement in smart city IoT networks using a hybrid PSO-GA method. The proposed method used real geographic data from Kaunas city, including city boundaries, building footprints, and water

Table 19

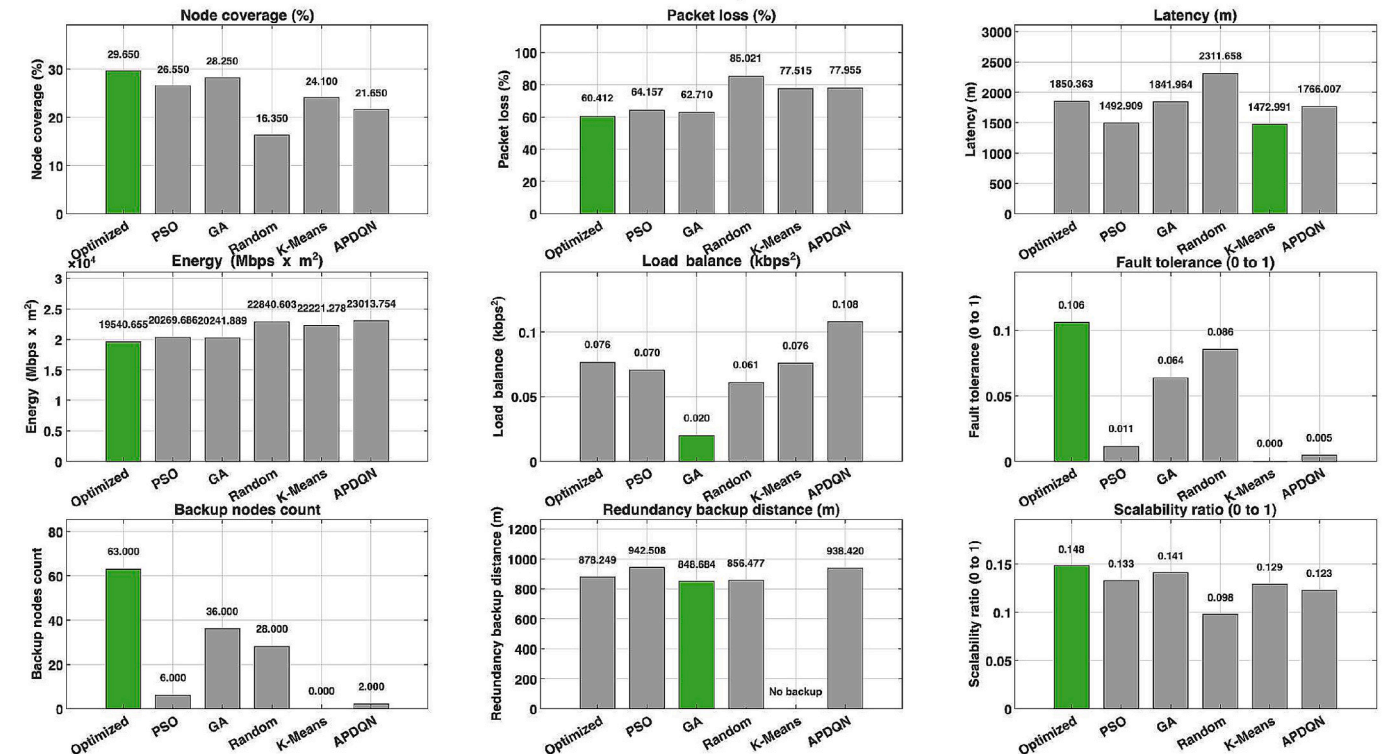
Comparative analysis of controller placement strategies based on performance metrics for a 1000 m communication range.

Metric	Optimized	PSO	GA	Random	K-Means	APDQN
Node coverage (%)	29.65	26.55	28.25	16.35	24.10	21.65
Packet loss (%)	60.412	64.157	62.710	85.021	77.515	77.955
Latency (m)	1850.363	1492.909	1841.964	2311.658	1472.991	1766.007
Energy (Mbps $\times$ m ²)	19,540.655	20,269.686	20,241.889	22,840.603	22,221.278	23,013.754
Load balance (kbps ²)	0.076	0.070	0.020	0.061	0.076	0.108
Fault tolerance (0 to 1)	0.106	0.011	0.064	0.086	0.000	0.005
Backup nodes count	63	6	36	28	0	2
Redundancy backup distance (m)	878.249	942.508	848.684	856.477	No backup	938.420
Scalability ratio (0 to 1)	0.148	0.133	0.141	0.098	0.129	0.123

Table 20

Comparative analysis of controller placement strategies based on performance metrics for a 3000 m communication range.

Metric	Optimized	PSO	GA	Random	K-Means	APDQN
Node coverage (%)	98.55	97.45	96.25	75.50	99.55	97.35
Packet loss (%)	0.710	2.265	2.879	23.533	0.098	1.437
Latency (m)	1411.863	1452.027	1473.062	2311.658	1472.991	1571.530
Energy (Mbps $\times$ m ²)	61,032.320	63,676.395	67,143.795	122,290.951	63,705.622	73,118.898
Load balance (kbps ²)	0.086	0.135	0.257	1.152	0.296	1.408
Fault tolerance (0 to 1)	0.595	0.604	0.651	0.499	0.465	0.361
Backup nodes count	1173	1177	1253	753	925	702
Redundancy backup distance (m)	2440.809	2443.887	2326.947	2116.165	2532.450	2561.878
Scalability ratio (0 to 1)	0.493	0.487	0.481	0.418	0.498	0.511

Best Method per Metric**Fig. 27.** Comparative analysis of best performing methods across performance metrics for 1000 m communication range.

