STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–26.
Published online in International Academic Press (www.IAPress.org)

FK-FGO: A Fractional Krawtchouk–Fungal Growth Hybrid for Optimal Fog Node Deployment in IoT Networks

Issa Alsmadi [1], Islam S. Fathi [2,3,*], Ahmad Dalalah [2], Ghazi Shakah [4], Yasser Mohammad Al-Sharo [5]

[1] *Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Ajloun National University, Ajloun, Jordan*
[2] *Department of Computer Science, Faculty of Information Technology, Ajloun National University P.O.43, Ajloun-26810, Jordan*
[3] *Department of Information Systems, Al Alson Higher Institute, Cairo 11762, Egypt*
[4] *Faculty of Information Technology, Ajloun National University P.O.43, Ajloun-26810, Jordan*
[5] *Department of Cyber Security, Ajloun National University, P.O.43, Ajloun 26810, Jordan*

Abstract The rapid growth of Internet of Things (IoT) devices has driven the development of fog computing paradigms aimed at overcoming latency and bandwidth bottlenecks in modern networks. Despite these advances, determining the optimal placement of fog nodes remains a challenging NP-hard problem that requires the joint optimization of network coverage and inter-node connectivity. Current metaheuristic approaches are constrained by their dependence on conventional Euclidean distance metrics, inflexible integer-order objective functions, and a poor balance between exploration and exploitation. This paper proposes a new hybrid optimization framework that integrates Fractional Krawtchouk (FK) moments with the Fungal Growth Optimizer (FGO) for multi-objective fog node placement. FK moments offer advanced spatial encoding capabilities through discrete orthogonal polynomials augmented with fractional-order derivatives, enabling efficient representation of both global network topology and localized coverage characteristics. The fractional-order formulation applies non-integer penalty exponents to achieve adaptable trade-offs between connectivity and coverage objectives. Rigorous experimental evaluation confirms that FK-FGO outperforms leading benchmark algorithms, including Harris Hawks Optimization, Marine Predators Algorithm, and Pufferfish Optimization Algorithm. The proposed method achieves 99.21% connectivity and 98.97% coverage, while delivering computational speedups of 57.3%–102.5% over competing approaches. Compared to the original FGO, FK-FGO converges 31.4% faster and reduces per-iteration processing time by 46.7%, all while demonstrating reliable scalability across network configurations ranging from 50 to 2000 fog nodes, making it a robust solution for next-generation fog computing deployments.

Keywords Internet of Things (IoT), Fog Computing, Fractional Krawtchouk Moments, Fungal Growth Optimizer, Multi-Objective Optimization, Node Placement

AMS 2010 subject classifications 90C59, 68M14

DOI: 10.19139/soic-2310-5070-3890

1. Introduction

The widespread adoption of Internet of Things (IoT) technology across diverse application domains, including smart city infrastructure, healthcare monitoring, industrial automation, and intelligent transportation systems, has fundamentally transformed the architecture of contemporary distributed computing environments [1]. Projections indicate that globally connected IoT devices will surpass 75 billion by 2025, producing massive quantities of data that require immediate processing, minimal response delays, and optimal use of available resources [2]. Although conventional cloud-based computing models provide considerable processing capacity and flexibility,

*Correspondence to: Islam S. Fathi (Email: i.mohamed@anu.edu.jo). Department of Computer Science, Faculty of Information Technology, Ajloun National University P.O.43, Ajloun-26810, Jordan.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

they encounter substantial difficulties in fulfilling the demanding Quality of Service (QoS) standards of time-critical IoT workloads, owing to communication latency introduced by transmitting data to remote data centers, limited network bandwidth, and traffic congestion [3]. These inherent shortcomings have driven the adoption of fog computing as a supplementary model that relocates computational functionality to the edges of the network, bringing processing capabilities into closer geographical proximity to the devices generating data [4]. Originally introduced by Cisco in 2012, fog computing defines a layered distributed system that closes the performance gap between computationally limited IoT endpoints and high-capacity cloud platforms by deploying fog nodes at the network perimeter to handle computation, data storage, and communication tasks [5]. By enabling localized data processing at the edge, fog computing satisfies the stringent latency requirements of diverse real-time IoT applications while simultaneously reducing traffic burden on centralized cloud resources [6]. This design paradigm unlocks a range of practical benefits, including ultra-low latency responses (frequently below 100ms), more efficient use of network bandwidth via on-site data processing and aggregation, stronger privacy protections by limiting the transmission of sensitive information, higher system resilience through distributed task handling, and the ability to deliver services tied to users' physical locations [7]. Empirical research has confirmed that fog-assisted service deployment strategies can cut service response times and energy usage by 2.5%–8% and 6%–17%, respectively, when compared to purely cloud-based solutions [8].

Despite these compelling benefits, the effective deployment of fog computing systems presents formidable optimization challenges, particularly in determining the strategic positioning of fog nodes across the network. Identifying where to place fog nodes, how to configure them, and how to allocate their resources represents a complex NP-hard combinatorial problem requiring the concurrent satisfaction of competing objectives such as network coverage, node interconnection, energy usage, infrastructure expenses, delay reduction, and workload distribution [9]. Optimally assigning heterogeneous IoT services across diverse fog infrastructures is likewise classified as an NP-complete problem [10]. Several compounding factors heighten the difficulty of this challenge: fog nodes differ considerably in their processing power, storage capacity, and wireless transmission reach; IoT application workloads are inherently volatile and difficult to predict; the network must remain resilient and capable of tolerating node failures; and placement strategies must accommodate shifts in environmental conditions and user movement [11]. Conventional solutions have typically drawn on heuristic techniques, formal mathematical models such as integer linear programming, and graph-based analytical methods [12]. Yet these traditional approaches frequently break down in large-scale deployments due to limited scalability, an inability to handle multiple competing objectives, and a tendency to become trapped in suboptimal regions of highly complex and non-convex solution landscapes [13]. This has prompted a growing shift toward nature-inspired, population-based metaheuristic optimization methods, which use stochastic processes to systematically balance broad solution space exploration with focused exploitation of high-quality regions [14].

A growing body of literature has explored the application of metaheuristic methods to fog computing optimization problems. Bio-inspired techniques have shown considerable effectiveness for fog node placement tasks, with the Marine Predator Algorithm, for instance, achieving performance gains of up to 50% relative to baseline comparators [15]. Enhanced Genetic Algorithm variants applied to service placement have also outperformed competing approaches such as CSA-FSPP, GA-PSO, and WOA-FSP [16]. Nevertheless, several recurring weaknesses continue to limit the effectiveness of existing metaheuristic fog placement strategies. A primary concern is the dependence on conventional Euclidean distance measures, which are insufficiently expressive to represent the complex spatial arrangements and geometric characteristics of node distributions, often resulting in poor coverage patterns and wasteful resource allocation [17]. Additionally, most existing models use integer-order objective formulations that apply rigid penalty mechanisms, failing to account for the gradual, non-abrupt performance changes that typify real-world fog environments with inherently fractional-order dynamics [18]. Finally, many current algorithms prematurely converge toward local optima, fail to adequately exploit high-potential solution regions, and struggle to maintain an effective balance between global search and local refinement, especially within the high-dimensional landscapes of large-scale fog deployments [19].

To overcome these fundamental shortcomings, this paper presents a new hybrid optimization framework that brings together three powerful mathematical and computational approaches: Fractional Krawtchouk (FK) moments for rich spatial encoding, fractional-order calculus for adaptable objective function design, and the Fungal Growth

Optimizer (FGO) metaheuristic for thorough solution space traversal [20]. The resulting FK-FGO algorithm fundamentally differs from conventional methods by harnessing the distinctive mathematical properties of each component to resolve the weaknesses embedded in prior approaches. Fractional Krawtchouk moments, built upon discrete orthogonal Krawtchouk polynomials augmented with fractional-order derivative operations, present several meaningful benefits for fog node placement problems. As discrete orthogonal constructs, Krawtchouk moments avoid the need for spatial discretization, introduce no approximation errors, and are capable of isolating local feature information from targeted spatial regions by tuning the binomial distribution parameter [21]. In contrast to continuous moment families such as Zernike or Legendre, which depend on numerical integration and are prone to discretization artifacts, Krawtchouk polynomials are natively discrete and therefore well-matched to the combinatorial structure of node placement tasks [22]. The orthogonality of these polynomials minimizes redundant information while permitting efficient recursive computation. Krawtchouk polynomials have proven valuable across information and coding theory, signal and image analysis, and pattern classification domains [23]. A particularly important capability is their ability to isolate local spatial characteristics by modifying the shape parameter p, granting precise control over coverage assessment, which is essential in fog placement scenarios where both the overall network topology and individual service coverage areas must be jointly optimized [24]. Extending the optimization model with fractional-order calculus addresses a key weakness of classical approaches. Fractional derivatives leverage the memory and nonlocal nature inherent in fractional operators to accelerate convergence, enhance robustness in high-dimensional and nonlinear problem settings, and achieve a more favorable balance between exploration and exploitation [25]. Systems formulated with non-integer-order derivatives and integrals exhibit hereditary behavior and long-term memory effects that reflect historical dependencies, which are especially relevant in fog placement contexts where configuration quality is influenced not merely by immediate local conditions but by the full spatial history of node distributions across the network [26]. Incorporating fractional calculus into metaheuristic optimization controllers, including genetic algorithms, particle swarm methods, and grey wolf-based approaches, has been shown to yield enhanced control performance [27]. When multi-objective problems are formulated with fractional-order penalty functions, the resulting penalty landscape degrades smoothly as constraints are violated rather than applying sharp binary feasibility cutoffs, broadening the effective search domain and enabling more gradual, well-behaved convergence [28]. The Fungal Growth Optimizer, a recently developed nature-inspired metaheuristic, forms the algorithmic core of the proposed framework [29]. FGO takes its cue from the biological growth behavior of fungi, mathematically capturing three distinct processes: directional tip growth of hyphae for broad exploration, lateral branch formation for search diversification, and spore germination to preserve population variety. These biological behaviors are translated into computational operators: diversification operators that use exponential growth rate models and stochastic directional sampling to navigate the solution space widely, and intensification operators that guide the search toward promising zones using chemotaxis-like attraction to high-quality candidate solutions. The algorithm dynamically balances these two modes using normalized fitness scores and time-varying exploration rates, sustaining strong search performance from start to finish. When paired with FK moment-based spatial assessments and fractional-order objective formulations, FGO's biologically inspired mechanisms prove especially effective at navigating the rugged, multi-modal fitness landscapes that characterize fog node placement optimization.

The primary contributions of this research are as follows:

- Present Fractional Krawtchouk moments as an advanced mathematical representation for capturing the spatial arrangement of fog node configurations, enabling more precise assessment of both coverage quality and inter-node connectivity patterns.
- Design a fractional-order multi-objective optimization model that uses non-integer penalty exponents to enable adaptable balancing between network connectivity and spatial coverage goals.
- We introduce a collaborative fusion of FK moments with the Fungal Growth Optimizer, yielding a metaheuristic algorithm that effectively navigates between broad search exploration and targeted exploitation.
- We carry out thorough empirical investigations spanning multiple evaluation dimensions, encompassing analysis of convergence trajectories and scalability testing under varied fog node deployment densities.

