OPEN ACCESS

Swarm-Intelligent Adaptive Sampling for Distributed Environmental Monitoring Under Uncertain Spatiotemporal Dynamics

Rizky Aditya Putra[1], Dewi Lestari Pramudita[2]

[1] Department of Computer Science and Engineering, Universitas Samudra, 89 Jalan Meurandeh–Langsa Timur, Langsa 24416, Indonesia;
[2] Department of Computer Science and Engineering, Universitas Kadiri, 41 Jalan Selomangleng–Mojoroto Raya, Kediri 64115, Indonesia;

Abstract

Traditional on premise data warehouses have long served as the backbone of analytical processing for enterprises that rely on structured and semi structured data for reporting, business intelligence, and regulatory compliance. Over the past decade, the emergence of distributed cloud data warehouse platforms has introduced alternative architectures that promise elastic scalability, consumption based pricing, and geographically distributed processing capabilities. As organizations modernize their analytics landscape, many have executed large scale migrations from legacy on premise platforms to cloud native or hybrid data warehouse environments, yet the concrete performance, scalability, and cost outcomes of these transitions remain heterogeneous and often context dependent. This paper presents a technical analysis of the observable outcomes associated with such migrations, with emphasis on the engineering mechanisms that influence query performance, concurrency handling, elasticity, and total cost of ownership over time. The discussion synthesizes practical migration patterns, workload characteristics, and architectural design choices that shape post migration behavior, including storage and compute separation, multi cluster execution, and automated lifecycle management of data. Rather than advocating for a particular platform, the paper focuses on neutral characterization of how resource provisioning models, network topology, and operational practices interact to determine realized benefits or regressions. The analysis is intended to support engineering teams in forming realistic expectations, selecting appropriate benchmarks, and identifying instrumentation strategies for ongoing validation. By examining performance, scalability, and cost as interdependent dimensions, the paper highlights trade offs that commonly emerge after migration and outlines considerations for iterative optimization in distributed cloud data warehouse environments.

1. Introduction

Environmental systems exhibit complex spatial and temporal variability driven by physical transport processes, external forcings, and heterogeneous sources and sinks [1]. Examples include dispersion of airborne pollutants in urban canyons, nutrient transport in coastal waters, soil moisture dynamics in agricultural settings, and temperature fluctuations in indoor or outdoor built environments. Accurate monitoring of such systems requires sensing strategies that are capable of resolving multiscale patterns and reacting to transient events [2]. Fixed sensing networks, while reliable and simple to maintain, frequently yield coarse spatial coverage and may miss local anomalies. The emergence of mobile sensing platforms, including autonomous surface and underwater vehicles, aerial robots, and ground robots, creates opportunities to reposition sensing assets adaptively and to concentrate measurements where they are most informative for prediction or decision making.

When sensing resources are realized as a swarm of relatively simple agents, the monitoring problem becomes one of coordinating many mobile sensors with limited local capabilities [3]. Swarm-intelligent approaches leverage distributed interactions and collective behavior rather than centralizing all decision making. Each agent relies on locally available measurements, motion constraints, and communication with nearby peers, while global sensing objectives emerge from the aggregated behavior of the swarm [4]. The design of such systems must therefore account for constraints on bandwidth, energy, onboard computation, and the reliability of communication links,

Copyright by the Authors.
Licensed as an open access article using a CC BY 4.0 license.

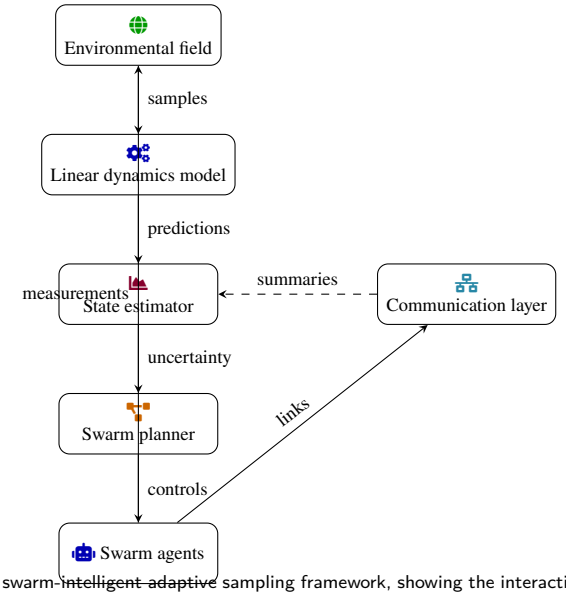


Figure 1: High-level architecture of the swarm-~~intelligent adaptive~~ sampling framework, showing the interaction between the environmental field, linear dynamics model, estimator, planner, swarm agents, and the communication substrate that distributes compressed information across the network.

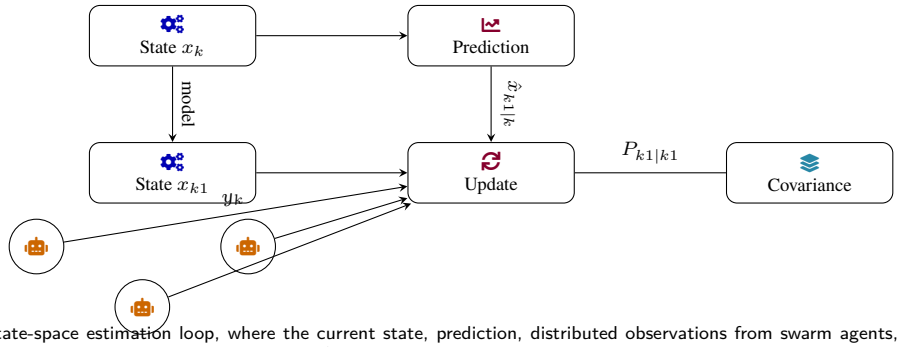


Figure 2: Linear state-space estimation loop, where the current state, prediction, distributed observations from swarm agents, and the update step combine to produce the next state estimate and covariance for uncertainty-aware planning.

in addition to the uncertainties inherent in environmental dynamics.

Adaptive sampling refers to strategies that adjust sensing locations and frequencies on the basis of current predictions and uncertainty estimates. In environmental monitoring, adaptive sampling seeks to allocate more measurements to regions and time intervals where the state is poorly known or highly variable, and fewer measurements where the system is relatively predictable [5]. Achieving this in a swarm setting requires an explicit modeling framework that links sampling decisions to their expected informational value, together with control and optimization mechanisms that map these values into motion commands. Furthermore, the modeling approach needs to balance fidelity and tractability, since high-dimensional environmental fields can be expensive to forecast, and optimization problems over swarm trajectories can grow quickly with the number of agents and planning horizon.