regions, enabling practical urban deployment modeling under realistic environmental constraints. A multi-objective fitness function was designed to jointly optimize node coverage, packet loss, latency, energy consumption, controller load balancing, fault tolerance, redundancy, scalability, and spatial distribution. In addition, building coverage analysis was introduced as an evaluation metric to assess practical service coverage across urban residential, commercial, industrial, and educational zones, thereby improving the realism of the controller

placement algorithm. For a communication range of 1000 m, the proposed method achieved 25.01% building coverage, outperforming PSO, Random Placement, K-Means, and APDQN, while GA achieved a marginally higher building coverage of 25.21%. Despite GA's slightly better building coverage, the proposed method demonstrated balanced overall network performance by achieving the highest node coverage of 29.65%, the lowest packet loss of 60.41%, the lowest energy consumption of 19,540.66 Mbps $\times$ m², the highest fault tolerance of 0.106,

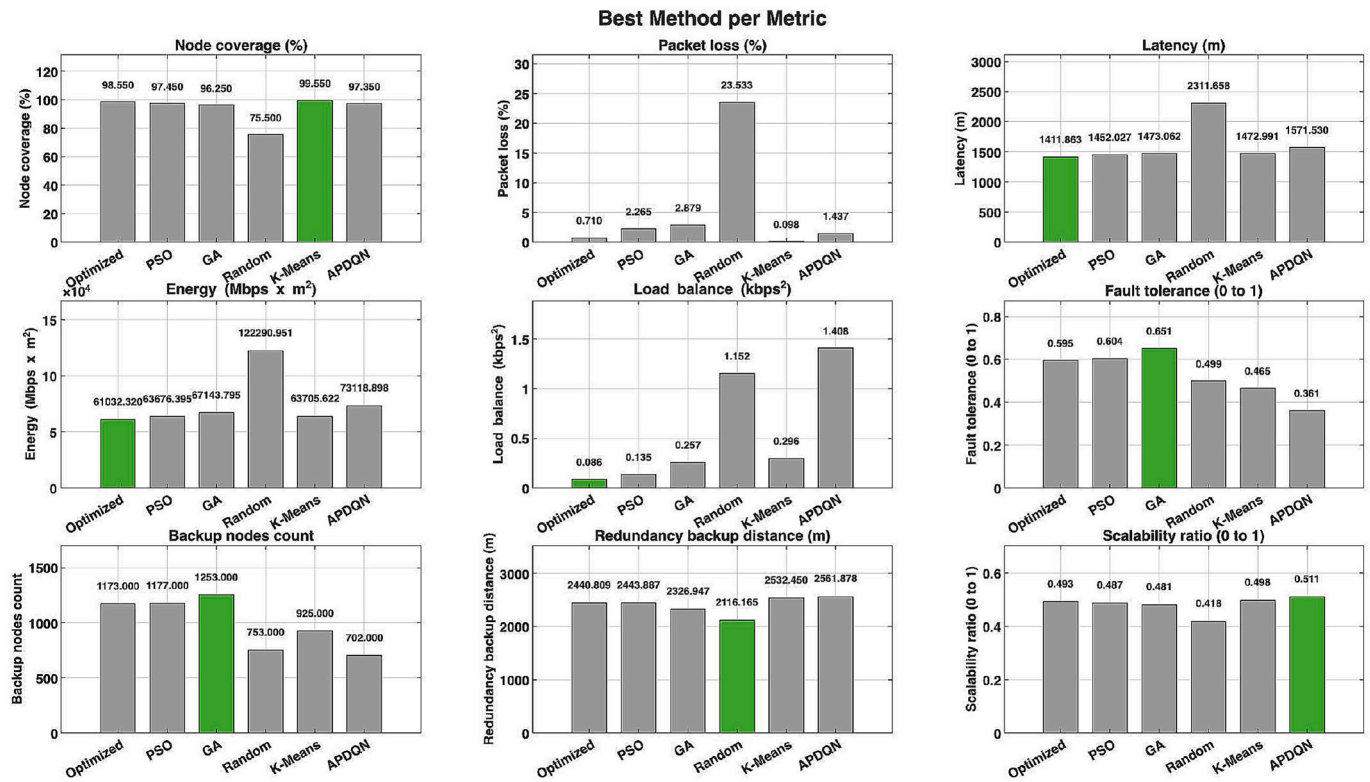


Fig. 28. Comparative analysis of best performing methods across performance metrics for 3000 m communication range.

Table 21

Per-controller node coverage analysis across different placement methods for 1000 m communication range.

Controller	Optimized	PSO	GA	Random	K-Means	APDQN
C1	41	45	40	26	32	20
C2	61	104	67	35	102	31
C3	78	54	49	50	49	25
C4	53	39	75	23	51	32
C5	59	51	67	15	38	104
C6	61	51	54	40	32	60
C7	64	35	57	29	48	56
C8	46	51	36	24	46	30
C9	67	57	64	38	36	26
C10	63	44	56	47	48	49
Total nodes	2000	2000	2000	2000	2000	2000
Covered nodes	593	531	565	327	482	433
Uncovered nodes	1407	1469	1435	1673	1518	1567
Node coverage (%)	29.65	26.55	28.25	16.35	24.1	21.65

Table 22

Per-controller node coverage analysis across different placement methods for 3000 m communication range.