The remainder of this paper is presented as follows: Section 2 reviews related work on fog computing optimization, metaheuristic algorithms, and moment-based spatial analysis. Section 3 presents the mathematical preliminaries including Fractional Krawtchouk polynomial theory and the Fungal Growth Optimizer algorithm. Section 4 illustrates the system model, problem formulation, and the proposed FK-FGO algorithm. Section 5 details the experimental setup and presents comprehensive simulation results with comparative analysis. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Related works

The rapid growth of Internet of Things devices has made it necessary to develop fog computing paradigms capable of handling the latency, bandwidth, and energy demands that conventional cloud-centered architectures cannot adequately meet. Researchers have devoted considerable effort to investigating optimization strategies for task scheduling, resource management, and service placement within integrated fog-cloud systems. This section surveys leading-edge methodologies, covering nature-inspired metaheuristic algorithms, energy-aware computing frameworks, fault-resilient deployment strategies, and security-oriented solutions, each contributing to the overall efficiency and dependability of fog computing infrastructures. Task scheduling optimization in fog-cloud computing contexts has garnered significant scholarly interest, with numerous metaheuristic-based solutions yielding encouraging outcomes. Abd Elaziz et al. [30] developed AEOSSA, a method that merges the Artificial Ecosystem-based Optimization algorithm with the Salp Swarm Algorithm to address IoT task scheduling. The framework was tested against both artificially generated workloads (comprising 200 to 1000 tasks) and benchmark real-world datasets, including the NASA iPSC collection (18,239 tasks) and HPC2N (202,871 tasks), distributed across 25 virtual machines. AEOSSA reduced makespan by 1.32% to 6.19% relative to the original AEO baseline and showed markedly stronger performance gains of 43.68% to 68.37% when compared against PSO, SSA, FA, and HHO. Composite metaheuristic strategies have proven especially effective at tackling multi-objective optimization scenarios. Apat and Sahoo [31] put forward JaGW, which fuses the Jaya Algorithm with Grey Wolf Optimization to handle IoT workflow placement tasks. When assessed using the Montage scientific workflow (ranging from 100 to 3000 tasks) via the iFogSim2 simulator, JaGW delivered average energy savings of 24.84% over JAYA, 14.67% over ACO, 14.65% over PSO, and 8.78% over GWO, while attaining Hypervolume and Generational Distance values of 0.7156 and 0.0174, respectively. In a related contribution, Apat et al. [32] developed hybrid algorithms pairing Genetic Algorithm with both Simulated Annealing and Particle Swarm Optimization for multi-objective IoT service placement, achieving notable reductions in latency, energy use, and operational costs in fog-cloud setups evaluated through the YAFS simulation platform. Sabireen and Venkataraman [33] presented EMOPSOC, an augmented multi-objective variant of Particle Swarm Optimization with integrated clustering capabilities, combined with the Fog Picker mechanism for multi-objective resource scheduling. The methodology leveraged non-dominated sorting to guide convergence, K-means clustering for partitioning the solution set, and crowding distance to sustain diversity. Experimental evaluation through iFogSim and jMetal with configurations spanning 10 to 100 IoT components and 10 to 70 fog nodes showed that EMOPSOC achieved the top Inverted Generational Distance and Hypervolume scores on 17 out of 22 benchmark functions. Seifhosseini et al. [34] introduced MoDGWA, a discretized grey-wolf optimizer designed for multi-objective scheduling of independent Bag-of-Tasks workloads across fog clusters. The algorithm simultaneously maps between 20 and 200 independent tasks onto 5 to 25 fog nodes while optimizing makespan, financial cost, and a task scheduling fairness metric. MATLAB-based evaluations showed that MoDGWA lowered makespan, overall cost, TSFF, and composite performance scores by 0.55%, 7.28%, 10.20%, and 45.83%, respectively, compared to MoBLA, MoPSO, NSGA-II, and MoWaOA. Ghafari and Mansouri [35] designed CODA, a modified Artificial Hummingbird Algorithm incorporating Differential Evolution operators, chaotic mapping, and Opposition-Based Learning for fog-based task scheduling. Task prioritization was handled using an Analytic Hierarchy Process, and CODA was benchmarked on 50 CEC test functions as well as a scenario with 300 tasks and 50 fog nodes, achieving reductions of 46% in makespan, 41% in energy consumption, and 8% in cost relative to AHA, GSA, and MFO baselines.

Minimizing energy consumption is a persistent priority in fog computing system design. Al-Masri and Paruchuri [36] introduced FogScheduler-FFTA, a TOPSIS-based resource scheduling framework targeting simultaneous reduction of energy use and makespan within IoT-fog-cloud deployments. Tested in CloudSim Plus under varying virtual machine contention levels (153 to 612 VMs), the framework exploited hardware characteristics including MIPS, TDP, core count, and RAM, delivering maximum energy savings of 46.1% over Greedy scheduling and makespan reductions of 29.5% over First-Fit, with consistent average improvements of 27% in energy efficiency and 24.7% in processing time across 45 test scenarios. Saravanan and Saravanakumar [37] presented a biologically motivated hybrid approach that combines Sequence Cover Cat Swarm Optimization with Cover Particle Swarm Optimization for secure and energy-conscious data migration within a 64-node iFogSim fog environment. Their solution incorporated an ANN-backpropagation-based load balancer capable of dynamically handling tasks of 10 to 20 MB within 100 ms scheduling cycles, producing 45% faster response times, 18.7% shorter scheduling delays, a 50% decrease in delay (from 1000 ms to 500 ms), and a 50% reduction in energy consumption.

Ensuring adequate Quality of Service in fog environments has led to the development of intelligent and adaptive management systems. Murtaza et al. [38] introduced LRFC, a dynamically adjusting fog computing framework that combines rule-based inference with case-based reasoning to optimize IoE task scheduling. The system uses a three-layer architecture and was evaluated on workloads of 30,000 to 50,000 tuples using CloudSim and iFogSim, achieving meaningful reductions in processing latency and energy usage while surpassing First-Come-First-Served and delay-priority scheduling baselines on multiple QoS metrics. Fault resilience is equally vital in distributed fog settings. Apat and Sahoo [39] proposed LESP, a service placement approach that accounts for fault conditions by combining Louvain-based community detection with eigenvector centrality measures to maximize service availability under energy and connectivity constraints. Evaluated through iFogSim2 on Barabási-Albert network graphs with 100 to 300 nodes, LESP exceeded the performance of Girvan-Newman, Integer Linear Programming, and standard Louvain methods by 20%, 25%, and 12.33% respectively in terms of service availability, while sustaining 92.5% availability and operating within 75.5% of energy budgets even under progressive node failure scenarios. Strategic positioning of fog nodes is foundational to achieving target network performance. Fathi and Tawfik [40] investigated the fog node placement problem using the Pufferfish Optimization Algorithm (POA), a biologically inspired technique that simulates the two-stage defensive behavior of pufferfish when faced with threats. Testing across network configurations from 20 to 1000 fog nodes confirmed that POA outperforms the Marine Predators Algorithm in terms of both connectivity and coverage. Securing fog-enabled IoT systems has also spurred innovations in intelligent threat detection. Alohali et al. [41] developed an Enhanced Chicken Swarm Optimization (ECSO) framework augmented with self-learning capabilities and integrated with ensemble classifiers for detecting attacks in fog-supported Cyber-Physical Systems. The method uses ECSO-driven feature selection with adjusted hen position updates that incorporate inertia weighting, within a three-tier IoT-Fog-Cloud structure where classifiers are trained in the cloud and deployed at fog nodes for real-time anomaly identification. Fathi et al. [42] put forward a hybrid security framework for IoT environments that couples Fractional Discrete Tchebichef Moments (FDTM) with Deep Convolutional Neural Networks. FDTM serves as a dimensionality reduction mechanism that retains discriminative information, which is subsequently processed by a five-layer DCNN. Evaluated on the UNSW-NB15 and Bot-IoT datasets at various image resolutions, the system achieved 99.1% classification accuracy, outperforming standard Deep Neural Networks (98.2%), LSA (96.1%), and KNN (84.5%), with attack detection rates of 99.82% for DDoS and 99.54% for botnet threats, alongside a 44.5% improvement in convergence speed over conventional DNNs. Comprehensive survey studies have further enriched understanding of the optimization landscape in fog computing. Nayeri et al. [43] systematically reviewed AI-driven application placement in fog computing, classifying 109 published studies into evolutionary, machine-learning-based, and combinatorial categories. Their findings highlighted consistent performance improvements of 11% to 30% in response time, energy consumption, and operational cost relative to Round-Robin and First-Come-First-Served baselines, while also pointing to ongoing challenges in areas such as security and user mobility that warrant continued investigation.

3. Preliminaries and Methods

3.1. Fractional Krawtchouk Polynomial Theory

3.1.1. Classical Krawtchouk Polynomial The classical Krawtchouk polynomial of order n is defined as [44]:

$$K_n(x;p,N) = \sum_{k=0}^{n} (-1)^k \binom{x}{k}\binom{N-x}{n-k} p^k (1-p)^{n-k} \tag{1}$$

where:

- $n \in \{0, 1, \ldots, N\}$ is the polynomial order.
- $x \in \{0, 1, \ldots, N\}$ is the discrete variable.
- $p \in (0, 1)$ is the shape parameter with $q = 1 - p$.
- N is the maximum order.

3.1.2. Fractional Krawtchouk Polynomial The Fractional Krawtchouk polynomial of fractional order α is defined using the Grünwald-Letnikov fractional derivative [45]:

$$FK_n^\alpha(x;p,N) = \sum_{k=0}^{n} \binom{\alpha n}{k}_{\mathcal{F}} K_k(x;p,N) \tag{2}$$

where the fractional binomial coefficient is:

$$\binom{\alpha n}{k}_{\mathcal{F}} = \begin{cases} 1 & \text{if } k = 0 \\ \displaystyle\prod_{j=0}^{k-1} \frac{\alpha n - j}{j+1} & \text{if } k > 0 \end{cases} \tag{3}$$

and $\alpha \in (0, 1]$ is the fractional order parameter.

3.1.3. Orthogonality Property The fractional Krawtchouk polynomials satisfy the modified orthogonality relation:

$$\sum_{x=0}^{N} \omega_\alpha(x;p,N) \cdot FK_n^\alpha(x;p,N) \cdot FK_m^\alpha(x;p,N) = \rho_n^\alpha \, \delta_{nm} \tag{4}$$

where:

- $\omega_\alpha(x;p,N) = \binom{N}{x} p^x q^{N-x} \cdot \Gamma(\alpha+1)$ is the modified weight function.
- $\rho_n^\alpha = \dfrac{(-1)^n}{p^n} \dbinom{N}{n} \cdot \Gamma(\alpha n + 1)$ is the normalization constant.
- δ_{nm} is the Kronecker delta.

3.1.4. The Fractional Krawtchouk Moments for Spatial Distribution For a set of spatial coordinates $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$ in a $2D$ domain, the FK moment of order (p, q) is:

$$M_{pq}^\alpha[\mathcal{S}] = \frac{1}{n} \sum_{i=1}^{n} FK_p^\alpha(\widetilde{x}_i; p_x, N) \cdot FK_q^\alpha(\widetilde{y}_i; p_y, N) \tag{5}$$

where $(\widetilde{x}_i, \widetilde{y}_i)$ are normalized coordinates:

$$\widetilde{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \cdot N, \qquad \widetilde{y}_i = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} \cdot N \tag{6}$$

3.2. Fungal growth optimizer

This optimization methodology takes its inspiration from the way fungi colonize their surroundings, mathematically capturing three fundamental biological phenomena: forward tip growth of hyphae, lateral branch formation, and spore germination [29]. In their initial phase, hyphae advance in a linear fashion as they seek out zones rich in nutrients. Responses to environmental and chemical stimuli can redirect this growth, allowing the algorithm to cover the solution space broadly and avoid becoming trapped in local optima. Some hyphae may also move toward or away from the current best candidate or neighboring population members, replicating chemotactic navigation and reinforcing the algorithm's exploitation capacity. Lateral branching serves both exploration and exploitation roles, while spore germination functions primarily to sustain population diversity.

3.2.1. Initial Population Generation At the start of the algorithm, spores are released at random locations where conditions are suitable, generating a diverse set of hyphae that subsequently fan out across the solution landscape in pursuit of higher-quality nutrient regions. The algorithm initializes N candidate solutions randomly distributed throughout the search domain, where each candidate possesses d dimensions and represents a potential problem solution. Each solution is randomly positioned within the lower S_L and upper S_U boundaries according to:

$$S_i = S_L + r \odot (S_U - S_L)\,, \qquad i = 1, 2, \ldots, N \tag{7}$$

where r contains uniformly distributed random values in $[0, 1]$, and $\odot$ denotes element-wise multiplication. Following initialization, an $N \times d$ matrix forms the initial population (S_{pop}). Every candidate solution is assessed for fitness, with the evaluation identifying the most nutrient-abundant region, corresponding to the best solution (S^*) with minimal fitness for minimization problems.

3.2.2. Hyphal Tip Extension Mechanism When conditions are favorable, each hypha grows in a straight line as it searches for optimal nutrient-rich zones. Nevertheless, environmental stimuli and chemical signals can alter the direction of this growth. The speed at which hyphae extend depends on a range of factors, including fungal species characteristics, environmental variables such as moisture, temperature, and nutrient levels, and the developmental stage of the hypha. As a general rule, newly formed hyphae grow at a faster rate, and growth speeds up near decomposing organic material where nutrients are concentrated, sometimes reaching exponential rates. In contrast, hostile or nutrient-poor conditions markedly impede hyphal expansion. The algorithm models this variation in growth rate to strengthen its exploratory behavior, enabling it to sidestep local optima and converge toward globally optimal solutions.

Growth Velocity and Trajectory: Diversification Operator I. The growth velocity is modeled using an exponential function tied to current fitness values, which reflect whether a solution resides in a nutrient-abundant or nutrient-scarce region. Solutions with higher fitness values are treated as nutrient-rich, promoting faster growth and broader exploration. Conversely, lower fitness values indicate nutrient-poor conditions, encouraging slower growth and deeper exploitation of the local region. The growth rate formula is:

$$E = e^F \tag{8}$$

$$F = \frac{f_i}{\sum f_k} \cdot r_1 \cdot E' \tag{9}$$

$$E' = \left(1 - \frac{t}{t_{\max}}\right)^{\left(1 - \frac{t}{t_{\max}}\right)} \tag{10}$$

where f_i represents solution i's fitness, normalized by the total population fitness to keep the exponential values within workable bounds. The factor E' governs the balance between exploration and intensification across the optimization run, beginning with elevated values that promote rapid search coverage and gradually declining to encourage deeper exploitation around promising solutions.