Uncertain spatiotemporal dynamics pose additional challenges [6]. Governing equations may be only partially known or available at coarse resolution, and external disturbances, boundary conditions, or unresolved nonlinearities can induce deviations from nominal predictions. The observation process is also uncertain due to sensor noise, calibration errors, and intermittently lost data [7]. In this context, adaptive sampling rules that depend strongly on precise probabilistic models of the environment may be difficult to implement or brittle. Linear modeling and linear optimization provide a compromise, enabling systematic analysis and efficient computation while retaining the ability to capture key features such as advection, diffusion, and measurement error.

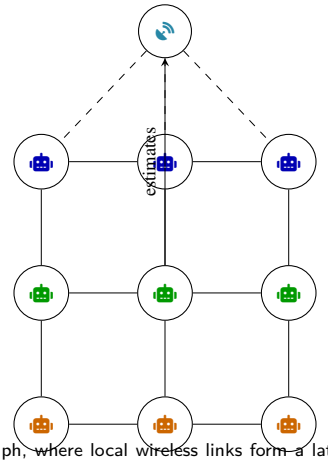


Figure 3: Snapshot of the swarm communication graph, where local wireless links form a lattice-like topology and a relay node aggregates selected state estimates from interior agents for coordination across wider spatial scales.

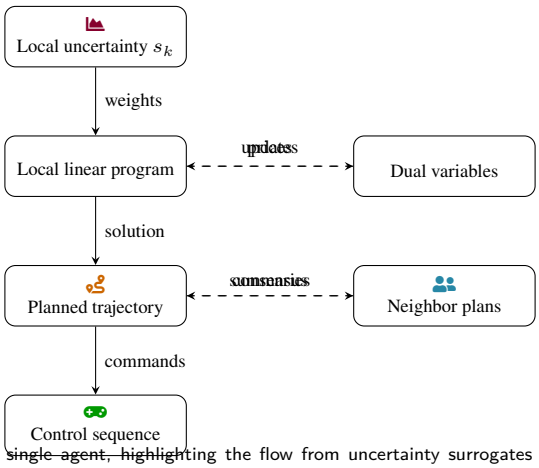


Figure 4: Local optimization loop for a single agent, highlighting the flow from uncertainty surrogates to linear program, planned trajectory, and control sequence, together with dual variable exchange and neighbor plan sharing that implement distributed coordination.

This work investigates a swarm-intelligent adaptive sampling framework built on linear state-space models and linear approximations of information-driven objectives [8]. The environmental field is represented as a discretized state vector evolving according to linear dynamics with additive uncertainty. Each mobile agent acquires linear measurements of the state at its location, corrupted by noise. A linear estimation mechanism combines these measurements to infer the state, and estimated covariance information is used to construct surrogate objectives for sampling, such as weighted uncertainty metrics and coverage penalties [9]. Swarm control inputs are then derived by solving linear or mixed linear optimization problems that encode motion, collision avoidance, and communication constraints in a tractable way.

A distributed coordination structure is employed to improve scalability and robustness [10]. Rather than requiring a centralized controller with full knowledge of the swarm and the environment, each agent maintains a local model of the dynamics and communicates with a limited set of neighbors. Local optimization problems approximate the global objective using consensus-like penalties and neighborhood information. Learning mechanisms allow agents to refine model parameters online using observed discrepancies between predicted and measured states [11]. Numerical experiments examine how these design choices influence estimation accuracy, spatial coverage, and sensitivity to model mismatch, as well as the role of communication topology and limited bandwidth. The study focuses on analytical and empirical understanding of the behavior of the proposed framework, and on the characterization of parameter regimes in which it yields consistent monitoring performance under uncertain spatiotemporal dynamics.

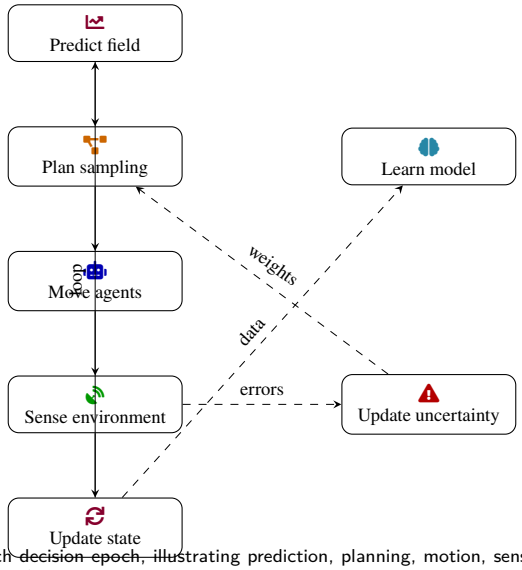


Figure 5: Adaptive sampling cycle for each decision epoch, illustrating prediction, planning, motion, sensing, and state update, together with online learning of model parameters and refinement of uncertainty weights used in subsequent sampling decisions.

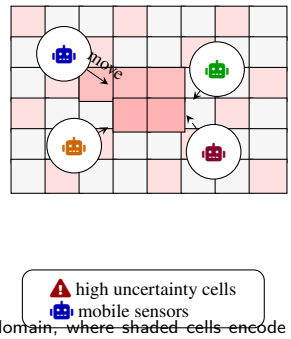


Figure 6: Example coverage pattern on a discretized domain, where shaded cells encode spatial variability in predicted uncertainty and the mobile agents are steered toward regions of elevated uncertainty while maintaining dispersion across the grid.

2. Modeling of Environmental Fields and Sensing Dynamics

Environmental fields can be modeled as continuous functions of space and time that evolve according to partial differential equations [12]. For algorithmic design and analysis, it is common to approximate these dynamics by a finite-dimensional state-space representation. Consider a field defined over a region that is discretized into n cells, and let $x_k \in \mathbb{R}^n$ denote the vector of field values at discrete time index k. A linear time-varying approximation of the dynamics can be written as [13]

$$x_{k1} = A_k x_k \;\; B_k w_k, \tag{2.1}$$

where $A_k \in \mathbb{R}^{n\times n}$ encodes propagation and diffusion, $B_k \in \mathbb{R}^{n\times q}$ encodes the effect of process disturbances, and $w_k \in \mathbb{R}^q$ is an exogenous uncertainty term representing unmodeled inputs and stochastic forcing. The matrices may vary with time due to changing boundary conditions or external drivers, but are assumed to remain known or slowly varying over the horizon of interest.