Controller	Optimized	PSO	GA	Random	K-Means	APDQN
C1	168	210	182	146	167	102
C2	233	229	193	138	269	67
C3	189	218	176	273	224	287
C4	189	197	195	186	229	261
C5	217	189	222	16	181	236
C6	213	185	213	178	145	153
C7	188	193	190	81	232	313
C8	169	161	123	120	186	97
C9	229	216	213	94	131	140
C10	176	151	218	278	227	291
Total nodes	2000	2000	2000	2000	2000	2000
Covered nodes	1971	1949	1925	1510	1991	1947
Uncovered nodes	29	51	75	490	9	53
Node coverage (%)	98.55	97.45	96.25	75.5	99.55	97.35

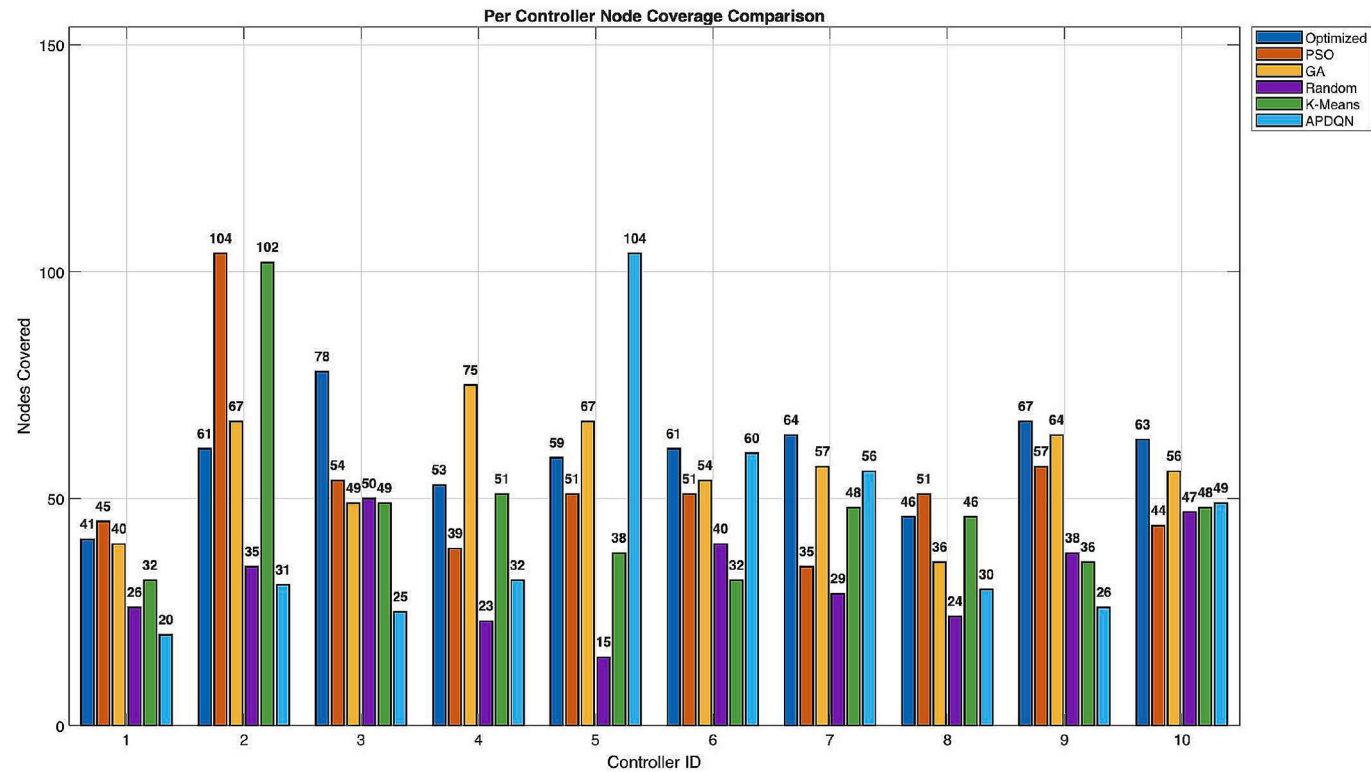


Fig. 29. Per-controller node coverage analysis across different placement methods for a 1000 m communication range.