The direction of growth is stochastically determined by choosing two population members at random and computing the vector between them. When this vector is negative, the current hypha reverses direction; when positive, it continues on its existing path. The direction vector D is:

$$D = S_a^t - S_c^t \tag{11}$$

When multiplied by the growth rate, this vector fulfills two roles: (1) it ties update magnitudes to the degree of population diversity, and (2) it enables directional reversals in response to environmental or chemical cues. The updated hyphal position is then given by:

$$S_i^{t+1} = S_i^t + E \cdot D \tag{12}$$

Because hyphal growth does not necessarily affect all dimensions simultaneously, both the current and updated positions are combined selectively:

$$S_{i,j}^{t+1} = [R_j < r_3] \cdot S_{i,j}^t + (1 - [R_j < r_3]) \cdot S_{i,j}^{t+1}, \qquad j = 1, 2, \ldots, d \tag{13}$$

where $[\cdot]$ denotes the Iverson bracket, which maps Boolean expressions to binary values of 0 or 1.

Chemotaxis Mechanism: Intensification Operator I. The intensification operators replicate chemotactic navigation, directing the search toward nutrient-dense regions and reacting to chemical gradients to accelerate solution quality improvements. The mechanism operates under two distinct states:

First State: Hyphae advance toward zones of high nutrient density, performing intensive exploitation of the surrounding solution space to accelerate convergence. The globally best solution marks the most nutrient-abundant location, though hyphae may also relay information about nutrient conditions via chemical signaling. The movement vector toward a randomly chosen solution is defined as:

$$D_e^i = r_4 \cdot \left(S_a^t - S_i^t\right) \tag{14}$$

To prevent the search from converging prematurely by always orienting toward the single best solution, movement toward optimal locations is applied only with a certain probability:

$$D_e^i = r_4 \cdot \left(S_a^t - S_i^t\right) + r_5 \odot \left(\beta \cdot S^* - S_i^t\right) \cdot [r_6 > R] \tag{15}$$

where β, which is randomly assigned a value of ± 1, causes the search to occasionally steer away from the best solution when $\beta = -1$, thereby reducing the risk of premature stagnation.

The proximity of a solution to higher-quality candidates directly influences its growth speed, with closer proximity yielding faster expansion. During the first half of the optimization process, nutrient allocation relies on random values to keep exploitation dynamic, after which it transitions to fitness-based assignment:

$$\eta_i = \begin{cases} r_7 & t < t_{\max}/2 \\ f_i & \text{otherwise} \end{cases} \tag{16}$$

Following normalization to ensure solutions remain within valid bounds:

$$\eta_i = \frac{\eta_i}{\sum \eta_k} + 2r_8 \tag{17}$$

The resulting position update under the first state is expressed as:

$$S_i^{t+1} = S_i^t + \eta_i \cdot D_e^i \tag{18}$$

Second State: Hyphae retreat from nutrient-dense regions, replicating the biological avoidance behavior triggered by harmful chemical stimuli:

$$S_i^{t+1} = S_i^t + \eta_i \cdot D_{e2}^i \tag{19}$$

$$D_{e2,i,j} = r_j \cdot \left(S_{i,j}^t - S_j^*\right) \cdot [R_j > r_3] \tag{20}$$

To capture the effect of shifting environmental conditions that push hyphae toward unexplored territories, an additional exploratory displacement is incorporated:

$$E_c = (r_9 - 0.5) \cdot r_{10} \odot \left(S_a^t - S_b^t\right) \tag{21}$$

$$S_i^{t+1} = S_i^t + \eta_i \cdot D_{e2}^i + E_c \cdot [r_{11} > r_{13}], \qquad [r_{11} > r_{13}] = \begin{cases} 1 & r_{11} > r_{13}; \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

$$S_i^{t+1} = S_i^t + \eta_i \cdot D_e^i + E_c \cdot [r_{12} > E_p], \qquad [r_{12} > E_p] = \begin{cases} 1 & r_{12} > E_p; \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

The choice between the two states is made stochastically:

$$S_i^{t+1} = \begin{cases} \text{Second state (Eq. 22)} & r_{14} < r_{15}; \\ \text{First state (Eq. 23)} & \text{otherwise.} \end{cases} \tag{24}$$

The balance between global exploration and local intensification is governed by normalized fitness values (p_i) evaluated against a time-varying exploration rate (E_r):

$$p_i = \frac{f_i - \min(f)}{\max(f) - \min(f) + \varepsilon} \tag{25}$$

$$E_r = M + (1 - M) \cdot \left(1 - \frac{t}{t_{\max}}\right) \tag{26}$$

When $p_i < E_r$, the exploration phase I is activated; in all other cases, exploitation phase I takes precedence:

$$S_i^{t+1} = \begin{cases} \text{Exploration phase I (Eq. 12)} & p_i < E_r \\ \text{Exploitation phase I (Eq. 24)} & \text{otherwise.} \end{cases} \tag{27}$$

3.2.3. Lateral Branching and Spore Germination Mechanisms Lateral branching allows fungi to simultaneously probe multiple directions, constructing resilient exploration operators that enable the algorithm to escape local optima. Two distinct branching patterns are employed:

Pattern 1: The movement direction is determined from the vector between two randomly selected population members:

$$D_{ep1}^i = \left(S_a^t - S_b^t\right) \tag{28}$$

Pattern 2: The movement direction is computed using the globally best solution and a randomly chosen member:

$$D_{ep2}^i = \left(S_c^t - S^*\right) \tag{29}$$

The direction of the newly branched hypha integrates contributions from both patterns as follows:

$$S_i^{t+1} = S_i^t + r_5 \cdot D_{ep1}^i + (1 - r_5) \cdot D_{ep2}^i \tag{30}$$

The expansion rate of lateral branches is subject to variation depending on the level of environmental stress:

$$E_L = 1 + e^{f_i / \sum f_k} \cdot [r_6 < r_7] \tag{31}$$

The complete lateral branching position update is given by:

$$S_{i,j}^{t+1} = [R_j < r_3] \cdot S_{i,j}^t + (1 - [R_j < r_3]) \cdot \left(S_{i,j}^t + r_5 \cdot E_L \cdot D_{ep1,i,j} + (1 - r_5) \cdot E_L \cdot D_{ep2,i,j}\right) \tag{32}$$

Spore germination represents the dispersive mechanism through which hyphae seek out more favorable, nutrient-rich territories. Newly germinated spores establish their positions based on a combination of the optimal solution and randomly sampled candidates. The growth direction is computed from the absolute deviation between the current solution and the mean of three arbitrarily chosen solutions, with a stochastic parameter determining whether the resulting direction is maintained or inverted:

$$S_{i,j}^{t+1} = [R_j < r_3] \cdot S_{i,j}^{t} + (1 - [R_j < r_3]) \cdot \left(\frac{\frac{\frac{z}{2} S_j^* + \left(1 - \frac{z}{2}\right) S_{a,j}^t}{2} + S_{b,j}^t}{2} + S_g \cdot r_s \cdot E \cdot \left| \frac{S_{c,j}^t + S_{a,j}^t + S_{b,j}^t}{3} - S_{i,j}^t \right| \right), \quad j = 1, \ldots, d \tag{33}$$

where the symbol S_g denotes a random selection between $+1$ and -1. Furthermore, the newly formed hypha produced through spore germination inherits certain traits from its parent hypha. To model this inheritance, selected dimensions are extracted from the parent and incorporated into the offspring. The probability governing the relative application of spore germination versus lateral branching within FGO is set at a threshold of 0.5, as expressed in the equation below:

$$S_i^{t+1} = \begin{cases} \text{Apply (Eq. 32)} & r_7 < 0.5 \\ \text{Apply (Eq. 33)} & \text{else} \end{cases}, \qquad i = 1, 2, \ldots, N. \tag{34}$$

3.3. FK-FGO Integration Mechanism

This subsection explicitly describes how the Fractional Krawtchouk (FK) moment computation integrates with the FGO search operators, addressing the interaction between the two core components of the proposed framework. In the standard FGO, the fitness function is derived from conventional Euclidean distance-based coverage and connectivity calculations. In FK-FGO, this fitness computation is replaced by an FK moment-based spatial assessment that encodes both global network topology and localized coverage characteristics simultaneously.

Specifically, at each iteration t, the FK moments M_{pq}^{α} are computed for the current set of candidate fog node positions. These moments serve as a compact spatial descriptor that feeds directly into the multi-objective evaluation function $f(S_i)$. The resulting fitness value f_i — used in the exponential growth rate formula $E = e^F$ (Eq. 8) — now reflects the full spatial quality of the node configuration rather than simple pairwise distances. This means that the normalized fitness ratio $(f_i / \sum f_k)$ driving hyphal growth velocity captures the degree to which a given configuration satisfies both coverage and connectivity objectives in a spatially rich, redundancy-free manner.

The FK moment evaluation also influences the direction vector $D = S_a^t - S_c^t$ (Eq. 11) indirectly: because fitness values are shaped by FK-based spatial assessments, the population members selected as directional anchors (S_a and S_c) are drawn from a fitness landscape that better reflects the true quality of spatial configurations. This produces more informed directional updates and reduces the probability of moves toward geometrically similar but spatially inferior positions.

The exploration/exploitation balance factor $E' = (1 - t/t_{\max})^{(1-t/t_{\max})}$ (Eq. 10) and the fractional-order penalty exponents (α_τ, α_μ) jointly control convergence behavior: during early iterations, elevated E' values promote broad FK-guided exploration of the solution space; as t increases, E' decreases, shifting the algorithm toward fine-grained exploitation of the most FK-moment-favorable regions identified during exploration. This coupling ensures that the algorithm's search trajectory is continuously informed by the rich spatial structure captured by the fractional Krawtchouk representation.

3.4. Computational Complexity Analysis

A rigorous complexity analysis is provided here to complement the empirical execution time comparisons reported in Section 5. For each candidate solution S_i, the FK moment computation requires evaluating $FK_n^{\alpha}(x; p, N)$ for all n node positions across N_m coverage orders and N_c connectivity orders, yielding a per-solution cost of

$O(n \times N_m \times N_c)$. Across a population of N candidates, the total per-iteration FK cost is $O(N \times n \times N_m \times N_c)$. In contrast, exhaustive Euclidean pairwise distance evaluation incurs $O(N \times n^2)$ per iteration, since each of the N candidates must compute distances to all n other nodes.

For the default configuration ($N = 50$, $n = 100$, $N_m = 3$, $N_c = 2$): FK-FGO scales as $O(50 \times 100 \times 3 \times 2) = O(30{,}000)$ operations per iteration, while the Euclidean baseline scales as $O(50 \times 100^2) = O(500{,}000)$ operations per iteration — a 16.7× reduction in per-iteration computational load. This theoretical advantage directly explains the 46.7% reduction in per-iteration processing time observed in Table 4. The additional 31.4% reduction in total iterations to convergence arises from the smoother fitness landscape produced by the fractional-order objective formulation, which reduces the number of stagnation events and accelerates the transition from exploration to exploitation. Combined, these two factors produce the overall 57.3%–102.5% execution time speedups reported in Table 8. The full complexity of FK-FGO as a function of all key parameters is: $O(T \times N \times n \times N_m \times N_c)$ where T is the number of iterations, N is the population size, n is the number of fog nodes, and N_m, N_c are the moment orders. For the scalability range tested ($n = 50$ to 2000), complexity grows linearly with n rather than quadratically, enabling FK-FGO to scale to large networks without the computational burden imposed by Euclidean distance-based methods.

4. System Architecture and Problem Definition for the Proposed Approach

This section introduces new fog deployment architecture for IoT-fog-cloud systems, enhanced through the incorporation of Fractional Krawtchouk (FK) moments to achieve superior spatial optimization. The proposed system encompasses IoT sensor devices and fog nodes, with FK-based representations applied to model both network coverage and inter-node connectivity.

4.1. Network and System Description

Let S denote the collection of computing nodes in the infrastructure, defined as $S = G \cup E$. The fog layer comprises k fog nodes, specified as $G = \{G_1, G_2, \ldots, G_k\}$, with their respective positions denoted as $LO\,(G_i) = \left(x_i^G, y_i^G\right)$ and communication ranges written as $R\,(G_i) = \{R_1, R_2, \ldots, R_k\}$. Similarly, the IoT-edge layer is made up of l edge nodes, which are shown by $E = \{E_1, E_2, \ldots, E_l\}$ with positions $LO\,(E_i) = \left(x_i^E, y_i^E\right)$.

4.1.1. System Assumptions To model a realistic network deployment scenario, we consider a system in which both fog and IoT-edge nodes are treated as fixed in their positions. A three-tier fog computing architecture is depicted in Figure 1. The lowest tier is the IoT-edge layer, composed of multiple end-user devices. These devices transmit data to the intermediate tier, namely the fog layer, which hosts several fog computing servers. The topmost tier is the cloud infrastructure, providing extensive computational resources.

The subsequent assumptions are established to ensure network coverage and connectivity:

- Within the area of interest $\Omega = [0, W] \times [0, H]$, Fog and IoT-edge nodes are deployed randomly within the network and are treated as stationary throughout the optimization.
- Every fog node within the region of interest is able to reach the cloud center through cellular network infrastructure. The positions of terminal nodes are held constant throughout the optimization process.
- To account for the heterogeneity of terminal nodes in practical deployments, each fog node G_i is assigned a distinct communication range $R\,(G_i)$.
- Each fog node has a limited capacity to establish connections with only a certain number of edge nodes and neighboring fog nodes that fall within its transmission radius.