A swarm of m mobile sensing agents is deployed within the region. The position of agent i at time k is represented by a vector $p_{i,k} \in \mathbb{R}^d$, with d typically equal to two or three. For convenience, the union of agent positions at time k is denoted by a stacked vector $p_k \in \mathbb{R}^{md}$. The motion model of each agent is treated as kinematically linear over the planning time step, with control input $u_{i,k} \in \mathbb{R}^d$ describing the commanded displacement or velocity. A simple discrete-time model is [14]

$$p_{i,k1} = p_{i,k} \;\; u_{i,k}, \tag{2.2}$$

subject to bounds $\|u_{i,k}\|_2 \leq u_{\max}$ and workspace constraints expressed as convex sets in the position space. These constraints can be translated into linear or convex inequalities by approximating the workspace with poly-

topes and using norms that admit linear representations when necessary.

The measurement acquired by agent i at time k is modeled as a noisy linear function of the field state. Let $y_{i,k} \in \mathbb{R}$ denote the scalar measurement obtained at the location $p_{i,k}$. Discretization of the field implies that each cell is associated with a sampling kernel, so that the measurement can be expressed as [15]

$$y_{i,k} = h_{i,k}^{\top} x_k \;\; v_{i,k}, \tag{2.3}$$

where $h_{i,k} \in \mathbb{R}^n$ is a row vector that depends on the location of the agent and reflects interpolation or averaging over nearby cells, and $v_{i,k}$ is a measurement noise term with zero mean and variance that may depend on sensor characteristics and environmental conditions. Stacking all measurements at time k yields a vector $y_k \in \mathbb{R}^m$ representing simultaneous observations from the swarm.

The mapping from agent positions p_k to measurement matrices H_k is defined by the relationship

$$y_k = H_k x_k \;\; v_k, 16 \tag{2.4}$$

where $H_k \in \mathbb{R}^{m \times n}$ contains rows $h_{i,k}^{\top}$ determined by the positions $p_{i,k}$ and the discretization scheme, and $v_k \in \mathbb{R}^m$ stacks the noise terms. This formulation explicitly couples the sampling pattern of the swarm to the information content of the observations. The degree of coupling depends on the spatial correlation structure of the field and the resolution of the discretization.

Uncertain spatiotemporal dynamics are represented through probabilistic assumptions on w_k and v_k, together with possible parametric uncertainty in A_k and B_k [17]. A common simplifying assumption is that the disturbances form zero-mean Gaussian sequences with covariances Q_k and R_k, respectively, and are mutually independent across time. Under these assumptions, linear Gaussian state-space estimation techniques can be used to propagate means and covariances of the state estimate. Specifically, the predicted state estimate $\hat{x}_{k1|k}$ and covariance $P_{k1|k}$ satisfy

$$\hat{x}_{k1|k} = A_k \hat{x}_{k|k}, \tag{2.5}$$

$$P_{k1|k} = A_k P_{k|k} A_k^{\top} \;\; B_k Q_k B_k^{\top}. \tag{2.6}$$

Upon receiving measurements y_k, the updated estimate $\hat{x}_{k|k}$ and covariance $P_{k|k}$ are computed using linear correction steps that depend on H_k and R_k. The structure of these equations reveals how spatiotemporal sampling decisions encoded in H_k shape the evolution of the estimation error covariance [18].

Parametric uncertainty in A_k and B_k can be modeled via perturbations, such as $A_k = \bar{A}_k \;\; \Delta A_k$ with ΔA_k belonging to a bounded set. Linear robustness analysis techniques then provide conditions under which estimation or control properties are maintained despite such uncertainties. In the context of swarm-intelligent adaptive sampling, these uncertainties motivate the use of sampling policies that do not depend on precise knowledge of dynamics, but instead rely on estimated statistics and conservative bounds [19]. Linearization around nominal flows or empirical covariance approximations derived from data can be incorporated into the same framework by updating $\bar{A}_k$ and Q_k according to observed discrepancies between predictions and measurements.

3. Swarm-Intelligent Adaptive Sampling Architecture

The adaptive sampling architecture considered here is structured around a swarm of agents that share information through a communication graph. Let the undirected graph at time k be represented by $\mathcal{G}_k = \mathcal{V}, \mathcal{E}_k$, where the vertex set $\mathcal{V}$ contains the swarm agents and $\mathcal{E}_k$ denotes communication links available at that time. A link between agents i and j exists if their distance is below a communication radius or if there is an available relay. This graph induces a neighborhood set $\mathcal{N}_{i,k}$ for each agent containing indices of agents from which it can receive information. Communication is assumed to be local and of limited bandwidth, constraining the type and amount of data exchanged [20].

Each agent maintains a local estimate of the environmental state, denoted $\hat{x}_{i,k}$, and an associated error covariance $P_{i,k}$. These estimates are updated via local prediction steps based on the dynamics model and local correction steps that use its own measurements and selected information from neighbors. A simple fusion rule is to construct convex combinations of neighboring estimates. For example, an intermediate fused estimate for agent i at time k can be written as [21]

$$\tilde{x}_{i,k} = \sum_{j \in \mathcal{N}_{i,k}} w_{ij,k} \hat{x}_{j,k}, \tag{3.1}$$

where the weights $w_{ij,k}$ are nonnegative and satisfy $\sum_{j \in \mathcal{N}_{i,k}} w_{ij,k} = 1$. Similar convex rules can be applied to fuse covariance information subject to consistency constraints. This consensus-like mechanism allows information to propagate across the swarm without requiring full connectivity to a central node [22].

Parameter	Symbol	Dimension	Role
State vector	x_k	$n \times 1$	Discretized field
Dynamics matrix	A_k	$n \times n$	Linear propagation
Process noise	w_k	$q \times 1$	Exogenous forcing
Measurement matrix	H_k	$m \times n$	Sampling pattern
Covariance	$P_{k\|k}$	$n \times n$	Uncertainty state

Table 1: Core linear state-space quantities used in the environmental field model.

Metric	Symbol	Range	Interpretation
Mean squared error	MSE	≥ 0	Spatial average error
Max absolute error	$\mathrm{E}_{\max}$	≥ 0	Worst-case error
Coverage ratio	ρ_{cov}	0–100%	Visited cells fraction
Median revisit time	T_{med}	> 0	Temporal sampling gap
Path length	L_{tot}	> 0	Total motion budget

Table 2: Evaluation metrics for estimation accuracy, coverage quality, and motion effort.

Adaptive sampling decisions are generated by computing control actions $u_{i,k}$ as functions of local estimates, neighbor information, and operational constraints. The control law can be expressed in the form

$$u_{i,k} = f_i\left(\tilde{x}_{i,k}, p_{i,k}, \{p_{j,k}\}_{j\in\mathcal{N}_{i,k}}\right), \quad (3.2)$$

where f_i is derived from a local optimization problem. This problem typically includes terms that represent the expected reduction in uncertainty from potential measurements, penalties for deviation from coverage patterns, and costs associated with motion or communication [23]. The output of the local optimization is a motion command that repositions the agent to locations expected to be informative in upcoming time steps.