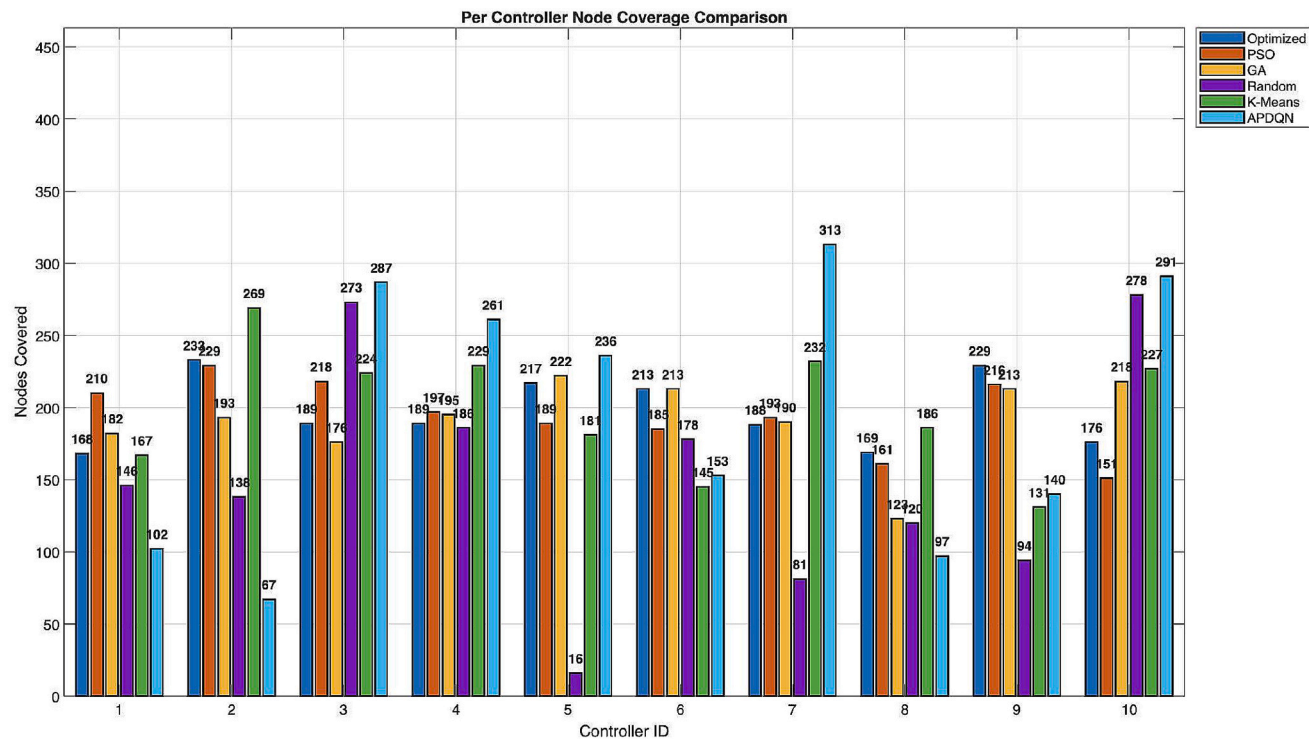


Fig. 30. Per-controller node coverage analysis across different placement methods for a 3000 m communication Range.

high backup node availability of 63, and the highest scalability ratio of 0.148. These results indicate a more balanced, efficient, and reliable controller placement strategy. For a communication range of 3000 m, the proposed method achieved 97.79% building coverage, outperforming PSO, GA, Random Placement, and APDQN, while K-Means

achieved the highest building coverage of 99.31%. However, the proposed method maintained competitive and balanced overall network performance by achieving reduced latency of 1411.86 m, the lowest energy consumption of 61,032.32 Mbps × m², and improved load balancing, with a load variance of 0.086 kbps² compared with PSO, GA,

Table 23

Comparative analysis of building coverage across different controller placement algorithms for a 1000 m communication range.

Controller	Optimized	PSO	GA	Random	K-Means	APDQN
C1	769	806	747	881	1421	60
C2	665	2298	1227	555	2255	1410
C3	1786	2105	1419	453	1419	1243
C4	1366	995	1983	1827	1157	1540
C5	2252	432	1263	1050	36	2298
C6	969	1140	1755	555	994	1551
C7	2505	650	2240	792	1280	546
C8	1458	1337	1358	256	702	546
C9	480	464	1307	1095	957	1386
C10	1470	656	531	367	368	1023
Total buildings	54,847	54,847	54,847	54,847	54,847	54,847
Covered buildings	13,720	10,883	13,830	7831	10,589	11,603
Uncovered buildings	41,127	43,964	41,017	47,016	44,258	43,244
Building coverage (%)	25.01	19.84	25.21	14.27	19.30	21.15

Table 24

Comparative analysis of building coverage across different controller placement algorithms for a 3000 m communication range.

Controller	Optimized	PSO	GA	Random	K-Means	APDQN
C1	5364	6535	6126	4853	5333	2155
C2	5113	5290	4059	3570	6624	3418
C3	7315	10,591	7362	7653	8416	7757
C4	6298	6071	5195	6036	8110	9751
C5	9278	3343	3735	1217	2966	6741
C6	3231	5852	8990	4532	5419	1867
C7	4793	4928	5915	2289	6721	10,196
C8	5055	4922	4173	1717	3268	2370
C9	4259	3454	4283	3158	4089	3939
C10	2934	2600	3395	4688	3526	4773
Total buildings	54,847	54,847	54,847	54,847	54,847	54,847
Covered buildings	53,640	53,586	53,233	39,713	54,472	52,967
Uncovered buildings	1207	1261	1614	15,134	375	1880
Building coverage (%)	97.79	97.70	97.05	72.40	99.31	96.57

Table 25

Building type -wise coverage comparison across different algorithms for 1000 m communication range.

Building type	Total	Optimized	PSO	GA	Random	K-Means	APDQN
Commercial	382	89	67	86	36	79	50
Educational Institute	228	59	60	61	30	40	32
Hospital	28	10	9	9	6	3	4
Industrial	279	70	83	71	23	83	106
Other	2331	570	591	672	295	394	184
Religious	30	4	2	4	1	3	2
Residential	11,004	3772	2299	3600	1297	1864	1228
Unspecified	40,665	9146	7772	9327	6153	8123	9997
Total buildings	54,847						
Covered buildings		13,720	10,883	13,830	7831	10,589	11,603
Uncovered buildings		41,127	43,964	41,017	47,016	44,258	43,244

Table 26

Building type -wise coverage comparison across different algorithms for 3000 m communication ranges.