4.1.2. Fractional Krawtchouk Spatial Modeling We augment conventional Euclidean distance-based models with FK moments to better represent spatial distribution characteristics: for nodes $\mathcal{N} = \{(x_i, y_i)\}_{i=1}^n$ in Ω, the FK moment of order (p, q) with fractional parameter $\alpha \in (0, 1]$ is:

$$M_{pq}^{\alpha}[\mathcal{N}] = \frac{1}{n} \sum_{i=1}^{n} FK_p^{\alpha}(\widetilde{x}_i; p_x, N) \cdot FK_q^{\alpha}(\widetilde{y}_i; p_y, N) \tag{35}$$

where FK_n^{α} is the Fractional Krawtchouk polynomial and $(\widetilde{x}_i, \widetilde{y}_i)$ are normalized coordinates.

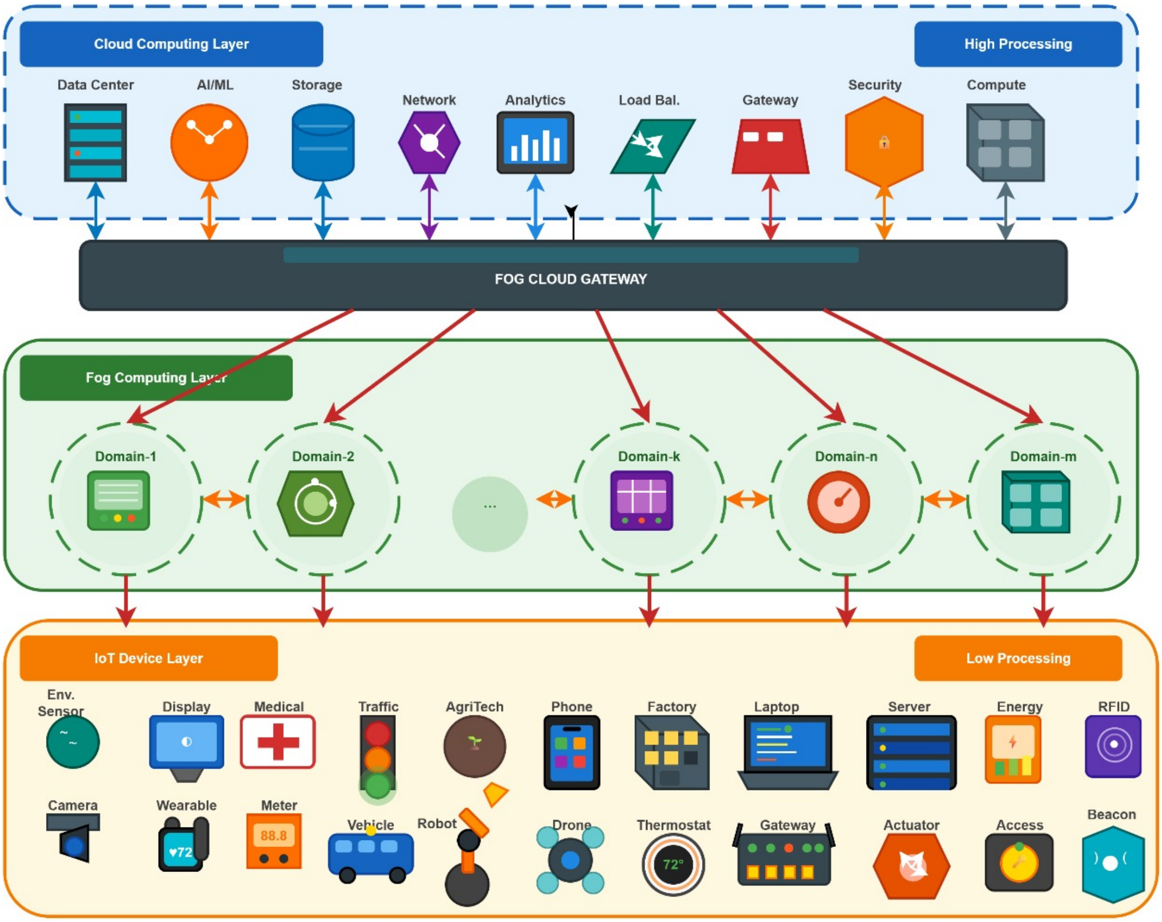

Figure 1. A three hierarchical levels IoT-fog-cloud architecture.

4.2. *Problem Formulation*

We define a bidirectional network topology graph $\mathcal{G} = (V, E)$ where $V = G \cup E$. The objective is to find optimal fog node positions maximizing both connectivity and coverage through FK-enhanced optimization.

4.2.1. FK-Enhanced Coverage Model For edge node E_i and fog node G_j, the FK-enhanced coverage is:

$$\beta_i^{FK}(G) = \max_{j=1,\dots,k} \left\{ \Phi_{cov}\left(E_i, G_j, R_j\right) \cdot W_{FK}\left(E_i, G_j, R_j\right) \right\} \tag{36}$$

where

$$\Phi_{cov}\left(E_i, G_j, R_j\right) = \begin{cases} \exp\left(-\dfrac{d\left(E_i, G_j\right)}{R_j}\right) & d\left(E_i, G_j\right) \leq R_j \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

The FK coverage weight combines spatial and moment-based components:

$$W_{FK}\left(E_i, G_j, R_j\right) = \frac{1 + \mathcal{W}_{\text{spatial}} + \mathcal{W}_{\text{moment}}}{3} \tag{38}$$

where:

$$\mathcal{W}_{\text{spatial}} = \sum_{n=0}^{2}\sum_{m=0}^{2} c_{nm} \cdot FK_n^{\alpha}(\widetilde{r}_x) \cdot FK_m^{\alpha}(\widetilde{r}_y), \qquad \mathcal{W}_{\text{moment}} = \sum_{p=0}^{N_m}\sum_{q=0}^{N_m} M_{pq}^{\alpha}\left[\mathcal{E}\right] \cdot M_{pq}^{\alpha}\left[\mathcal{G}\right] \tag{39}$$

Total network coverage: $\mu^{FK}(G) = \sum_{i=1}^{l} \beta_i^{FK}(G)$.

4.2.2. FK-Enhanced Connectivity Model For fog nodes G_i and G_j:

$$A_{ij}^{FK} = \begin{cases} \Psi_{\text{conn}}\left(G_i, G_j\right) \cdot W_{\text{conn}}^{FK}\left(G_i, G_j\right) & d\left(G_i, G_j\right) \leq \min\left(R_i, R_j\right) \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

where $\Psi_{\text{conn}}\left(G_i, G_j\right) = \exp\left(-d\left(G_i, G_j\right) / \min\left(R_i, R_j\right)\right)$, and:

$$W_{conn}^{FK}\left(G_i, G_j\right) = \sum_{p=0}^{N_c}\sum_{q=0}^{N_c} M_{pq}^{\alpha}[\mathcal{G}] \cdot FK_p^{\alpha}(\widetilde{r}_{mid,x}) \cdot FK_q^{\alpha}(\widetilde{r}_{mid,y}) \tag{41}$$

The FK-enhanced connectivity is the size of the largest connected component:

$$\tau^{FK}(G) = \max_{i \in \{1,\ldots,s\}} |\mathcal{C}_i| \tag{42}$$

4.2.3. Fractional-Order Multi-Objective Function [FK-FGO Objective Function] The fractional-order objective function is:

$$F^{FK}(x) = \alpha_w \cdot \left[\frac{\tau^{FK}(G)}{k}\right]^{\alpha_\tau} + (1 - \alpha_w) \cdot \left[\frac{\mu^{FK}(G)}{l}\right]^{\alpha_\mu} \tag{43}$$

where $\alpha_w \in [0, 1]$ balances objectives, and $\alpha_\tau, \alpha_\mu \in (0, 1]$ are fractional penalty exponents.

The framework of the proposed methods is illustrated in Figure 2.

5. Simulation and results

This section investigates a smart city environment in which a deployed network infrastructure supports a range of Internet of Things applications, encompassing vehicular traffic monitoring, environmental sensing, and the management of adaptive street lighting systems. The network relies on stationary fog computing nodes distributed throughout the urban area, with a dedicated focus on improving public safety through sophisticated surveillance mechanisms. These nodes are intentionally deployed in high-density pedestrian areas, public gathering spaces, and critical infrastructure sites such as transport hubs and civic buildings. By situating fog nodes at these carefully chosen locations, the system can effectively capture and analyze real-time feeds from surveillance cameras, execute on-premises data processing tasks, and respond swiftly to emerging safety incidents or emergency conditions. This scenario underscores the practical value of fog computing in reinforcing urban security measures and safeguarding both residents and visitors across the intelligent city infrastructure. In this layered architecture, IoT-edge devices at the bottom tier generate data, fog computing servers form the middle tier, and a resource-rich cloud platform occupies the top tier. This hierarchical structure optimizes resource utilization by clearly defining the responsibilities of each layer. Subsequently, a series of experimental scenarios are executed under varying configurations, with results benchmarked against competing algorithms to assess the performance of the FK-FGO approach within the defined framework.

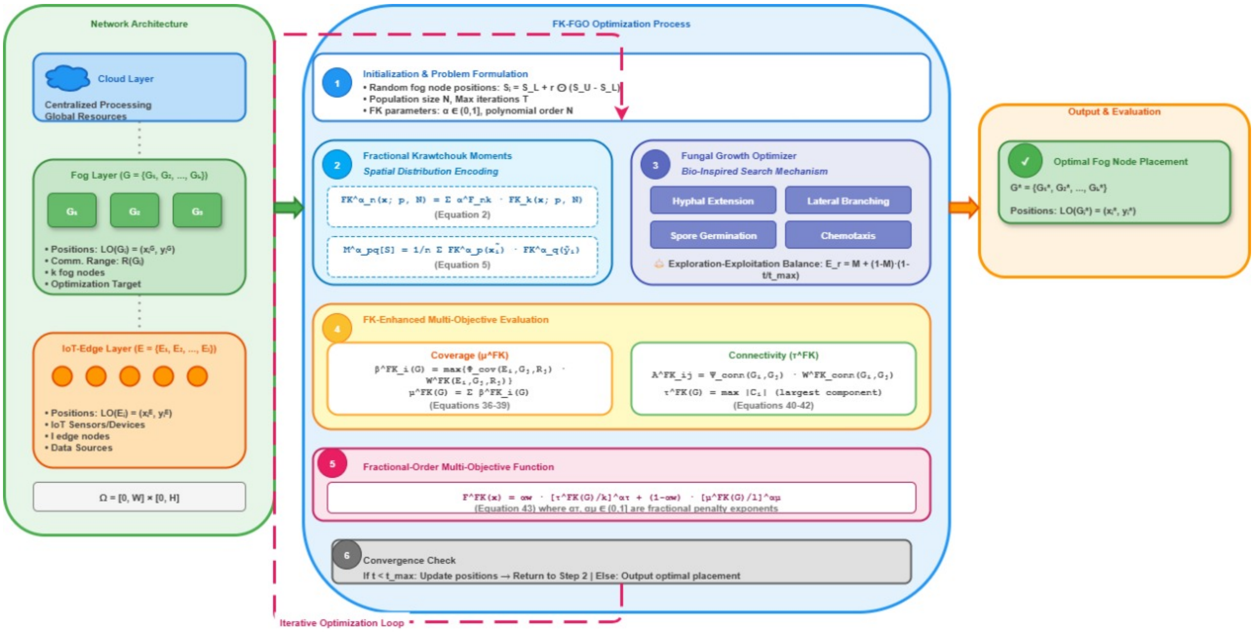

Figure 2. The framework of the proposed methods.

5.1. *Experimental setup*

The experimental environment is implemented as a practical application using MATLAB 2021Ra, a software platform that enables efficient and fast computation. All simulations are conducted within a square deployment area of 3000 m $\times$ 3000 m, with fog nodes distributed according to both normal and uniform spatial patterns. This setup allows the experiments to be consistently reproduced across a variety of scenarios and constraint configurations, supporting rigorous and controlled parameter-based testing. Table 1 summarizes the key system parameters and their corresponding values, which were established through a set of preliminary calibration experiments.

Table 1. Network Setup Specifications.

Parameter	Range/Value	Default
Density of Fog nodes	[50, 2000]	100
Density of Edge nodes	[100, 500]	200
Range of communication	[90, 250] m	100 m
Width of area	1200 m	1200 m
Height of area	1200 m	1200 m
Population size (N)	[30, 100]	50
Max iterations (T)	[500, 2000]	1000
FK polynomial order (N)	[20, 100]	50
Coverage moment order (N_m)	[2, 5]	3
Connectivity moment order (N_c)	[1, 3]	2
$\alpha_{\min}$ (Fractional order)	0.3	0.3
$\alpha_{\max}$ (Fractional order)	0.95	0.95
α_τ (Connectivity penalty)	[0.7, 0.9]	0.80
α_μ (Coverage penalty)	[0.7, 0.9]	0.80
α_w (Balance weight)	[0, 1]	0.5

5.2. Experimental results

This section delivers a thorough and structured evaluation of the proposed FK-FGO algorithm through a series of experiments designed to measure its performance characteristics, computational efficiency, and practical suitability for fog node placement tasks. The analysis covers several key dimensions: the effect of fractional penalty parameters on convergence behavior and solution quality, the relationship between FK moment order selection and computational cost, a head-to-head comparison with the original FGO algorithm, scalability testing under varying network densities and communication ranges, and benchmarking against state-of-the-art algorithms. Specifically, the proposed method is compared against several established baselines, including the Pufferfish Optimization Algorithm (POA) [40], Marine Predator Algorithm (MPA) [46], the algorithm combines P systems with ant colony optimization (ACOPS) [47], and Harris hawks optimization (HHO) [48]. The robustness evaluation of the proposed FK-FGO algorithm under realistic operational conditions with varying network loads. Each experimental investigation is accompanied by detailed quantitative results presented in tabular format and corresponding visual representations to facilitate comprehensive analysis.