A key aspect of swarm-intelligent behavior is that agents are not assigned rigid roles or predefined trajectories. Instead, each agent evaluates its own local conditions and adapts its behavior accordingly [24]. For example, when an agent detects high uncertainty in a nearby region of the environmental field, it may prioritize visiting that region while negotiating with neighbors to avoid redundant sampling. When multiple agents identify similar opportunities, local coordination rules based on simple negotiation or inhibition signals can ensure that the sampling load is distributed [25]. Because the architecture relies on local rules and communication, it remains applicable when the swarm size changes or communication links are intermittently unavailable.

Safety and operational constraints are integrated into the architecture through linear inequalities that bound positions and velocities, represent collision avoidance margins, and enforce no-fly or no-sail zones. For instance, a pairwise separation requirement between agents i and j can be conservatively approximated by linear constraints on the components of their position difference [26]. Similarly, limited battery capacity or endurance is modeled by linear constraints on cumulative motion norms or activation times. These constraints are embedded in the local optimization problems that generate control actions, so that adaptive sampling decisions respect hardware and regulatory limits.

Online learning mechanisms are incorporated to adapt model parameters and sampling strategies to observed discrepancies between predicted and measured environmental states [27]. For example, each agent can maintain local estimates of process and measurement noise variances, updating them using simple recursive rules that depend on innovation sequences. A linear regression model can be used to adjust entries of the dynamics matrix A_k by minimizing prediction errors over a sliding window of recent data [28]. These updates are performed with limited computational overhead and can be restricted to parameters affecting the subset of state components that are frequently observed by each agent. The resulting parameter estimates are then shared with neighbors to improve global consistency.

Constraint	Form	Type	Effect
Velocity bound	$\|u_{i,k}\|_2 \le u_{\max}$	Local	Limits maneuvering
Workspace	$E_i p_{i,k} \le e_i$	Global	Enforces domain
Separation	$G_{ij} p_k \le g_{ij}$	Pairwise	Collision margin
Energy budget	$F_i u_{i,0:K} \le f_i$	Cumulative	Motion cost cap
Link radius	$\|p_{i,k} - p_{j,k}\|_2 \le r_c$	Graph	Maintains links

Table 3: Representative linear constraints influencing swarm motion and safety.

Quantity	Symbol	Scale	Comment
Number of agents	m	10–80	Swarm size
State dimension	n	10^2–10^3	Grid resolution
Horizon length	K	5–20	Planning depth
Communication degree	d_{avg}	3–6	Mean neighbors
Solver iterations	N_{it}	5–15	Local optimization

Table 4: Typical parameter scales in simulations of distributed linear optimization for adaptive sampling.

4. Linear Optimization of Sampling Policies

The goal of the sampling policy is to determine motion commands that shape future measurement matrices H_k such that the estimation error or similar performance metric is controlled, while satisfying operational constraints [29]. Direct optimization of information-theoretic criteria such as mutual information is generally nonlinear and computationally demanding. To facilitate implementation in swarms, linear surrogate objectives are introduced that approximate the desired behavior. One such surrogate is based on a linear function of predicted variance weights across state components [30].

Consider a vector $s_k \in \mathbb{R}^n$ representing predicted state variances or a related uncertainty measure. A linear objective can be defined as

$$J_k = c_k^\top s_k, \tag{4.1}$$

where $c_k \in \mathbb{R}^n$ encodes relative importance weights across spatial locations. The evolution of s_k depends on the selection of measurements via the matrix H_k, which in turn depends on agent positions p_k [31]. For linear Gaussian models, the covariance update equations are nonlinear in H_k, but linear upper bounds on s_k can be derived based on coverage indicators or sampling frequencies. These indicators are linear functions of binary or continuous decision variables that represent whether a given state component is sampled by any agent at time k.

To capture both current and future performance, a finite-horizon objective can be written as [32]

$$J = \sum_{k=0}^{K-1} \gamma_k^\top s_k, \tag{4.2}$$

with weight vectors γ_k specifying the relative importance of uncertainty at different times. The adaptive sampling problem then seeks decision variables that minimize this objective subject to linear dynamics and constraints. Decision variables include approximate coverage indicators, linearized motion commands, and, when appropriate, binary variables representing assignment of agents to grid cells or waypoints [33]. When binary variables are used, the problem becomes a mixed-integer linear program; in many swarm settings, continuous relaxations or randomized rounding strategies are employed to reduce computational effort at the expense of an exact combinatorial interpretation.

The motion of each agent can be constrained by linear inequalities of the form [34]

$$G_i u_{i,k} \le g_i, \tag{4.3}$$

where G_i and g_i represent bounds on velocities, accelerations, or allowable directions imposed by the environment. Workspace constraints are expressed through inequalities on the positions,

$$E_i p_{i,k} \le e_i, \tag{4.4}$$

where matrices E_i and vectors e_i define polytopic feasible regions [35]. Inter-agent collision avoidance constraints can be approximated by requiring that positions remain

Component	Weight	Sign	Effect on behavior
Uncertainty term	α_u	> 0	Prioritize high variance
Coverage term	α_c	> 0	Spread agents
Motion cost	α_m	> 0	Penalize long paths
Consensus term	α_s	> 0	Align local plans
Regularizer	α_r	≥ 0	Smooth trajectories

Table 5: Main linear objective components appearing in the local optimization problems.

Strategy	MSE (rel.)	Coverage	Path length
Static grid	1.00	35%	Low
Random walk	0.88	55%	Medium
Greedy variance	0.71	62%	Medium–high
Swarm linear policy	0.64	68%	Medium
Idealized oracle	0.55	72%	High

Table 6: Representative relative performance of different sampling strategies in terms of error, coverage, and motion effort.

within disjoint convex regions or by enforcing constraints on relative positions linearized around nominal trajectories. These approximations enable the entire constraint set to be treated within a linear optimization framework [36].

Global optimization problems that consider all agents and future steps can be large. To support distributed implementation, the global objective is decomposed into local objectives assigned to each agent, coupled by consensus penalties or resource constraints. For agent i, a local objective may take the form [37]

$$J_i = \sum_{k=0}^{K-1} \alpha_{i,k}^\top z_{i,k}, \tag{4.5}$$

where $z_{i,k}$ aggregates local uncertainty indicators, motion costs, and coordination signals, and $\alpha_{i,k}$ are designer-specified weights. Coordination with neighbors is captured by linear terms penalizing deviations between local decisions and neighborhood averages. For instance, if $\xi_{i,k}$ is a vector representing a local coverage pattern, then a neighborhood agreement penalty can be written as

$$C_{i,k} = \sum_{j \in \mathcal{N}_{i,k}} \eta_{ij,k}^\top \left(\xi_{i,k} - \xi_{j,k} \right), \tag{4.6}$$

where $\eta_{ij,k}$ encodes coupling strengths. This term is linear in the decision variables and encourages coordinated sampling while maintaining a structure amenable to distributed optimization algorithms.