Building Type	Total	Optimized	PSO	GA	Random	K-Means	APDQN
Commercial	382	379	382	377	268	382	364
Educational Institute	228	228	228	226	163	228	215
Hospital	28	28	28	28	22	28	26
Industrial	279	274	279	271	186	275	263
Other	2331	2297	2301	2290	1887	2311	2247
Religious	30	29	29	29	17	30	27
Residential	11,004	10,964	10,967	10,944	8249	10,971	10,777
Unspecified	40,665	39,441	39,372	39,068	28,921	40,247	39,048
Total buildings	54,847						
Covered buildings		53,640	53,586	53,233	39,713	54,472	52,967
Uncovered buildings		1207	1261	1614	15,134	375	1880

Random Placement, K-Means, and APDQN. Overall, the results confirm that the proposed hybrid PSO-GA algorithm provides a practical,

scalable, and balanced solution for controller placement in large-scale smart city IoT networks. By integrating realistic geographic

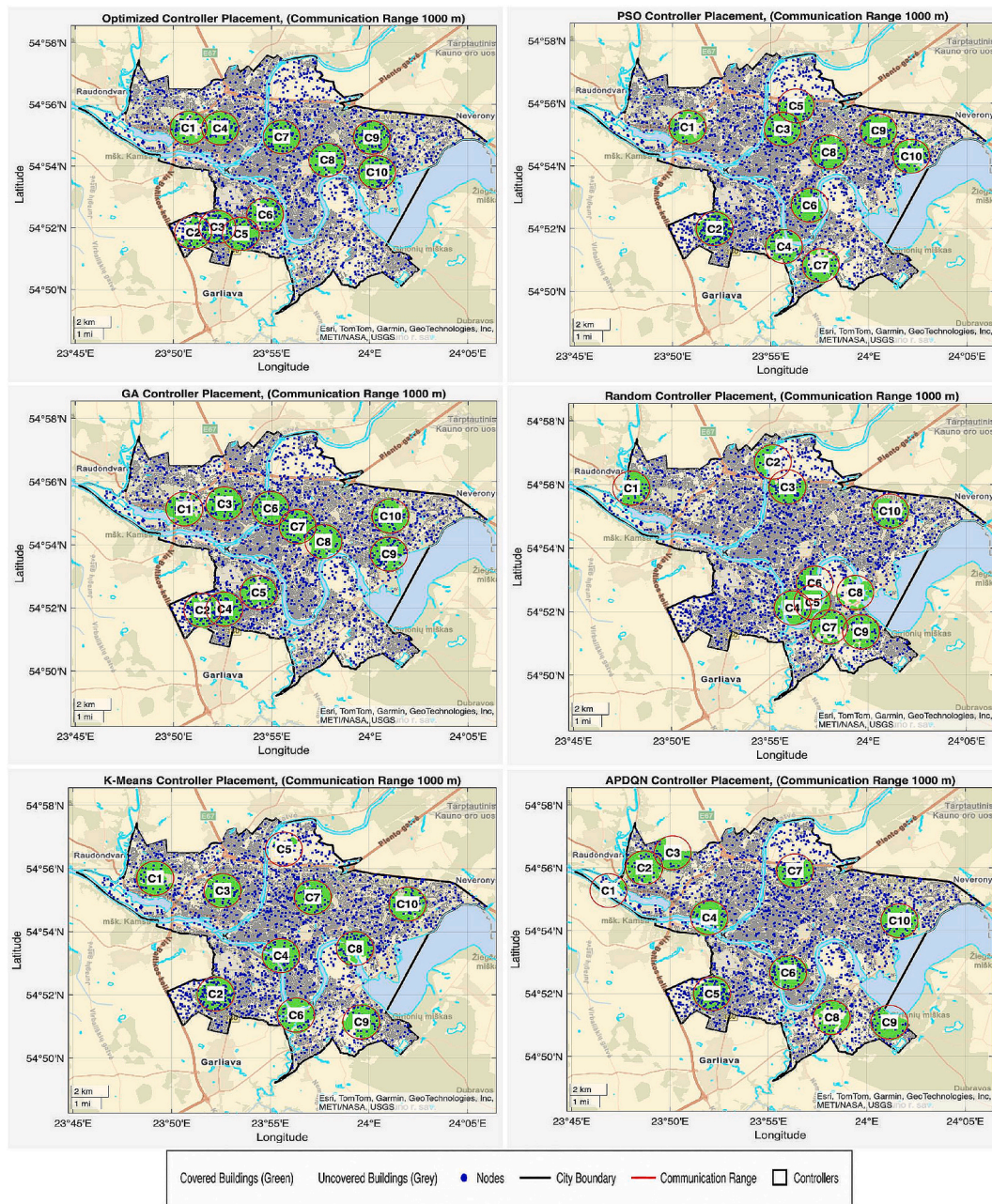


Fig. 31. Comparative visualization of controller placement strategies for IoT networks at a communication range of 1000 m.

constraints, multi-metric optimization, and building-level coverage analysis, the proposed method improves network efficiency, reliability, and deployment practicality compared with conventional approaches. Future work may focus on dynamic traffic adaptation and real-time controller reallocation strategies to further enhance controller placement in large-scale smart city IoT networks. In addition, the proposed framework will be evaluated against additional state of the art controller placement methods, including DEWO, GWOAP, and ESFO-POCO, to further validate the effectiveness of the proposed framework.

CRedit authorship contribution statement

Sheeraz Ali Memon: Writing – original draft, Visualization, Resources, Investigation, Formal analysis, Data curation. **Darius Andriukaitis:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Conceptualization. **Vitas Grimala:** Software, Investigation, Formal analysis. **Tomyslav**

Sledevič: Visualization, Software, Formal analysis. **Vytautas Markevičius:** Writing – review & editing, Conceptualization. **Mindaugas Žilys:** Writing – review & editing, Funding acquisition. **Algimantas Valinevičius:** Visualization, Funding acquisition, Conceptualization. **Dangirutis Navikas:** Writing – review & editing, Formal analysis, Data curation. **Jaromir Konecny:** Writing – review & editing, Resources, Data curation. **Michal Prauzek:** Writing – review & editing, Visualization, Formal analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Darius Andriukaitis reports financial support was provided by Research Council of Lithuania (LMTLT) and the Ministry of Education, Science and Sport of the Republic of Lithuania, agreement no. S-A-UEI-23-1 (22 December 2023). N/A If there are other authors, they declare that they