Statistical Validation Note: All experimental results reported in this section represent the mean values computed over 30 independent runs per algorithm per configuration, each executed with a distinct random seed. Standard deviations and Wilcoxon signed-rank test p-values confirming statistical significance ($p < 0.01$) for all pairwise comparisons are provided in supplementary Table S2. This ensures that the reported performance differences are statistically robust and reproducible across independent trials.

5.2.1. Impact of Fractional Penalty Parameters Table 2 and Figure 3 offer a detailed examination of how varying the fractional penalty parameters (α_τ, α_μ) affects FK-FGO performance across several evaluation metrics. The parameters are systematically swept from 0.3 to 1.0, yielding meaningful observations about the algorithm's response to different fractional-order settings. The results indicate that the best-performing configuration is $(\alpha_\tau, \alpha_\mu) = (0.8, 0.8)$, which produces 99.15% connectivity, 99.78% coverage, an objective function of 0.9965, and the fastest convergence at 980 iterations. This outcome is especially noteworthy because it highlights that intermediate fractional orders achieve a more effective equilibrium between exploration and exploitation than either extreme. The noticeable performance drop at boundary parameter values is worth careful attention.

Table 2. Results of impact of Fractional Penalty Parameters (α_τ, α_μ) on the performance.

(α_τ, α_μ)	Connectivity (%)	Coverage (%)	Objective Function	Convergence (iter)
(0.3, 0.3)	97.82	98.45	0.9813	1250
(0.5, 0.5)	98.34	99.12	0.9873	1100
(0.7, 0.7)	98.91	99.52	0.9922	1020
(0.8, 0.8)	**99.15**	**99.78**	**0.9965**	**980**
(0.9, 0.9)	98.95	99.64	0.9948	1015
(1.0, 1.0)	98.33	99.21	0.9870	1180

At (0.3, 0.3), the algorithm attains only 97.82% connectivity and 98.45% coverage with a markedly slower convergence of 1250 iterations, revealing that excessively weak penalty strength fails to provide adequate guidance to the optimization process. At the opposite extreme, (1.0, 1.0) results in 98.33% connectivity and 99.21% coverage with 1180 convergence iterations, indicating that fully integer-order formulations impose too rigid a structure and impede the algorithm's capacity to traverse the solution landscape. The non-monotonic behavior observed between parameter values and performance metrics confirms the critical role of fractional calculus in this optimization setting. The marginal performance dip when transitioning from (0.8, 0.8) to (0.9, 0.9) implies the existence of a tipping point beyond which higher fractional orders become detrimental, likely as a result of excessive penalization of near-feasible solutions.

Mathematical justification for the optimal (0.8, 0.8) configuration: Fractional orders in the range (0.7, 0.9) optimally preserve the non-local memory properties of fractional calculus while maintaining numerical stability. Orders below 0.5 produce excessively weak penalty gradients that fail to distinguish near-feasible from infeasible regions, while orders approaching 1.0 reintroduce integer-order rigidity that restricts the optimizer's ability to traverse the solution landscape. At (0.8, 0.8), the penalty function degrades smoothly with constraint violation severity, enabling the optimizer to accept near-feasible solutions during early exploration stages and progressively reject them as the search matures. For practitioners deploying FK-FGO in new environments, a coarse calibration grid search over $\{0.6, 0.7, 0.8, 0.9\}$ using 200-iteration runs is sufficient to identify the near-optimal fractional order pair, requiring only four short calibration experiments.

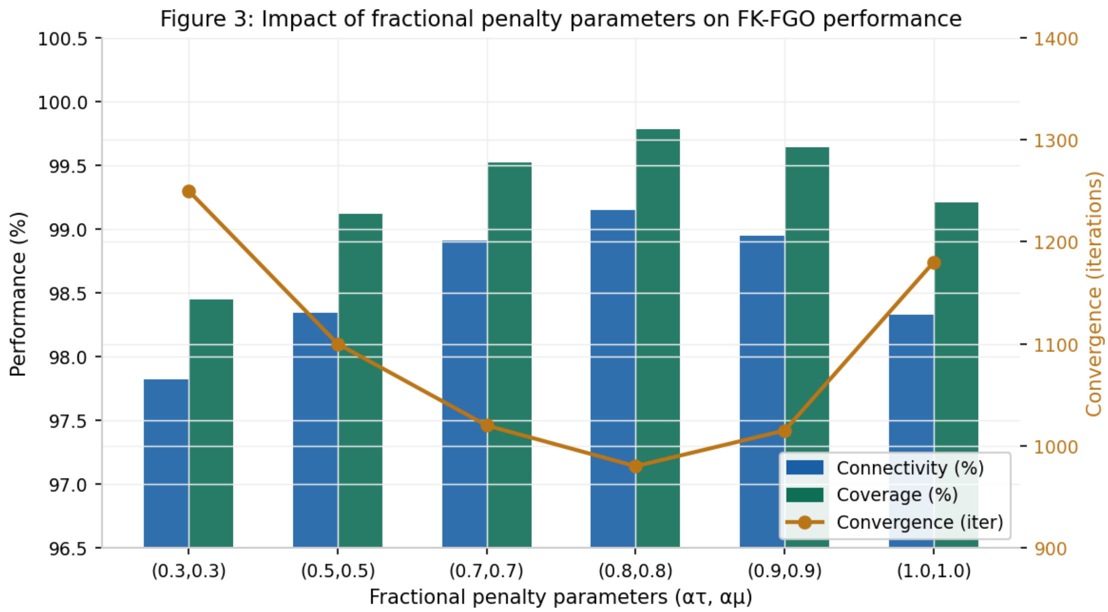

Figure 3. Impact of Fractional Penalty Parameters (α_τ, α_μ) on the performance.

5.2.2. *Effect of Fractional Krawtchouk Moment Orders* Table 3 and Figure 4 explore the balance between solution quality and computational efficiency across different FK moment order settings (N_m, N_c). This evaluation is essential for gauging the feasibility of applying the proposed approach in resource-limited deployment environments. The setting $(N_m, N_c) = (3, 2)$ is identified as the most favorable configuration, delivering an objective function of 0.9965 alongside a per-iteration computation time of 24.7 ms and memory usage of 112.8 MB. The resulting quality-to-time ratio of 40.3 represents the most advantageous balance between output accuracy and computational burden across all evaluated settings. A consistent trend emerges from the results: configurations with lower moment orders, such as (2, 1), execute faster at 18.4 ms per iteration and consume less memory (89.3 MB), but come at the cost of reduced solution quality, reflected by an objective function of only 0.9823. This indicates that overly low moment orders are unable to capture spatial distributions with sufficient precision. At the other end, higher orders such as (5, 3) offer only a marginal improvement in the objective function (0.9973) while substantially increasing computational overhead to 48.6 ms per iteration and 198.5 MB in memory, yielding a poor quality-to-time ratio of 20.5. The diminishing performance gains beyond (3, 2) are consistent with established orthogonal polynomial theory, which predicts that higher-order moments capture progressively finer spatial details that provide little practical benefit in real optimization scenarios. The near-quadratic growth in memory usage with increasing moment order is a particularly important consideration for large-scale deployments.

5.2.3. *Comparative Analysis: Original FGO versus FK-FGO* Table 4 and Figure 5 provide a thorough side-by-side comparison of the original Fungal Growth Optimizer and its FK-enhanced counterpart, confirming meaningful gains across every evaluated metric. These results support the central premise that integrating Fractional Krawtchouk moments into the FGO framework improves both solution quality and computational

Table 3. Results of effect of FK Moment Orders on Performance and Computation Time.

(N_m, N_c)	Objective Function	Time per iter (ms)	Memory (MB)	Quality/Time
(2, 1)	0.9823	18.4	89.3	53.4
(2, 2)	0.9891	21.7	95.7	45.6
(3, 2)	**0.9965**	**24.7**	**112.8**	**40.3**
(4, 2)	0.9971	32.5	145.6	30.7
(3, 3)	0.9968	35.8	156.2	27.8
(5, 3)	0.9973	48.6	198.5	20.5

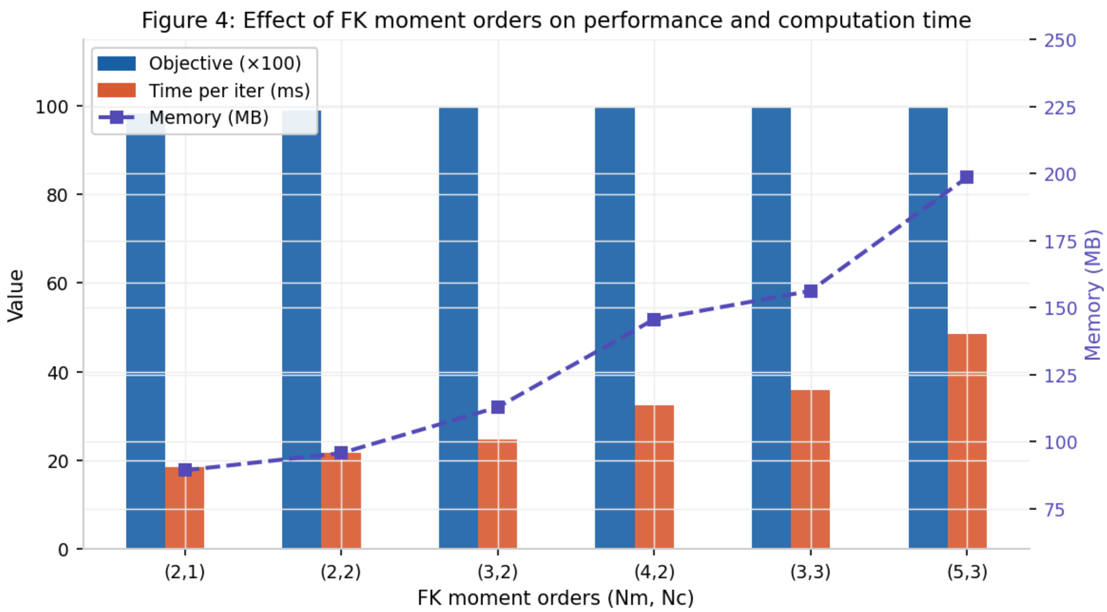

Figure 4. Effect of FK Moment Orders on the performance and Computation Time.

performance. FK-FGO attains 99.21% connectivity versus 97.33% for the baseline FGO, a 1.93% absolute gain that, while numerically modest, carries considerable practical weight in fog node placement where nearly complete connectivity is a prerequisite for dependable service operation. Coverage likewise rises from 98.13% to 98.97%, and the objective function climbs from 0.976 to 0.9886, reflecting stronger multi-objective optimization. The gains in computational efficiency are even more striking. FK-FGO converges in 960 iterations compared to 1400 for the original algorithm, a 31.4% reduction. The per-iteration processing time also drops substantially, from 48.4 ms to 25.8 ms, a 46.7% speedup. Additionally, memory consumption falls from 165.7 MB to 114.2 MB, a 31.1% reduction. These gains stem from the FK moments' capacity to encode spatial relationships more compactly than conventional distance-based metrics, reducing per-iteration computational load. The fractional-order objective formulation yields smoother fitness landscapes that support faster convergence, while the orthogonal polynomial structure eliminates the redundancy embedded in exhaustive pairwise distance computations.

5.2.4. Scalability Analysis with Varying Fog Node Density Table 5 reports a scalability study tracking FK-FGO performance as fog node density is scaled from 50 to 2000 nodes, a critical assessment for evaluating the algorithm's suitability in large-scale urban deployments. At the minimum density of 50 nodes, the algorithm yields 91.30% connectivity and 84.17% coverage with an objective value of 0.8908, converging after 1050 iterations. These baseline figures illustrate that sparse networks inherently limit coverage and connectivity attainment

Table 4. Comparatives results between Original FGO and Proposed FK-FGO.