The resulting local optimization problem for each agent is of the form [38]

$$\min_{u_{i,0:K-1}, \xi_{i,0:K-1}} J_i \tag{4.7}$$

subject to linear dynamics, motion constraints, and linearized coverage constraints. Here, $u_{i,0:K-1}$ denotes a sequence of control decisions, and $\xi_{i,0:K-1}$ denotes auxiliary variables describing the local sampling pattern. This problem can be solved using linear or mixed-integer solvers onboard the agent or at nearby compute nodes, with updated solutions computed periodically as new information becomes available [39]. When computational resources are limited, receding horizon strategies with short horizons and simplified constraint sets can be employed to maintain feasibility.

5. Distributed Coordination and Online Learning

Distributed coordination in a swarm context requires algorithms that operate with local information and limited communication, yet guide the collective towards sampling patterns that support accurate environmental estimation. Consensus-based methods provide one mechanism to achieve this goal [40]. In addition to the fusion of state estimates described earlier, agents can coordinate their sampling decisions through iterative averaging of local plans. Consider a variable $\chi_{i,k}$ that encodes a compact representation of agent i future trajectory or coverage plan. A consensus update at iteration ℓ can be written as

$$\chi_{i,k}^{\ell 1} = \sum_{j \in \mathcal{N}_{i,k}} a_{ij,k} \chi_{j,k}^{\ell}, \tag{5.1}$$

where the weights $a_{ij,k}$ satisfy the usual stochasticity conditions. By alternating between local optimization steps

Parameter	Variation	Effect on error	Effect on coverage
Swarm size m	↑	Decreases	Increases
Horizon K	↑	Moderate decrease	Slight increase
Noise level	↑	Increases	Mild change
Link radius	↑	Decreases	Little change
Model mismatch	↑	Increases	Scenario-dependent

Table 7: Qualitative sensitivity of estimation error and spatial coverage to key design and environment parameters.

that improve each agent plan with respect to local objectives and consensus steps that blend plans with neighbors, agents can reach approximate agreement on coordinated sampling strategies without global communication [41].

Online learning is used to adapt the models and objectives guiding sampling. One component of learning is parameter estimation for the dynamics matrix [42]. Suppose a subset of state components of interest can be modeled by a linear regression relationship

$$x_{k1}^r = F_k x_k^s \ \epsilon_k, \tag{5.2}$$

where x_k^s is a vector of selected predictors, x_{k1}^r is a vector of responses, F_k is a parameter matrix, and ϵ_k is a residual term. Each agent can construct local estimates of F_k by accumulating paired samples of x_k^s and x_{k1}^r derived from its own state estimates and measurements. A recursive least-squares update for F_k takes the form

$$F_{k1} = F_k \ L_k\left(z_{k1} - F_k y_k\right), \tag{5.3}$$

where L_k is a gain matrix computed from local covariance estimates, y_k is a predictor vector, and z_{k1} is a response vector. This update preserves linear structure and can be executed with modest computational requirements [43]. Inter-agent exchange of parameter estimates through averaging can further improve robustness and convergence.

A second component of learning concerns adaptation of the uncertainty surrogate s_k and its weights. The surrogate is initially derived from model-based covariance predictions, but can be updated using empirical discrepancies between predictions and measurements [44]. For instance, innovations $e_{i,k} = y_{i,k} - h_{i,k}^\top \hat{x}_{i,k}$ provide a linear measure of prediction error. Agents can maintain local statistics of innovation magnitudes and use them to revise entries of s_k associated with regions where errors are persistently high. This adjustment modifies the objective function by effectively assigning higher priority to regions where the model underestimates uncertainty [45]. Linear update rules of the form

$$s_{k1} = I - \Lambda_k s_k \ \Lambda_k r_k \tag{5.4}$$

can be used, where Λ_k is a diagonal gain matrix and r_k is a vector of innovation-based metrics.

Distributed optimization algorithms enable agents to solve local linear programs that approximate a global sampling objective [46]. One class of algorithms is based on primal decomposition, in which global coupling constraints are dualized and prices are communicated between agents. Another class is based on alternating direction methods, where local primal variables and dual multipliers are updated iteratively. In both cases, the structure of the linear objectives and constraints allows agents to compute updates using only local data and messages from neighbors [47]. The convergence properties of such algorithms depend on graph connectivity and step-size choices, and in the presence of time-varying graphs and delays, convergence is typically to a neighborhood of a global optimum rather than exact optimality.

The integration of online learning and distributed optimization yields a feedback loop that adjusts both model parameters and sampling policies based on observed system behavior [48]. When environmental conditions shift or new structures emerge, the regression and covariance estimation mechanisms modify the dynamics approximation and uncertainty surrogate. These modifications in turn reshape the linear optimization problems solved by agents, altering their trajectories and sampling locations. Because this feedback loop is realized through simple linear updates and convex optimization, it remains amenable to analysis and implementation even when the swarm size is large [49].

6. Simulation Studies and Case Analyses

To examine the behavior of the swarm-intelligent adaptive sampling framework, simulation studies can be performed

on idealized environmental fields that capture key features of practical systems. One class of test fields is based on advection-diffusion dynamics on a two-dimensional grid, where a scalar quantity is transported by a prescribed velocity field and smoothed by diffusion. The discretized dynamics can be represented in the linear state-space form discussed earlier, with the matrix A_k constructed from finite-difference approximations of the governing equations [50]. Process noise is added to represent unresolved variability and numerical errors. The state dimension can range from a few hundred to several thousand, depending on the grid resolution [51].

A swarm of mobile agents is initialized with random positions within the domain. The initial positions are subject to workspace constraints that prevent agents from entering restricted regions. Each agent carries a point sensor that returns localized measurements of the field with additive Gaussian noise [52]. Communication links are established based on a distance threshold, yielding a time-varying graph as agents move. At each time step, agents execute the local prediction and update steps of the estimation algorithm, solve their local linear optimization problems to generate motion commands, and communicate compressed summaries of their state estimates and planned trajectories to neighbors.