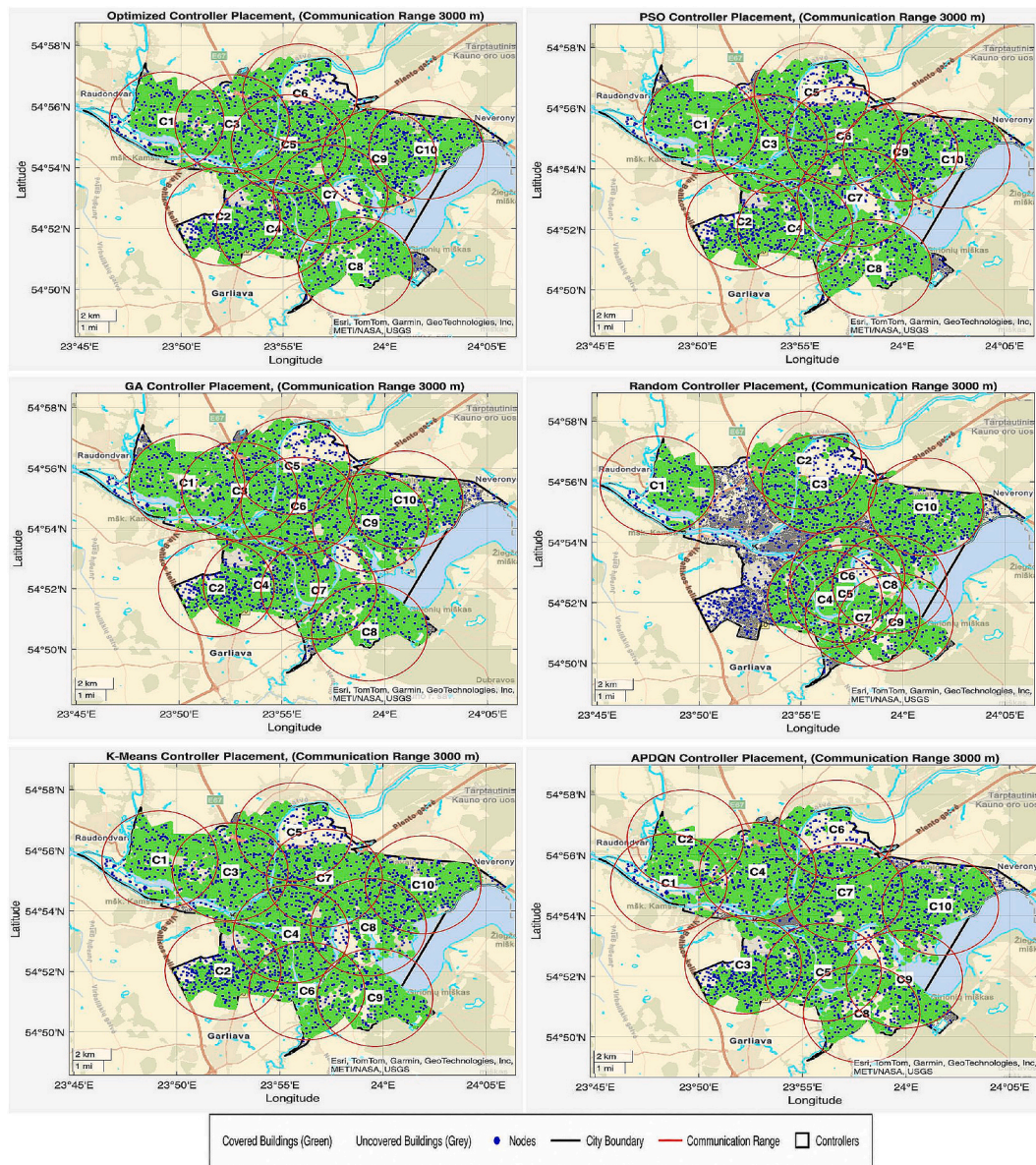


Fig. 32. Comparative visualization of controller placement strategies for IoT networks at a communication range of 3000 m.

have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This project has received financial support from the Research Council of Lithuania (LMTLT) and the Ministry of Education, Science and Sport of the Republic of Lithuania, agreement No: S-A-UEI-23-1.

Data availability

Data will be made available on request.