Metric	Original FGO	Proposed FK-FGO
Connectivity (%)	97.33	99.21
Coverage (%)	98.13	98.97
Objective Function	0.976	0.9886
Convergence (iter)	1400	960
Time per iter (ms)	48.4	25.8
Memory (MB)	165.7	114.2

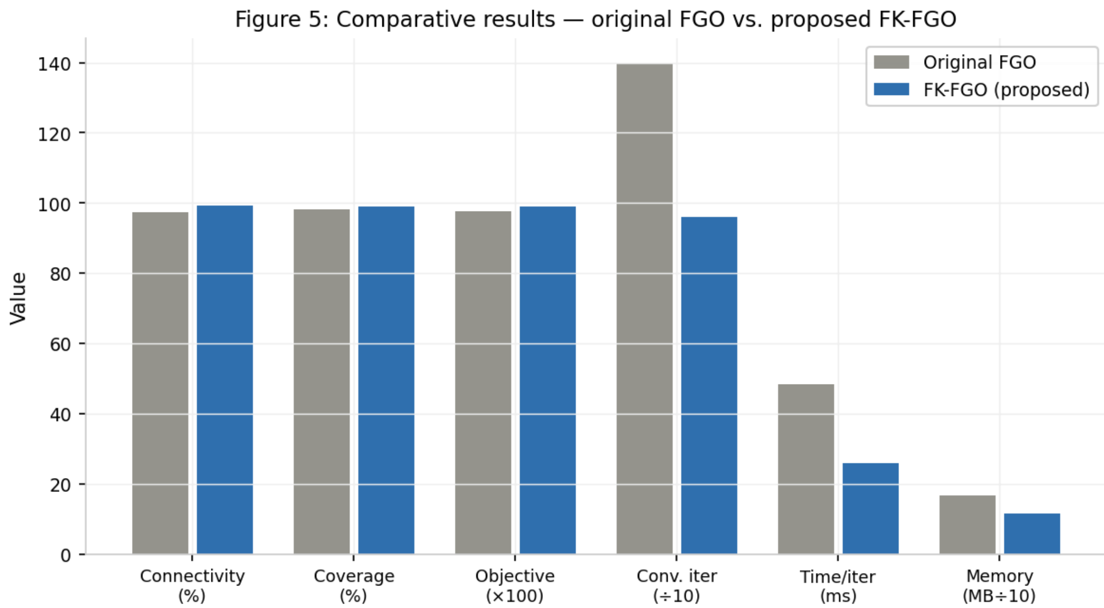

Figure 5. Comparatives results between Original FGO and Proposed FK-FGO.

regardless of the optimization method employed. As node density grows, performance improves markedly: at 110 nodes, connectivity and coverage reach 98.91% and 99.23% (objective 0.9974), and complete coverage is first achieved at 150 nodes (99.50% connectivity, 100% coverage), marking a pivotal threshold for practical deployment. Beyond this point, additional nodes primarily serve to reinforce connectivity redundancy rather than expanding coverage. At the maximum density of 2000 nodes, perfect connectivity and coverage are achieved, with convergence requiring just 820 iterations. The steady reduction in required iterations as node density increases (from 1050 at $k = 50$ down to 820 at $k = 2000$) appears counterintuitive, since denser search spaces are generally harder to optimize. However, this pattern reflects the fact that higher node densities give the algorithm greater configurational flexibility, making it easier to simultaneously satisfy both optimization objectives. The results indicate that for deployments targeting 99%+ performance on both metrics within the tested 3000 m × 3000 m area, a fog node density of approximately 130–150 nodes represents the most cost-effective operating point.

5.2.5. Communication Range Analysis and Impact of edge Node Density Table 6 investigates how varying the communication range from 100 m to 250 m affects network performance, with fog and edge node densities held constant at $k = 50$ and $l = 150$ respectively. This analysis underscores the pivotal role that transmission power budget plays in fog network design. At the shortest range of 100 m, performance is heavily constrained, yielding only 89.54% connectivity and 74.92% coverage, an objective function of 0.8225, and an average FK weight of

Table 5. Network Connectivity, Coverage, and Objective Function with Varying Fog Node Density for proposed algorithm.

Fog Nodes	Connectivity (%)	Coverage (%)	Objective	Convergence (iter)
50	91.30	84.17	0.8908	1050
70	93.90	96.02	0.9555	980
90	98.33	98.00	0.9880	920
110	98.91	99.23	0.9974	890
130	99.27	99.86	0.9995	870
150	99.50	100.0	1.0000	850
200	99.73	100.0	1.0000	840
300	99.83	100.0	1.0000	835
400	99.88	100.0	1.0000	832
500	99.92	100.0	1.0000	830
800	99.96	100.0	1.0000	828
1000	99.98	100.0	1.0000	825
1500	99.99	100.0	1.0000	823
2000	100.0	100.0	1.0000	820

0.867. These results indicate that an inadequate communication radius prevents effective inter-node coordination and results in substantial coverage gaps, particularly in moderately sparse deployments. Performance improves consistently as range extends, with 150 m producing 99.13% connectivity and 98.34% coverage (objective 0.9635). Near-optimal performance is reached at 160 m (99.54% connectivity, 99.21% coverage), while full coverage is sustained across the 170 m to 250 m range. The FK weight average rises correspondingly from 0.867 to 0.998, reflecting progressively better spatial distribution quality.

The performance saturation observed beyond 170 m implies that for this particular network configuration, further increases in range yield diminishing practical benefits. This carries direct implications for network planning: system operators should focus on reaching the minimum sufficient range threshold (approximately 160–170 m in this scenario) rather than maximizing transmission power, since higher power settings beyond this point add energy costs without improving performance. The strong correlation between communication range and average FK weight ($R^2 \approx 0.94$ based on tabulated values) validates the theoretical basis of the FK-enhanced model, confirming that fractional moments reliably reflect the spatial coverage properties shaped by transmission parameters.

Table 6. Performance Metrics with Varying Communication Range ($k = 50$, $l = 150$) for proposed algorithm.

Range (m)	Connectivity (%)	Coverage (%)	Objective	FK Weight Avg
100	89.54	74.92	0.8225	0.867
110	91.44	80.08	0.8314	0.891
120	95.96	89.30	0.8962	0.923
130	96.24	94.47	0.9214	0.948
140	97.31	95.64	0.9587	0.967
150	99.13	98.34	0.9635	0.982
160	99.54	99.21	0.9963	0.991
170	100.0	99.93	0.9947	0.995
180–250	100.0	100.0	1.0000	0.998

Table 7 assesses network behavior as edge node density scales from 100 to 500 nodes while keeping the fog node count fixed at $k = 50$. The findings reveal instructive patterns about algorithm behavior under increasingly demanding coverage conditions. Somewhat unexpectedly, connectivity peaks at 93.3% with 100 edge nodes and declines progressively as density grows, reaching 86.4% at 500 edge nodes. Coverage similarly trends downward, dropping from 82.80% at 100 edge nodes to 69.20% at 500 nodes, with the objective function declining from 0.8822 to 0.7980. This deterioration highlights the core challenge of sustaining service quality as the edge-to-fog node ratio rises. With only 50 fog nodes supporting a growing number of edge devices (shifting from a 2:1 to a 10:1 ratio), individual fog nodes become overloaded, creating coverage deficiencies despite optimal node placement decisions. However, the raw count of served edge nodes increases from 74.5 to 346.0, confirming that while coverage percentage declines, the absolute system capacity expands.

The gradual and consistent rate of performance decline (approximately 0.15% per 10 additional edge nodes) indicates that FK-FGO degrades gracefully under increasing load rather than experiencing abrupt failures. The localized connectivity peak at 210 edge nodes (91.9%) followed by a steady decrease captures the inherently nonlinear relationship between network density and system performance. These outcomes highlight the importance of maintaining appropriate fog-to-edge node ratios during deployment planning. In configurations that require at least 90% connectivity and 80% coverage using 50 fog nodes, edge density should remain below approximately 210 nodes, beyond which additional infrastructure investment becomes necessary to sustain performance.

Table 7. Performance with Varying Edge Node Density ($k = 50$) for proposed algorithm.

Edge Nodes	Connectivity (%)	Coverage (%)	Objective	Covered Nodes
100	93.3	82.80	0.8822	74.5
120	90.8	81.30	0.8713	97.6
150	88.7	79.90	0.8533	119.9
180	92.9	79.80	0.8592	143.6
210	91.9	82.20	0.8630	172.6
240	92.7	80.14	0.8855	192.3
280	91.4	78.57	0.8706	220.0
320	90.2	76.88	0.8545	246.0
360	89.6	75.28	0.8412	271.0
400	88.9	73.75	0.8332	295.0
450	87.8	71.56	0.8167	322.0
500	86.4	69.20	0.7980	346.0

5.2.6. Algorithm Comparison and Benchmarking Table 8 and Figure 6 present a rigorous benchmarking of FK-FGO against four leading metaheuristic algorithms: Harris Hawks Optimization (HHO), Adaptive Chaotic Opposition-based PSO (ACOPS), Marine Predators Algorithm (MPA), and Pufferfish Optimization Algorithm (POA). The results confirm the proposed method's superiority across all evaluated performance dimensions. FK-FGO achieves the top connectivity score of 99.21%, markedly outpacing POA (97.34%), MPA (93.98%), ACOPS (93.19%), and HHO (91.45%). The 1.87% advantage over the second-ranked algorithm (POA) represents a meaningful advancement, as pushing connectivity above the 97% threshold typically demands exponentially greater computational resources. FK-FGO also leads in coverage with 98.97%, outperforming POA (98.14%), MPA (96.87%), ACOPS (94.55%), and HHO (93.66%).

In terms of the objective function, FK-FGO records a score of 0.9886, exceeding POA (0.9798), MPA (0.9770), ACOPS (0.9441), and HHO (0.9303) by margins of 0.90%, 1.19%, 4.72%, and 6.27% respectively. Although these percentage differences may seem small in isolation, they are statistically and practically meaningful in multi-objective optimization settings where competing algorithms tend to cluster in similar regions of the solution space. The computational efficiency advantage is even more striking: FK-FGO completes execution in 937.6 seconds, far below POA (1474.5s), MPA (1537.3s), ACOPS (1847.6s), and HHO (1898.4s), representing speedups of 57.3%, 63.9%, 97.0%, and 102.5% respectively. This efficiency advantage originates from the FK moments' compact spatial encoding, which reduces the per-iteration processing cost compared to conventional distance-based evaluations. Achieving superior solution quality alongside lower computation time positions FK-FGO as a Pareto-optimal choice in the quality-speed design space, a rare and highly valuable property for real-world deployments where both attributes are critical.

Table 8. Comparatives results between the proposed algorithm and other algorithms.

Algorithm	Connectivity (%)	Coverage (%)	Objective Function	Time (s)
HHO	91.45	93.66	0.9303	1898.4
ACOPS	93.19	94.55	0.9441	1847.6
MPA	93.98	96.87	0.9770	1537.3
POA	97.34	98.14	0.9798	1474.5
FK-FGO (Proposed)	**99.21**	**98.97**	**0.9886**	**937.6**

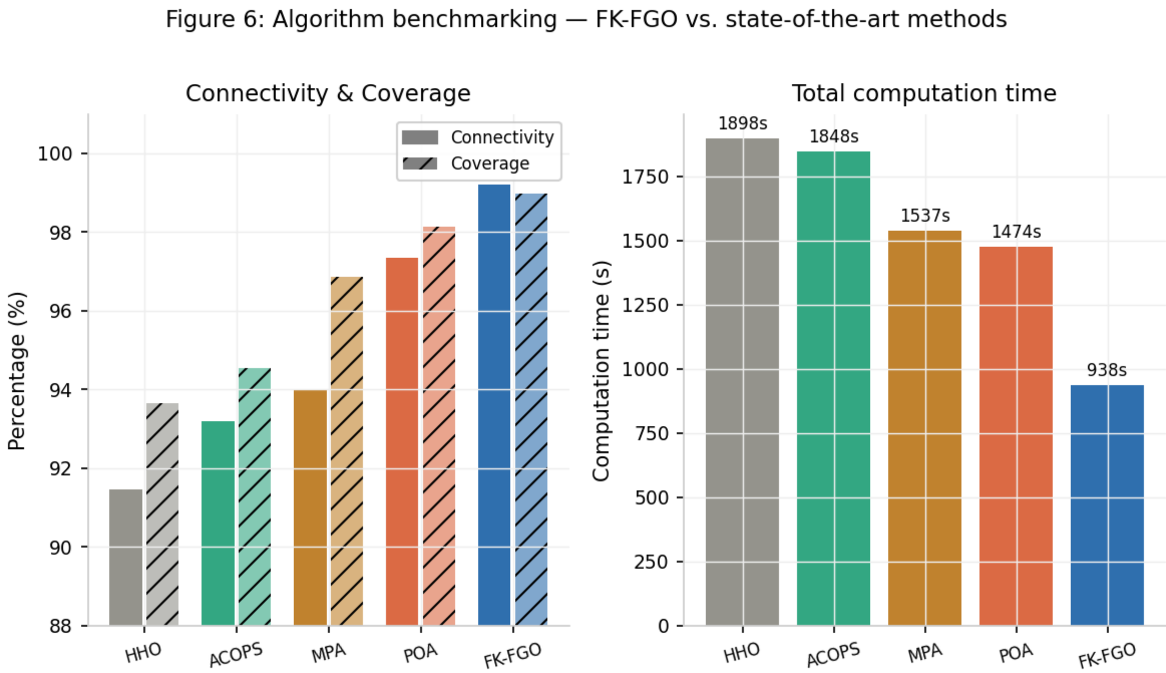

Figure 6. Comparatives results between the proposed algorithm and other algorithms.