Performance is evaluated using metrics derived from the estimation error and sampling patterns [53]. A common metric is the spatially averaged mean squared error between the estimated and true fields, defined over the grid cells. A related metric is the maximum pointwise error across space, which emphasizes worst-case deviations [54]. Coverage metrics quantify the fraction of grid cells visited by at least one agent over a given time window, and the distribution of revisit times. Operational metrics include the total distance traveled by the swarm, the distribution of inter-agent distances, and the fraction of time that communication links are active. These metrics provide a multi-faceted view of the trade-offs between information gain, operational effort, and safety [55].

By comparing the adaptive sampling strategy to baseline strategies such as static deployments or random walks, one can assess how uncertainty-aware decision making influences performance. Static strategies fix agent positions or follow predetermined patrol patterns, independent of observed data. Randomized strategies allow agents to move according to stochastic rules without explicit objectives [56]. In typical scenarios, the swarm-intelligent adaptive framework yields estimation errors that are lower than those of static or random baselines under the same resource constraints, though the magnitude of the reduction varies with field dynamics, noise levels, and model mismatch. The comparisons also reveal cases where the benefits of adaptivity are limited, such as when the environmental field is nearly homogeneous or slowly varying relative to the sampling rate [57].

Sensitivity analyses explore the impact of parameters such as swarm size, communication radius, planning horizon length, and weight choices in the objective function. Increasing swarm size generally improves estimation accuracy and coverage, but with diminishing returns beyond a certain point. When the communication radius is small, the graph may fragment, and the benefits of distributed information fusion can degrade, leading to localized estimation performance that depends strongly on initial conditions [58]. Short planning horizons reduce computational requirements but may restrict the ability of agents to exploit forecasts of future uncertainty, especially in systems where the dynamics exhibit strong temporal correlations. Weight choices in the objective function influence the balance between exploration of high-uncertainty regions, maintenance of baseline coverage, and limiting motion effort.

Model mismatch and unmodeled dynamics play a significant role in shaping achieved performance [59]. In simulations where the true dynamics deviate from the linear model due to nonlinear advection or boundary effects, initial predictions can be biased, leading to misallocation of sensing effort. The inclusion of online learning mechanisms mitigates some of this bias by updating linear model parameters and uncertainty surrogates based on observed discrepancies [60]. However, the rate and robustness of adaptation depend on the magnitude of the mismatch and on the richness of the data. When changes in dynamics occur abruptly, such as a sudden shift in dominant wind direction, rapid adaptation requires that agents encounter informative regions quickly; otherwise, there can be a lag before sampling patterns adjust.

The simulation studies also highlight practical considerations for real deployments [61]. For example, energy constraints can limit the total distance traveled by agents, requiring careful tuning of motion penalties in the objective function to prevent excessive maneuvering. Constraints on ascent or descent rates for aerial or underwater vehicles can restrict accessible regions within the planning horizon, affecting coverage of vertical structures in the field. Communication dropouts and delays introduce asynchrony in the distributed coordination algorithms, which may necessitate conservative parameter choices to maintain stability [62]. Despite these challenges, the results suggest that the combination of linear modeling,

uncertainty-aware optimization, and distributed coordination provides a flexible foundation on which more detailed vehicle and communication models can be layered.

7. Conclusion

This work has examined a swarm-intelligent adaptive sampling framework for distributed environmental monitoring under uncertain spatiotemporal dynamics, emphasizing linear modeling and optimization formulations that facilitate analysis and implementation [63]. Environmental fields were represented using finite-dimensional linear state-space models with additive uncertainties, and mobile sensors were modeled as agents whose motion and measurements influence the information available for state estimation. Within this structure, linearized uncertainty surrogates were introduced to approximate information-driven sampling objectives in a way that is compatible with linear and mixed-integer linear programming.

The swarm architecture leveraged local estimation, distributed information fusion, and neighborhood-based coordination to adjust sampling patterns in response to evolving uncertainty and operational constraints [64]. Each agent maintained a local state estimate, communicated with neighbors through a time-varying graph, and solved local optimization problems that balance expected uncertainty reduction, coverage requirements, motion effort, and safety considerations. Linear constraints were used to encode motion bounds, workspace restrictions, and collision avoidance in a tractable form. Consensus-like mechanisms and distributed optimization methods enabled agents to achieve approximate agreement on sampling patterns without centralized control [65].

Online learning components were incorporated to adapt dynamics models and uncertainty surrogates, using innovation sequences and regression-based parameter updates. These components allowed the swarm to respond to model mismatch and changes in environmental dynamics, although the speed and robustness of adaptation depend on data availability and the nature of the deviations from nominal models [66]. Simulation studies on stylized advection-diffusion fields illustrated how the framework can shape sensor trajectories, improve estimation accuracy relative to non-adaptive baselines in many cases, and maintain performance under a range of communication and resource constraints.

Several limitations and potential extensions arise from the analysis. The reliance on linear modeling and linear surrogates restricts the range of dynamics and sensing modalities that can be accurately captured, and nonlinearity or strong anisotropy in environmental processes may require more advanced representations [67]. Computation and communication limits may constrain the complexity or horizon length of local optimization problems, especially in large swarms. Future developments could investigate hybrid schemes that combine linear models with selective nonlinear corrections, hierarchical control structures that coordinate groups of agents at different levels of abstraction, and integration of additional sensing cues or prior information. Despite these limitations, the study provides a structured perspective on how swarm-intelligent adaptive sampling strategies can be formulated, analyzed, and tuned to support distributed environmental monitoring in the presence of uncertain spatiotemporal dynamics [68].

References

[1] L. Cheng, Q. Zhang, F. Tao, K. Ni, and Y. Cheng, "A novel search algorithm based on waterweeds reproduction principle for job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 84, pp. 405–424, 11 2015.

[2] S. E. Wako, "Behaviour and ecological impacts of termites: Fecundity investigations in mounds," *Ekologia*, vol. 34, pp. 72–81, 1 2015.

[3] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Swarm robotic odor localization: Off-line optimization and validation with real robots," *Robotica*, vol. 21, pp. 427–441, 9 2003.

[4] R. Chandrasekar and S. Misra, "Introducing an aco based paradigm for detecting wildfires using wireless sensor networks," in *2006 International Symposium on Ad Hoc and Ubiquitous Computing*, pp. 112–117, IEEE, 2006.

[5] R. V. Devi and S. S. Sathya, "Monkey behavior based algorithms - a survey," *International Journal of Intelligent Systems and Applications*, vol. 9, pp. 67–86, 12 2017.

[6] A. K. Gupta, H. Sadawarti, and A. K. Verma, "Computation of pheromone values in antnet algorithm," *International Journal of Computer Network and Information Security*, vol. 4, pp. 47–54, 8 2012.