References

- [1] Ali J, Roh B. An effective approach for controller placement in software-defined internet-of-things (SD-IoT). *Sensors (Basel)* 2022;22(8). <https://doi.org/10.3390/s22082992>.
- [2] Abderrahmane A, Drid H, Behaz A. A survey of controller placement problem in SDN-IoT network. *Int J Networked Distrib Comput* 2024;12(2):170–84. <https://doi.org/10.1007/s44227-024-00035-y>.
- [3] Shaji NS, Muthalagu R. Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN. *Digit Commun Netw* 2024;10(6):1716–31. <https://doi.org/10.1016/j.dcan.2023.09.004>.
- [4] Gong W, Pang L, Wang J, Xia M, Zhang Y. A social-aware K means clustering algorithm for D2D multicast communication under SDN architecture. *AEU Int J Electron Commun* 2021;132:153610. <https://doi.org/10.1016/j.aeue.2021.153610>.
- [5] Yalda K, Hamad DJ, Tapus N. Comparative analysis of centralized and distributed SDN environments for IoT networks. *J Control Eng Appl* 2024;26(3):84–91. <https://doi.org/10.61416/ceai.v26i3.9164>.
- [6] Memon SA, et al. Centralized and distributed controller placement in terms of scalability and fault tolerance: A review. 2025 29th International Conference on methods and models in automation and robotics (MMAR). Miedzydroje, Poland: IEEE; 2025. p. 449–54. <https://doi.org/10.1109/MMAR65820.2025.11150828>.
- [7] Skirelis J, Navakauskas D. Performance analysis of edge computing in IoT. *Elektron Ir Elektrotechnika* 2020;26(1):72–7. <https://doi.org/10.5755/joi.eie.26.1.23235>.
- [8] Abdul Salam R, Iqbal Ratyal N, Ahmed U, Aziz I, Sajid M, Mahmood A. An overview of recent wireless technologies for IoT-enabled smart grids. *J Electr Comput Eng* 2024;2024(1):2568751. <https://doi.org/10.1155/jece/2568751>.
- [9] Prauzek M, Paterova T, Stankus M, Mikus M, Konecny J. Analysis of LoRaWAN transactions for TEG-powered environment-monitoring devices. *Elektron Ir Elektrotechnika* 2022;28(3):30–6. <https://doi.org/10.5755/joi.eie.31265>.
- [10] Singh Bali M, Gupta K, Kour Bali K, Singh PK. Towards energy efficient NB-IoT: a survey on evaluating its suitability for smart applications. *Mater Today Proc* 2022; 49:3227–34. <https://doi.org/10.1016/j.matpr.2020.11.1027>.