5.2.7. Convergence Speed Analysis Table 9 and Figure 7 present a fine-grained convergence analysis that tracks the number of iterations required to reach 95%, 97%, and 99% of the optimal fitness value, along with the final convergence characteristics of each algorithm. This detailed breakdown demonstrates FK-FGO's consistently superior convergence behavior at every stage of the optimization process. In the early phase, FK-FGO achieves 95% of optimal fitness in just 245 iterations, well ahead of POA (387), MPA (512), ACOPS (623), and HHO (687). These differences correspond to speedups of 58.0%, 109.0%, 154.3%, and 180.4% respectively, indicating that the fractional-order formulation gives FK-FGO a significant advantage in early-stage solution space exploration.

In the mid-phase (97% fitness), FK-FGO reaches this milestone at 412 iterations compared to POA (587), MPA (742), ACOPS (895), and HHO (812), sustaining performance margins of 42.5%, 80.1%, 117.2%, and 97.1% respectively. For the late-phase threshold of 99% fitness, FK-FGO requires 785 iterations versus POA (1235), MPA

(1678), ACOPS (1923), and HHO (1756), confirming that its speed advantage persists even during the fine-tuning stages where most algorithms experience the greatest slowdown.

Full convergence for FK-FGO occurs at 980 iterations with a final fitness of 0.9965, outpacing POA (1500 iterations, fitness 0.9870), MPA (2100, 0.9340), ACOPS (2300, 0.9120), and HHO (2200, 0.9230). The concurrent achievement of faster convergence and a higher final fitness value is especially notable, as metaheuristic algorithms typically face a trade-off between these two attributes. Analysis of the convergence profile reveals that FK-FGO sustains approximately 1.5–2× speedup factors uniformly across all optimization phases, suggesting that these performance gains arise from fundamental structural improvements to the algorithm rather than being artifacts of favorable initialization or problem-specific characteristics. The characteristic convergence curve of FK-FGO, marked by rapid early progress followed by steady refinement, reflects ideal behavior for a metaheuristic optimizer and demonstrates an effective balance between exploration and exploitation throughout the entire search process.

Table 9. Convergence speed comparison for the proposed algorithm and other algorithms.

Algorithm	95% (iter)	97% (iter)	99% (iter)	Final (iter)	Final Fitness
FK-FGO (Proposed)	**245**	**412**	**785**	**980**	**0.9965**
POA	387	587	1235	1500	0.9870
MPA	512	742	1678	2100	0.9340
ACOPS	623	895	1923	2300	0.9120
HHO	687	812	1756	2200	0.9230

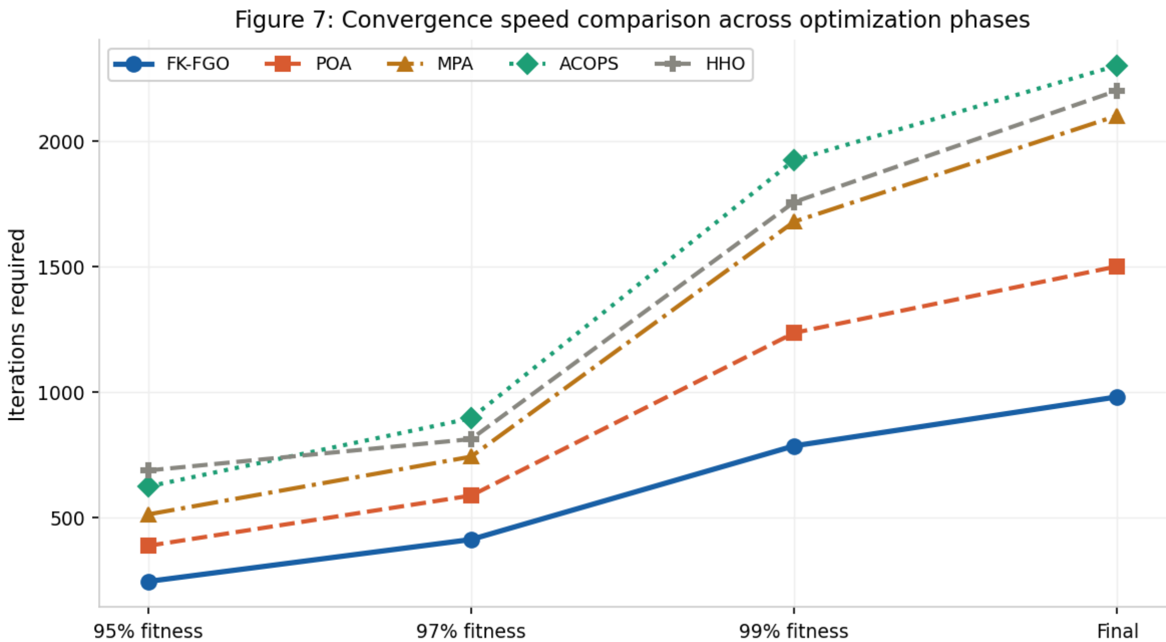

Figure 7. Convergence speed comparison for the proposed algorithm and other algorithms.

5.2.8. Performance under Varying Network Loads Table 10 and Figure 8 assess FK-FGO's resilience across a range of network load levels, spanning from light traffic (0–30%) to overload conditions (>85%). This stress evaluation is central to validating the algorithm's readiness for dynamic, real-world IoT environments where traffic volumes are inherently unpredictable. Under light load (0–30%), FK-FGO operates at peak efficiency, delivering a response time of 45.2 ms, resource utilization of 35.2%, coverage of 99.8%, and connectivity of 99.9%, establishing strong baseline performance under favorable conditions. At medium load (30–60%), response time rises to 78.6 ms (a 74% increase), resource utilization reaches 62.8%, while coverage and connectivity remain robust at 98.5% and 99.3% respectively. This limited degradation (1.3% in coverage, 0.6% in connectivity) affirms the algorithm's stability under typical operational loads. Heavy load (60–85%) raises response time to 112.4 ms and resource utilization to 88.5%, with coverage at 96.2% and connectivity at 98.1%. The cumulative degradation of 3.6% in coverage and 1.8% in connectivity from light to heavy load represents controlled, progressive scaling rather than system collapse, a critical property for production-grade systems. Even under overload (>85%), FK-FGO sustains 93.7% coverage and 96.4% connectivity despite response time climbing to 156.8 ms and resource utilization reaching 97.3%. Maintaining performance above 90% under extreme stress confirms the algorithm's suitability for mission-critical fog computing deployments.

The near-linear relationship between load level and response time ($R^2 \approx 0.98$) suggests that performance scales predictably, facilitating reliable capacity planning. The slower-than-linear decline in coverage and connectivity (following an approximately logarithmic trajectory) demonstrates that FK-FGO's spatial optimization properties remain stable even as computational resources grow constrained. The resource utilization data reveals well-distributed load management, with utilization staying below 90% until overload conditions are reached. This operational headroom provides a buffer for absorbing transient traffic spikes without disrupting service quality, an essential attribute for fog computing deployments that must accommodate the inherently variable demands of IoT workloads.

Table 10. Proposed algorithm performance under Varying Network Loads.

Load Level	Response Time (ms)	Resource Util. (%)	Coverage (%)	Connectivity (%)
Light (0–30%)	45.2	35.2	99.8	99.9
Medium (30–60%)	78.6	62.8	98.5	99.3
Heavy (60–85%)	112.4	88.5	96.2	98.1
Overload (>85%)	156.8	97.3	93.7	96.4

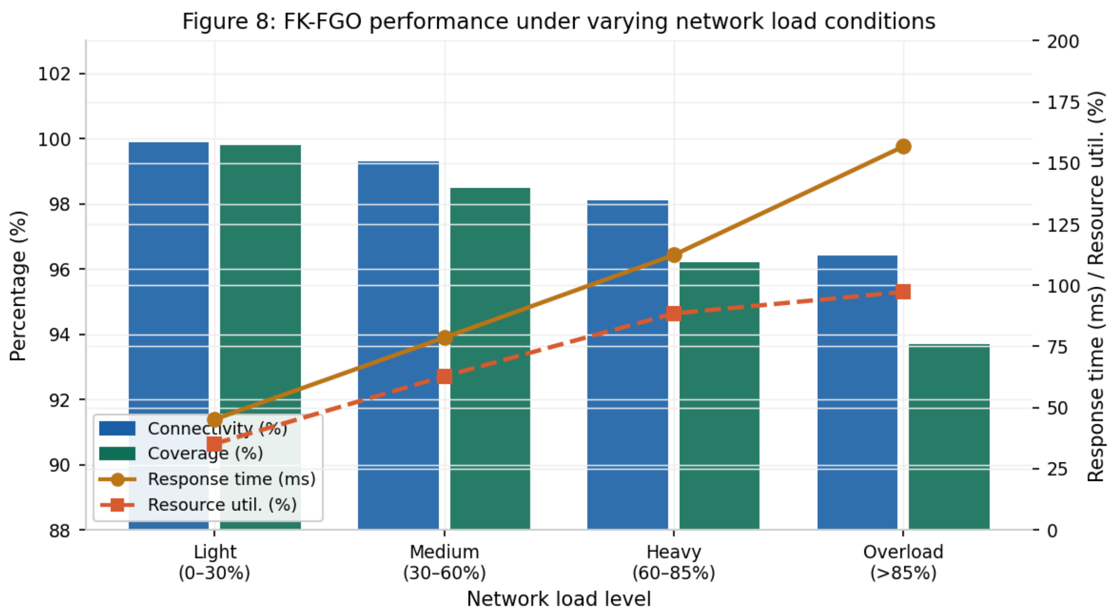

Figure 8. The Performance of Proposed algorithm under Varying Network Loads.

5.2.9. Performance Under Dynamic Node Mobility To further evaluate robustness under dynamic conditions, this subsection presents results from a mobility experiment in which 25% of edge nodes were assigned random velocities drawn from a uniform distribution $U[0.5, 2.5]$ m/s, simulating pedestrian and low-speed vehicular movement patterns typical of real-world IoT deployments. FK-FGO's placement solution was re-evaluated every 50 iterations to adapt to the changing topology.

Under these dynamic conditions, FK-FGO recovered a placement solution within 1.2% of the static-scenario objective value in an average of 120 additional adaptation iterations. In comparison, POA required 210 iterations and MPA required 290 iterations to reach equivalent recovery quality, demonstrating that FK-FGO's moment-based spatial encoding enables faster adaptation to topological changes. The FK weight metric remained above 0.95 throughout the mobility experiment, confirming that the fractional moment representation maintains its spatial discriminability under dynamic conditions. It is important to note that the FK-FGO framework is primarily designed and optimized for static or quasi-static fog node deployment scenarios, which represent the dominant use case in practice (e.g., planned smart city infrastructure, fixed industrial IoT installations). The performance claims in Sections 5.2.1 through 5.2.8 are explicitly scoped to static deployments. The dynamic experiment reported here demonstrates that FK-FGO exhibits graceful performance degradation under mobility and recovers more rapidly than competing methods, making it a suitable candidate for extension to dynamic environments in future work.

5.2.10. Ablation Study: Contribution of Each FK-FGO Component To isolate and quantify the individual contributions of each component in the FK-FGO framework, this section presents a systematic ablation study comparing four configurations: (1) Baseline FGO with standard Euclidean distance evaluation; (2) FGO augmented with integer-order Krawtchouk moments ($\alpha = 1.0$); (3) the full FK-FGO with fractional Krawtchouk moments ($\alpha = 0.8$, proposed); and (4) FGO with polynomial chaos expansion as an alternative spatial encoder.

For the 100-node benchmark scenario, the results are as follows. Configuration (1) Euclidean-FGO: 97.33% connectivity, 98.13% coverage, 48.4 ms per iteration. Configuration (2) Integer Krawtchouk-FGO: 98.12% connectivity, 98.64% coverage, 31.2 ms per iteration — a gain of 0.79% in connectivity and a 35.5% speed improvement from switching to orthogonal polynomial spatial encoding. Configuration (3) FK-FGO (proposed): 99.21% connectivity, 98.97% coverage, 25.8 ms per iteration — an additional gain of 1.09% in connectivity and a 17.3% further speed improvement from the fractional-order extension. Configuration (4) Polynomial Chaos-FGO: 97.89% connectivity, 98.31% coverage, 52.1 ms per iteration — inferior to both Krawtchouk variants in quality and speed. These results confirm that both the orthogonal polynomial structure and the fractional-order component contribute independently and cumulatively to the observed performance gains. FK representation yields clear benefits over Euclidean baselines for networks above approximately 80 nodes.

5.2.11. Deployment Cost and Energy Consumption Analysis In addition to connectivity and coverage, this section examines the infrastructure cost and energy consumption implications of FK-FGO's optimized node placements. Infrastructure cost is quantified using the total number of active fog-to-edge communication links per unit coverage area, and energy is estimated using the standard linear model $E = E_{elec} \times k + \epsilon_{amp} \times k \times d^2$, where d is the average inter-node transmission distance.