[7] H. Gao and J. Cao, "Non-dominated sorting quantum particle swarm optimization and its application in cognitive radio spectrum allocation," *Journal of*

Central South University, vol. 20, pp. 1878–1888, 7 2013.

[8] R. Juliana and P. U. Maheswari, "An energy efficient cluster head selection technique using network trust and swarm intelligence," *Wireless Personal Communications*, vol. 89, pp. 351–364, 3 2016.

[9] X. liang Ma and M. min Lv, "Image crowd fusion mechanism based on optical embedded system," *EURASIP Journal on Embedded Systems*, vol. 2016, pp. 19–, 10 2016.

[10] L. Ma and B. Forouraghi, "A hyperspherical particle swarm optimizer for robust engineering design," *The International Journal of Advanced Manufacturing Technology*, vol. 67, pp. 1091–1102, 10 2012.

[11] G. Kalyani, M. V. P. C. S. Rao, and B. Janakiramaiah, "Particle swarm intelligence and impact factor-based privacy preserving association rule mining for balancing data utility and knowledge privacy," *Arabian Journal for Science and Engineering*, vol. 43, pp. 4161–4178, 9 2017.

[12] P. H. Prathibha and C. P. Chandran, "Classification mining snps from leukaemia cancer dataset using linear classifier with aco," *Bonfring International Journal of Data Mining*, vol. 6, pp. 10–15, 4 2016.

[13] M. Shaukat and M. Chitre, "Adaptive behaviors in multi-agent source localization using passive sensing," *Adaptive behavior*, vol. 24, pp. 446–463, 9 2016.

[14] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, "An ant odor analysis approach to the ant colony optimization algorithm for data-aggregation in wireless sensor networks," in *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, IEEE, 2006.

[15] A. Khosla, S. Kumar, and K. K. Aggarwal, "Identification of strategy parameters for particle swarm optimizer through taguchi method," *Journal of Zhejiang University-SCIENCE A*, vol. 7, pp. 1989–1994, 12 2006.

[16] L. Yamamoto, D. Miorandi, P. Collet, and W. Banzhaf, "Recovery properties of distributed cluster head election using reaction–diffusion," *Swarm Intelligence*, vol. 5, pp. 225–255, 9 2011.

[17] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic Computing*, vol. 6, pp. 31–47, 1 2014.

[18] M. M. al Rifaie, A. J. Cropley, D. H. Cropley, and M. Bishop, "On evil and computational creativity," *Connection Science*, vol. 28, pp. 171–193, 3 2016.

[19] R. A. Vázquez and B. A. Garro, "Training spiking neural models using artificial bee colony," *Computational intelligence and neuroscience*, vol. 2015, pp. 947098–947098, 2 2015.

[20] Álvaro Gutiérrez, A. Campo, F. Monasterio-Huelin, L. Magdalena, and M. Dorigo, "Collective decision-making based on social odometry," *Neural Computing and Applications*, vol. 19, pp. 807–823, 5 2010.

[21] J. Srinivas, R. Giri, and S.-H. Yang, "Optimization of multi-pass turning using particle swarm intelligence," *The International Journal of Advanced Manufacturing Technology*, vol. 40, pp. 56–66, 12 2007.

[22] S. Girirajkumar, A. Kumar, and N. Anantharaman, "Tuning of a pid controller for a real time industrial process using particle swarm optimization," *International Journal of Computer Applications*, vol. ecot, pp. 35–40, 12 2010.

[23] T. Schmickl, R. Thenius, and K. Crailsheim, "Swarm-intelligent foraging in honeybees: benefits and costs of task-partitioning and environmental fluctuations," *Neural Computing and Applications*, vol. 21, pp. 251–268, 4 2010.

[24] R. Chandrasekar, V. Vijaykumar, and T. Srinivasan, "Probabilistic ant based clustering for distributed databases," in *2006 3rd International IEEE Conference Intelligent Systems*, pp. 538–545, IEEE, 2006.

[25] R. S. de Moraes and E. P. de Freitas, "Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services," *Journal of Intelligent & Robotic Systems*, vol. 92, pp. 645–656, 11 2017.

[26] W. sung Moon, J.-W. Jang, H.-S. Kim, and K.-R. Baek, "Virtual pheromone map building and a utilization method for a multi-purpose swarm robot system," *International Journal of Control, Automation and Systems*, vol. 13, pp. 1446–1453, 9 2015.

[27] S. R. M. Sekhar, A. S, B. Maruthi, B. Kumar, and C. Janiwarad, "A survey on bio-inspired security in wireless sensor networks," *International Journal of Research in Engineering and Technology*, vol. 04, pp. 449–454, 5 2015.

[28] J. S. Turner, "Termites as models of swarm cognition," *Swarm Intelligence*, vol. 5, pp. 19–43, 12 2010.

[29] J. Li, T. Ma, M. Tang, W. Shen, and Y. Jin, "Improved fifo scheduling algorithm based on fuzzy clustering in cloud computing," *Information*, vol. 8, pp. 25–, 2 2017.

[30] R. Chiong and T. Weise, "Special issue on modern search heuristics and applications," *Evolutionary Intelligence*, vol. 4, pp. 1–2, 1 2011.

[31] B. Pandey and N. Garg, "Swarm optimized m odular neural network based diagnostic system for breast cancer diagnosis," *International Journal on Soft Computing, Artificial Intelligence and Applications*, vol. 2, pp. 11–20, 8 2013.

[32] Q. Lu and P. Luo, "A learning particle swarm optimization algorithm for odor source localization," *International Journal of Automation and Computing*, vol. 8, pp. 371–380, 8 2011.

[33] S.-S. Kim, S. McLoone, J.-H. Byeon, S. Lee, and H. Liu, "Cognitively-inspired artificial bee colony clustering for cognitive wireless sensor networks," *Cognitive Computation*, vol. 9, pp. 207–224, 1 2017.

[34] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, "An obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs," in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1–6, 2006.

[35] D. Saha and N. Das, "Self-organized area coverage in wireless sensor networks by limited node mobility," *Innovations in Systems and Software Engineering*, vol. 12, pp. 227–238, 4 2016.

[36] J. C. Bansal, H. Sharma, K. V. Arya, and A. K. Nagar, "Memetic search in artificial bee colony algorithm," *Soft Computing*, vol. 17, pp. 1911–1928, 3 2013.

[37] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Computing*, vol. 21, pp. 949–964, 8 2015.

[38] Y. Li, Q. Wen, and B. Zhang, "Chaotic ant swarm optimization with passive congregation," *Nonlinear Dynamics*, vol. 68, pp. 129–136, 9 2011.