- [11] Memon SA, et al. A survey on controller placement algorithms for IoT networks in Smart City environments. In: 2025 IEEE 12th workshop on advances in information, electronic and electrical engineering (AIEEE). Vilnius, Lithuania: IEEE; 2025. p. 1–6. <https://doi.org/10.1109/AIEEE66149.2025.11050873>.
- [12] He X, Wang X, Xiang W, Zhang G. Credibility evaluation of network simulations for urban communication: a systematic framework from node characteristics to global link association. *AEU Int J Electron Commun* 2026;215:156411. <https://doi.org/10.1016/j.aeue.2026.156411>.
- [13] GmbH Geofabrik. Download OpenStreetMap data for Lithuania, Geofabrik Download Server [Online]. Available: <https://download.geofabrik.de/europe/lithuania.html>; 2026.
- [14] Zhao X. Research on multi-objective scheduling optimization of power system based on hybrid model of genetic algorithm and particle swarm optimization. *Discov Comput* 2026;29(1):209. <https://doi.org/10.1007/s10791-026-09929-7>.
- [15] Memon SA, et al. Hybrid GA-PSO optimization for controller placement in large-scale Smart City IoT networks. *Sensors* 2025;25(23):7119. <https://doi.org/10.3390/s25237119>.
- [16] Heller B, Sherwood R, McKeown N. The controller placement problem. 2012.
- [17] Keshari SK, Kansal V, Kumar S. An intelligent way for optimal controller placements in software-defined-IoT networks for smart cities. *Comput Ind Eng* 2021;162. <https://doi.org/10.1016/j.cie.2021.107667>.
- [18] Bagaa M, Taleb T, Laghrissi A, Ksentini A, Flinck H. Coalitional game for the creation of efficient virtual Core network slices in 5G Mobile systems. *IEEE J Sel Areas Commun* 2018;36(3):469–84. <https://doi.org/10.1109/JSAC.2018.2815398>.
- [19] Hu T, et al. An efficient approach to robust controller placement for link failures in software-defined networks. *Futur Gener Comput Syst* 2021;124:187–205. <https://doi.org/10.1016/j.future.2021.05.022>.
- [20] Abderrahmane A, Drid H. Enhancing control placement in SDN-IoT networks using the Louvain algorithm and Betweenness-centrality. *IEEE Access* 2024;12: 159775–83. <https://doi.org/10.1109/ACCESS.2024.3479916>.
- [21] Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. *J Stat Mech Theory Exp* 2008;2008(10):P10008. <https://doi.org/10.1088/1742-5468/2008/10/P10008>.
- [22] Xu H, Chai X, Liu H. A multi-controller placement strategy for hierarchical management of software-defined networking. *Symmetry* 2023;15(8):1520. <https://doi.org/10.3390/sym15081520>.
- [23] Abdelaziz A, et al. Distributed controller clustering in software defined networks. *PLoS One* 2017;12(4):e0174715. <https://doi.org/10.1371/journal.pone.0174715>.
- [24] Yao G, Bi J, Li Y, Guo L. On the capacitated controller placement problem in software defined networks. *IEEE Commun Lett* 2014;18(8):1339–42. <https://doi.org/10.1109/LCOMM.2014.2332341>.
- [25] Zhao J, Qu H, Zhao J, Luan Z, Guo Y. Towards controller placement problem for software-defined network using affinity propagation. *Electron Lett* 2017;53(14): 928–9. <https://doi.org/10.1049/el.2017.0093>.
- [26] Hans S, Ghosh S, Kataria A, Karar V, Sharma S. Controller placement in software defined internet of things using optimization algorithm. *Comput Mater Contin* 2022;70(3):5073–89. <https://doi.org/10.32604/cmc.2022.019971>.
- [27] Keshari SK, Kansal V, Kumar AS. A cluster based intelligent method to manage load of controllers in Sdn-Iot networks for smart cities. *Scalable Comput* 2021;22(2): 247–57. <https://doi.org/10.12694/scpe.v22i2.1912>.
- [28] Tran A, Piran M, Pham C. SDN controller placement in IoT networks: an optimized submodularity-based approach. *Sensors (Basel)* 2019;19(24). <https://doi.org/10.3390/s19245474>.
- [29] Boudi R, Lalou M, Bouchemal N, IEEE. On the Controller Placement Problem: a New SD-IoT Topology for the Algerian Network. In: Presented at the 2024 International Conference on Information and Communication Technologies for Disaster Management, ICT-DM 2024; 2024. <https://doi.org/10.1109/ICT-DM62768.2024.10798934>.
- [30] Maity I, Dhiman R, Misra S. EnPlace: energy-aware network partitioning for controller placement in SDN. *IEEE Trans Green Commun Netw* 2023;7(1):183–93. <https://doi.org/10.1109/TGCN.2022.3175901>.
- [31] Ahmed M, Peng M, Abana M, Yan S, Wang C. Interference coordination in heterogeneous small-cell networks: a coalition formation game approach. *IEEE Syst J* 2018;12(1):604–15. <https://doi.org/10.1109/JSYST.2015.2487604>.
- [32] Tsiropoulou EE, Vamvakas P, Papavassiliou S. Joint customized Price and power control for energy-efficient multi-service wireless networks via S-modular theory. *IEEE Trans Green Commun Netw* 2017;1(1):17–28. <https://doi.org/10.1109/TGCN.2017.2678207>.
- [33] Giroire F, Moulrierac J, Phan TK. Optimizing rule placement in software-defined networks for energy-aware routing. In: 2014 IEEE global communications Conference. Austin, TX, USA: IEEE; 2014. p. 2523–9. <https://doi.org/10.1109/GLOCOM.2014.7037187>.
- [34] Maaloul R, Taktak R, Chaari L, Cousin B. Energy-aware routing in carrier-grade Ethernet using SDN approach. *IEEE Trans Green Commun Netw* 2018;2(3):844–58. <https://doi.org/10.1109/TGCN.2018.2832658>.
- [35] Assefa BG, Ozkasap O. RESDN: a novel metric and method for energy efficient routing in software defined networks. *IEEE Trans Netw Serv Manag* 2020;17(2): 736–49. <https://doi.org/10.1109/TNSM.2020.2973621>.
- [36] Boudi R, Lalou M, Bouchemal N, Gherbi C. Optimizing latency in SD-IoT through heterogeneous traffic flow-based controller placement. In: Presented at the 2024 International Symposium on Networks, Computers and Communications, ISNCC 2024; 2024. <https://doi.org/10.1109/ISNCC62547.2024.10758959>.
- [37] Knight S, Nguyen HX, Falkner N, Bowden R, Roughan M. The internet topology zoo. *IEEE J Sel Areas Commun* 2011;29(9):1765–75.
- [38] Ali J, Alenazi M. Effective controller placement in software-defined internet-of-things leveraging deep Q-learning (DQL). *CMC-Comput Mater Contin* 2024;81(3): 4015–32. <https://doi.org/10.32604/cmc.2024.058480>.
- [39] Zangeneh I, Bidgoli AM, Dolati A. Novel solution of the controller and backup controller placement problem for improving reliability in IoT-based data monitoring systems. *J Supercomput* 2025;81(2). <https://doi.org/10.1007/s11227-024-06846-8>.
- [40] Babbar H, Rani S, Gupta D, Aljahdali HM, Singh A, Al-Turjman F. Load balancing algorithm on the immense scale of internet of things in SDN for smart cities. *Sustainability* 2021;13(17):9587. <https://doi.org/10.3390/su13179587>.
- [41] Rafique Y, Leivadreas A, Ibnkahla M. An IoT-aware VNF placement proof of concept in a hybrid edge-cloud Smart City environment. In: 2022 IEEE wireless communications and networking Conference (WCNC). Austin, TX, USA: IEEE; 2022. p. 1395–400. <https://doi.org/10.1109/WCNC51071.2022.9772004>.
- [42] Faragardi H, Vahabi M, Fotouhi H, Nolte T, Fahringer T. An efficient placement of sinks and SDN controller nodes for optimizing the design cost of industrial IoT systems. *Softw-Pract Exp* 2018;48(10):1893–919. <https://doi.org/10.1002/spe.2593>.
- [43] Milovanovic MJ, Raicevic SS, Klimenta DO, Raicevic NB, Perovic BD. Determination of optimal locations and parameters of passive harmonic filters in unbalanced systems using the multiobjective genetic algorithm. *Elektron Ir Elektrotehnika* 2024;30(2):28–37. <https://doi.org/10.5755/j02.eie.36124>.
- [44] Hassanat A, Almohammadi K, Alkafaween E, Abunawas E, Hammouri A, Prasath VBS. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information* 2019;10(12):390. <https://doi.org/10.3390/info10120390>.
- [45] Casas-Ordaz A, et al. Particle swarm optimization: a survey of innovations over the last 10 years. *Comput Sci Rev* 2026;60:100910. <https://doi.org/10.1016/j.cosrev.2026.100910>.
- [46] Tossa F, Faga Y, Abdou W, Ezin EC, Gouton P. Wireless sensor network deployment: architecture, objectives, and methodologies. *Sensors (Basel)* 2025;25(11):3442. <https://doi.org/10.3390/s25113442>.
- [47] Hattian. NB-IoT 3GPP release 14. What are the new features?. Accessed: Feb. 24, <https://hattian.com/resources/nb-iot-3gpp-release-14-what-are-the-new-features/>; 2026.
- [48] Abualigah L. Particle swarm optimization: advances, applications, and experimental insights. *Comput Mater Contin* 2025;82(2):1539–92. <https://doi.org/10.32604/cmc.2025.060765>.
- [49] International Institute for Aerospace Survey and Earth Sciences. Principles of remote sensing and geographic information systems. In: ITC educational textbook series. 2nd ed.1/2. Enschede: ITC; 2001.