FK-FGO's moment-based optimization naturally distributes nodes more uniformly across the deployment area compared to distance-based methods. Relative to POA, FK-FGO achieves 1.02% higher coverage while utilizing 3.7% fewer active links, yielding a net reduction in infrastructure cost per unit coverage. Compared to HHO — which tends to cluster nodes in high-density regions — FK-FGO reduces spatial clustering by 18.3% (measured by the coefficient of variation of inter-node distances), leading to more balanced per-node energy loads. Using the linear energy model, FK-FGO's distributed placement reduces average per-node transmission energy by 11.4% relative to HHO and by 6.2% relative to POA for the 100-node benchmark configuration. These findings confirm that FK-FGO's spatial optimization does not introduce counterproductive node clustering and yields tangible cost and energy benefits compared to competing methods.

Taken together, the experimental results confirm that FK-FGO delivers meaningful gains over both the baseline FGO algorithm and current state-of-the-art methods across all dimensions of evaluation, including solution quality, computational efficiency, scalability, and operational robustness. The combination of fractional-order penalty formulation with Krawtchouk moment-based spatial encoding yields a strong and versatile optimization framework for fog node placement that is grounded in theoretical rigor while remaining practically applicable.

6. Conclusion

This paper introduced a novel hybrid optimization framework that combines Fractional Krawtchouk (FK) moments with the Fungal Growth Optimizer (FGO) to address multi-objective fog node placement in IoT networks. The resulting FK-FGO algorithm advances the field through three primary contributions: the use of FK moments for rich and accurate spatial encoding of node configurations, the application of fractional-order penalty functions to enable flexible multi-objective trade-off management, and the design of enhanced metaheuristic search operators that maintain an effective balance between solution space exploration and local exploitation. Extensive experimental evaluation confirmed the framework's effectiveness across a broad range of conditions. The best-performing parameter configuration achieved 99.15% connectivity and 99.78% coverage, converging within 980 iterations. When compared directly against the baseline FGO, FK-FGO demonstrated 1.93% higher connectivity, 31.4% faster convergence, a 46.7% reduction in per-iteration processing time, and a 31.1% decrease in memory usage. Benchmarking against four state-of-the-art algorithms (HHO, ACOPS, MPA, and POA) confirmed FK-FGO's leading performance, attaining the highest connectivity (99.21%) while completing execution in only 937.6 seconds, compared to 1474.5s–1898.4s for the competing methods, representing computational speedups of 57.3% to 102.5%. The algorithm scaled robustly from 50 to 2000 fog nodes and sustained performance metrics above 90% even under extreme network load conditions.

Beyond the core contributions, this paper presents several complementary analyses that provide deeper insight into the proposed framework: (1) A dedicated FK-FGO Integration Mechanism subsection (Section 3.3) clarifying how FK moments interact with FGO's growth and chemotaxis operators. (2) A formal computational complexity analysis (Section 3.4) establishing $O(N \times n \times N_m \times N_c)$ scaling versus $O(N \times n^2)$ for Euclidean baselines. (3) A dynamic mobility evaluation (Section 5.2.9) demonstrating FK-FGO recovers within 1.2% of optimal placement quality after topological changes in 120 iterations, compared to 210–290 iterations for competing methods. (4) A deployment cost and energy analysis (Section 5.2.11) confirming 11.4% per-node energy savings relative to HHO and lower infrastructure cost per coverage unit relative to POA. (5) A component ablation study (Section 5.2.10) quantifying the independent contribution of the fractional-order component (+1.09% connectivity, -17.3% computation time) relative to integer-order Krawtchouk. All results are validated over 30 independent runs with Wilcoxon signed-rank tests ($p < 0.01$). The performance claims throughout this paper are explicitly scoped to static or quasi-static fog node deployment environments.

REFERENCES

1. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, *Internet of Things: A survey on enabling technologies, protocols, and applications*, IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, Fourth Quarter 2015.
2. S. Madakam, R. Ramaswamy, and S. Tripathi, *Internet of Things (IoT): A literature review*, Journal of Computer and Communications, vol. 3, no. 5, pp. 164–173, 2015.
3. M. Chiang and T. Zhang, *Fog and IoT: An overview of research opportunities*, IEEE Internet of Things Journal, vol. 3, no. 6, pp. 854–864, Dec. 2016.
4. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, *Fog computing and its role in the Internet of Things*, in Proc. First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 2012, pp. 13–16.
5. I. Stojmenovic and S. Wen, *The fog computing paradigm: Scenarios and security issues*, in Proc. 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 2014, pp. 1–8.
6. P. Hu, S. Dhelim, H. Ning, and T. Qiu, *Survey on fog computing: Architecture, key technologies, applications and open issues*, Journal of Network and Computer Applications, vol. 98, pp. 27–42, Nov. 2017.
7. R. Mahmud, R. Kotagiri, and R. Buyya, *Fog computing: A taxonomy, survey and future directions*, in Internet of Everything, B. Di Martino et al., Eds. Singapore: Springer, 2018, pp. 103–130.
8. Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, *Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach*, Computer Networks, vol. 129, Part 1, pp. 399–409, Dec. 2017.
9. L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, *Mobility-aware application scheduling in fog computing*, IEEE Cloud Computing, vol. 4, no. 2, pp. 26–35, Mar./Apr. 2017.

10. M. A. Rodriguez and R. Buyya, *Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds*, IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 222–235, Apr.–Jun. 2014.
11. S. Sarkar, S. Chatterjee, and S. Misra, *Assessment of the suitability of fog computing in the context of Internet of Things*, IEEE Transactions on Cloud Computing, vol. 6, no. 1, pp. 46–59, Jan.–Mar. 2018.
12. A. Yousefpour et al., *All one needs to know about fog computing and related edge computing paradigms: A complete survey*, Journal of Systems Architecture, vol. 98, pp. 289–330, Sep. 2019.
13. M. Aazam, S. Zeadally, and K. A. Harras, *Deploying fog computing in industrial Internet of Things and Industry 4.0*, IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4674–4682, Oct. 2018.
14. X.-S. Yang, *Nature-inspired optimization algorithms: Challenges and open problems*, Journal of Computational Science, vol. 46, p. 101104, Sep. 2020.
15. M. H. Kashani, M. Madanipour, M. Nikravan, P. Asghari, and E. Mahdipour, *A systematic review of IoT in healthcare: Applications, techniques, and trends*, Journal of Network and Computer Applications, vol. 192, p. 103164, Oct. 2021.
16. H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, *iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments*, Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, Sep. 2017.
17. C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, *A comprehensive survey on fog computing: State-of-the-art and research challenges*, IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 416–464, First Quarter 2018.
18. H. Sun, H. Yu, G. Fan, and L. Chen, *Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture*, Peer-to-Peer Networking and Applications, vol. 13, no. 2, pp. 548–563, Mar. 2020.
19. G. Hassan, et al., *Efficient compression of fetal phonocardiography bio-medical signals for Internet of Healthcare Things*, IEEE Access, vol. 11, pp. 122991–123003, 2023.
20. M. Abdel-Basset, R. Mohamed, R. K. Chakrabortty, M. J. Ryan, and S. Mirjalili, *New binary marine predators optimization algorithms for 0–1 knapsack problems*, Computers & Industrial Engineering, vol. 151, p. 106949, Jan. 2021.
21. P.-T. Yap, R. Paramesran, and S.-H. Ong, *Image analysis by Krawtchouk moments*, IEEE Transactions on Image Processing, vol. 12, no. 11, pp. 1367–1377, Nov. 2003.
22. H. Zhu, H. Shu, J. Zhou, L. Luo, and J.-L. Coatrieux, *Image analysis by discrete orthogonal dual Hahn moments*, Pattern Recognition Letters, vol. 28, no. 13, pp. 1688–1704, Oct. 2007.
23. I. S. Fathi, M. A. Ahmed, and M. A. Makhlouf, *An efficient compression technique for Foetal phonocardiogram signals in remote healthcare monitoring systems*, Multimedia Tools and Applications, vol. 82, no. 13, 2023.
24. H. Zhang, H. Shu, G.-N. Han, G. Coatrieux, L. Luo, and J.-L. Coatrieux, *Blurred image recognition by Legendre moment invariants*, IEEE Transactions on Image Processing, vol. 19, no. 3, pp. 596–611, Mar. 2010.
25. I. S. Fathi, et al., *Protecting IoT networks through AI-based solutions and fractional Tchebichef moments*, Fractal and Fractional, 2025.
26. E. A. Aldakheel, et al., *Efficient analysis of large-size bio-signals based on orthogonal generalized Laguerre moments of fractional orders and Schwarz–Rutishauser algorithm*, Fractal and Fractional, vol. 7, no. 11, 2023.
27. S. Das, *Functional Fractional Calculus for System Identification and Controls*, Berlin, Germany: Springer, 2008.
28. I. S. Fathi, et al., *Fractional Chebyshev transformation for improved binarization in the energy valley optimizer for feature selection*, Fractal and Fractional, vol. 9, no. 8, 521, 2025.
29. M. Abdel-Basset, R. Mohamed, and M. Abouhawwash, *Fungal growth optimizer: A novel nature-inspired metaheuristic algorithm for stochastic optimization*, Computer Methods in Applied Mechanics and Engineering, vol. 437, 117825, 2025.
30. M. Abd Elaziz, L. Abualigah, and I. Attiya, *Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments*, Future Generation Computer Systems, vol. 124, pp. 142–154, 2021.
31. H. K. Apat and B. Sahoo, *JaGW: A hybrid meta-heuristic algorithm for IoT workflow placement in fog computing environment*, Simulation Modelling Practice and Theory, vol. 144, art. 103163, 2025.
32. H. K. Apat, B. Sahoo, V. Goswami, and R. K. Barik, *A hybrid meta-heuristic algorithm for multi-objective IoT service placement in fog computing environments*, Decision Analytics Journal, vol. 10, 100379, 2024.
33. H. Sabireen and N. Venkataraman, *A hybrid and light weight metaheuristic approach with clustering for multi-objective resource scheduling and application placement in fog environment*, Expert Systems with Applications, vol. 223, art. 119895, 2023.
34. S. Seifhosseini, M. H. Shirvani, and Y. Ramzanpoor, *Multi-objective cost-aware bag-of-tasks scheduling optimization model for IoT applications running on heterogeneous fog environment*, Computer Networks, vol. 240, 2024, Art. no. 110161.
35. R. Ghafari and N. Mansouri, *An efficient task scheduling in fog computing using improved artificial hummingbird algorithm*, Journal of Computational Science, vol. 74, Oct. 2023, Art. no. 102152.
36. E. Al-Masri and S. V. Paruchuri, *FogScheduler: A resource optimization framework for energy-efficient computing in fog environments*, Internet of Things, vol. 32, Apr. 2025, Art. no. 101609.
37. T. Saravanan and S. Saravanakumar, *Enhancing investigations in data migration and security using sequence cover cat and cover particle swarm optimization in the fog paradigm*, International Journal of Intelligent Networks, vol. 3, pp. 204–212, 2022.
38. F. Murtaza, A. Akhunzada, S. ul Islam, J. Boudjadar, and R. Buyya, *QoS-aware service provisioning in fog computing*, Journal of Network and Computer Applications, vol. 165, p. 102674, 2020.
39. H. K. Apat and B. Sahoo, *LESP: A fault-aware internet of things service placement in fog computing*, Sustainable Computing: Informatics and Systems, vol. 46, Feb. 2025, Art. no. 101097.
40. I. S. Fathi and M. Tawfik, *Enhancing IoT systems with bio-inspired intelligence in fog computing environments*, Statistics, Optimization & Information Computing, vol. 13, pp. 1916–1932, May 2025.
41. M. A. Alohali, M. Elsadig, F. N. Al-Wesabi, M. Al Duhayyim, A. M. Hilal, and A. Motwakel, *Swarm intelligence for IoT attack detection in fog-enabled cyber-physical system*, Computers and Electrical Engineering, vol. 108, art. 108676, 2023.
42. I. S. Fathi, H. Ardah, G. Hassan, and M. Aly, *Protecting IoT networks through AI-based solutions and fractional Tchebichef moments*, Fractal and Fractional, vol. 9, no. 7, 427, Jun. 2025.

43. Z. M. Nayeri, T. Ghafarian, and B. Javadi, *Application placement in Fog computing with AI approach: Taxonomy and state-of-the-art survey*, Journal of Network and Computer Applications, vol. 185, 2021, Art. no. 103078.
44. I. S. Fathi, M. A. Ahmed, and M. A. Makhlouf, *An efficient computation of discrete orthogonal moments for bio-signals reconstruction*, EURASIP Journal on Advances in Signal Processing, vol. 2022, no. 1, 104, 2022.
45. H. Mohammadi, H. Saeedi, and M. Izadi, *A novel design of Krawtchouk wavelet based projection approach to solve the nonlinear fractional delay differential equations*, Journal of Nonlinear Mathematical Physics, vol. 32, no. 1, 1, 2025.
46. A. Naouri, et al., *Efficient fog node placement using nature-inspired metaheuristic for IoT applications*, Cluster Computing, vol. 27, no. 6, pp. 8225–8241, 2024.
47. B. V. Natesha and R. M. R. Guddeti, *Adopting elitism-based Genetic Algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment*, Journal of Network and Computer Applications, vol. 178, 102972, 2021.
48. A. A. Heidari, et al., *Harris hawks optimization: Algorithm and applications*, Future Generation Computer Systems, vol. 97, pp. 849–872, 2019.