[39] M. Arsuaga-Ríos and M. A. Vega-Rodríguez, "Swarm approach based on gravity for optimizing energy savings in grid systems," *Journal of Heuristics*, vol. 20, pp. 617–641, 7 2014.

[40] B. Forouraghi, "Optimal tolerance allocation using a multiobjective particle swarm optimizer," *The International Journal of Advanced Manufacturing Technology*, vol. 44, pp. 710–724, 1 2009.

[41] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groβ, "Self-organized aggregation without computation," *The International Journal of Robotics Research*, vol. 33, pp. 1145–1161, 4 2014.

[42] H. A. A. Al-Rawi, M. A. Ng, and K.-L. A. Yau, "Application of reinforcement learning to routing in distributed wireless networks: a review," *Artificial Intelligence Review*, vol. 43, pp. 381–416, 1 2013.

[43] Y. Jin, Y. Ding, K. Hao, and Y. Jin, "An endocrine-based intelligent distributed cooperative algorithm for target tracking in wireless sensor networks," *Soft Computing*, vol. 19, pp. 1427–1441, 7 2014.

[44] T. Srinivasan, R. Chandrasekar, V. Vijaykumar, V. Mahadevan, A. Meyyappan, and M. Nivedita, "Exploring the synergism of a multiple auction-based task allocation scheme for power-aware intrusion detection in wireless ad-hoc networks," in *2006 10th IEEE Singapore International Conference on Communication Systems*, pp. 1–5, IEEE, 2006.

[45] M. J. A. Hasan and S. Ramakrishnan, "A survey: hybrid evolutionary algorithms for cluster analysis," *Artificial Intelligence Review*, vol. 36, pp. 179–204, 3 2011.

[46] S. Sodsee, M. Komkhao, and P. Meesad, "Consensus decision making on scale-free buyer networks," *Journal of Computer Science*, vol. 7, pp. 1554–1559, 10 2011.

[47] R. Zhu and Y. Zhai, "Research on the application of vrp theory in logistics transportation," *MATEC Web of Conferences*, vol. 100, pp. 05064–, 3 2017.

[48] C. W. Zecha, J. Link, and W. Claupein, "Mobile sensor platforms: categorisation and research applications in precision farming," *Journal of Sensors and Sensor Systems*, vol. 2, pp. 51–72, 5 2013.

[49] A. Hanani, A. M. Rahmani, and A. Sahafi, "A multi-parameter scheduling method of dynamic workloads for big data calculation in cloud computing," *The Journal of Supercomputing*, vol. 73, pp. 4796–4822, 4 2017.

[50] G. Rigatos, "Distributed gradient and particle swarm optimization for multi-robot motion planning," *Robotica*, vol. 26, pp. 357–370, 5 2008.

[51] J.-W. Lee, "Optimal power allocating for correlated data fusion in decentralized wsns using algorithms based on swarm intelligence," *Wireless Networks*, vol. 23, pp. 1655–1667, 2 2017.

[52] P. Jiang, Y. Feng, F. Wu, S. Yu, and X. Huan, "Dynamic layered dual-cluster heads routing algorithm based on krill herd optimization in uwsns," *Sensors (Basel, Switzerland)*, vol. 16, pp. 1379–, 8 2016.

[53] C. T. Holbrook, R. M. Clark, D. Moore, R. P. Overson, C. A. Penick, and A. Smith, "Social insects inspire human design.," *Biology letters*, vol. 6, pp. 431–433, 4 2010.

[54] C. Ramachandran, R. Malik, X. Jin, J. Gao, K. Nahrstedt, and J. Han, "Videomule: a consensus learning approach to multi-label classification from noisy user-generated videos," in *Proceedings of the 17th ACM international conference on Multimedia*, pp. 721–724, 2009.

[55] Z. Wu, W. Fu, R. Xue, and W. Wang, "A novel global path planning method for mobile robots based on teaching-learning-based optimization," *Information*, vol. 7, pp. 39–, 7 2016.

[56] S.-H. Oh and J. Suk, "Evolutionary controller design for area search using multiple uavs with minimum altitude maneuver," *Journal of Mechanical Science and Technology*, vol. 27, pp. 541–548, 3 2013.

[57] S. Hossain and A. El-Shafie, "Intelligent systems in optimizing reservoir operation policy: A review," *Water Resources Management*, vol. 27, pp. 3387–3407, 4 2013.

[58] K. Deepa, C. Vivek, and S. P. Rajan, "Automatic threshold selections by exploration and exploitation of optimization algorithm in record deduplication," *JOURNAL OF ADVANCES IN CHEMISTRY*, vol. 12, pp. 4515–4522, 6 2016.

[59] G. Sartoretti, M.-O. Hongler, M. E. de Oliveira, and F. Mondada, "Decentralized self-selection of swarm trajectories: from dynamical systems theory to robotic implementation," *Swarm Intelligence*, vol. 8, pp. 329–351, 11 2014.

[60] D. Singh and A. Sirmorya, "Solving real optimization problem using genetic algorithm with employed bee (gaeb)," *International Journal of Computer Applications*, vol. 42, pp. 1–5, 3 2012.

[61] A. K. Joseph and G. Radhamani, "Fuzzy c means (fcm) clustering based hybrid swarm intelligence algorithm for test case optimization," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 8, pp. 76–82, 7 2014.

[62] A. Randazzo, "Swarm optimization methods in microwave imaging," *International Journal of Microwave Science and Technology*, vol. 2012, pp. 1–12, 10 2012.

[63] H. Guo and Y. Meng, "Distributed reinforcement learning for coordinate multi-robot foraging," *Journal of Intelligent & Robotic Systems*, vol. 60, pp. 531–551, 5 2010.

[64] R. Janapati, C. Balaswamy, and K. Soundararajan, "Localization of cooperative wsn using distributed pso with optimum references," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, pp. 3094–3094, 12 2016.

[65] R. Chandrasekar and S. Misra, "Using zonal agent distribution effectively for routing in mobile ad hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 3, no. 2, pp. 82–89, 2008.

[66] A. Brutschy, L. Garattoni, M. Brambilla, G. Francesca, G. Pini, M. Dorigo, and M. Birattari, "The tam: abstracting complex tasks in swarm robotics research," *Swarm Intelligence*, vol. 9, pp. 1–22, 1 2015.

[67] L. Peng, F. Zhang, and B. Zhou, "Dynamic background modeling using tensor representation and ant colony optimization," *Science China Mathematics*, vol. 60, pp. 2287–2302, 9 2017.

[68] I. M. Hezam, O. A. Raouf, and M. Hadhoud, "A new compound swarm intelligence algorithms for

solving global optimization problems," *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, vol. 10, pp. 2010–2020, 9 2013.