Eclipta Journals

RESEARCH

Swarm-Intelligence–Guided Clustering for Distributed Data Repositories with Communication-Aware Constraints

Rendra Saputra[1] and Yudha Firmansyah[2]

[1]Universitas Bosowa, Jalan Urip Sumoharjo No. 242, Departemen Sistem Informasi, Makassar, Sulawesi Selatan, Indonesia
[2]Universitas Kadiri, Jalan Selomangleng No. 1, Departemen Teknik Komputer, Kediri, Jawa Timur, Indonesia

Abstract

Distributed data repositories arise in settings where data are naturally partitioned across administrative domains, geographic regions, or edge devices, and where moving raw records is undesirable due to cost, latency, or policy. Clustering is frequently used in such environments to build shared structure for indexing, summarization, anomaly triage, and downstream learning, yet standard clustering formulations typically assume a centralized view of the dataset and ignore the mechanisms by which cluster state must be coordinated. This paper studies clustering over distributed repositories under communication-aware constraints, focusing on regimes where bandwidth budgets, heterogeneous links, and asynchronous participation materially affect achievable clustering quality. The core contribution is a swarm-intelligence–guided approach that treats cluster parameters as a population of candidate solutions that evolve through locally evaluated fitness signals, while explicitly accounting for the communication required to maintain and refine those solutions. Communication-aware constraints are incorporated through a constrained objective with explicit per-link and per-round budgets, delay-sensitive penalties, and message-size–dependent accounting of overhead. The resulting method uses distributed swarm updates, repository-level sub-swarms, and intermittent consensus to trade off distortion reduction against coordination cost. Analytical considerations are provided for stability and constraint satisfaction in stochastic and asynchronous settings, along with complexity characterizations that separate computation from network overhead. An evaluation framework is described to quantify clustering quality, constraint adherence, and communication efficiency across synthetic and repository-derived workloads.

1. Introduction

Modern data-intensive systems are increasingly organized around collections of repositories rather than monolithic stores [1]. Data may be produced and retained at the edge by sensors, mobile devices, and embedded controllers, or held in separate administrative silos such as hospitals, financial institutions, or business units. Even within a single organization, geographic replication and data governance can lead to multiple repositories with limited cross-site transfer [2]. In such settings, analytic tasks that rely on global structure must operate with partial views and constrained coordination. Clustering is a representative example because it is both a standalone objective and a building block for other workflows, including approximate nearest neighbor indexing, prototype-based classification, stratified sampling, and exploratory monitoring.

A typical centralized clustering pipeline repeatedly alternates between assigning points to clusters and updating cluster representatives such as centroids, medoids, or mixture parameters [3]. When data are distributed, those updates implicitly require aggregation of statistics across repositories. If the repositories are connected via a network whose links have limited bandwidth, variable latency, intermittent availability, or nontrivial monetary and energy costs, then the mechanics of aggregation can become the dominant factor determining whether a method is usable. Moreover, the network may impose heterogeneous constraints: some links may be high-throughput but high-latency, others low-power but reliable, and others intermittent [4]. Communication may also be governed by policy, where only specific message types, sketching mechanisms, or privacy-preserving summaries are allowed. These realities motivate formulations in which the quality of a clustering solution is evaluated jointly with the cost of achieving and maintaining it under explicit constraints.

A second challenge arises from the optimization landscape of clustering objectives [5]. Even in the centralized case, many commonly used clustering formulations are nonconvex, nonsmooth, or both. Distributed constraints add further structure: objectives may include penalty terms for network utilization, and feasible sets may include mixed discrete and continuous constraints reflecting message budgets and quantization. In such contexts, heuristic and metaheuris-

Table 1. Notation used in the proposed swarm-intelligence–guided clustering method

Symbol	Description	Typical range
N	Number of data objects across all repositories	10^3–10^7
K	Number of clusters	2–50
S	Number of distributed sites	2–32
P	Swarm population size	10–200
$T_{\max}$	Maximum iterations	50–500
λ	Communication-awareness weight	0–1

Table 2. Characteristics of the distributed benchmark datasets

Dataset	#Objects (N)	#Attributes	#Sites (S)
SensorNet	50 000	24	4
ClickStream	120 000	18	6
MedRecords	32 000	32	8
IoT-Grid	80 000	14	5
SocialTrace	150 000	20	10

tic approaches can be useful as they explore multiple candidate solutions and tolerate noisy or partially observed fitness evaluations [6]. Swarm intelligence methods, which maintain a population of candidate solutions updated by stochastic rules influenced by local best and collective best states, are a natural fit for distributed environments because evaluation can be decomposed and population members can be moved and mixed across nodes in communication-efficient ways.

The focus of this paper is a swarm-intelligence–guided clustering approach designed for distributed repositories with communication-aware constraints [7]. The guiding design principle is to treat communication not as an implementation detail but as an explicit part of the optimization problem. Candidate cluster parameter sets are evolved by a distributed swarm whose members are evaluated using local distortion and constraint-aware penalties computed at repositories. Coordination is achieved through intermittent and compressed exchanges of swarm state and of aggregated statistics [8]. The method aims to provide a tunable continuum between purely local clustering, which requires little communication but may miss global structure, and tightly synchronized global clustering, which can be costly or infeasible under stringent budgets.

Several considerations shape the formulation. Repositories may have heterogeneous feature representations due to differing preprocessing pipelines, sensor modalities, or missingness patterns, so a shared latent representation or alignment mechanism is often needed [9]. The network may be modeled as a directed graph with edge-specific capacities and costs, and with time-varying availability. Constraints may apply per round, per link, and per repository, and may couple across links due to shared bottlenecks. Communication overhead depends on message size, which is influenced by the number of clusters, the dimension of the representation, and the quantization or sketching strategy [10]. The clustering objective itself may vary, ranging from squared Euclidean distortion to Bregman divergences and robust alternatives. The paper therefore emphasizes a modular structure in which objective terms and constraint penalties can be adapted without changing the overall algorithmic template.

The remainder of the paper proceeds by specifying the system model and clustering objective in a way that makes the role of communication explicit, then introducing constraint models that capture bandwidth, latency, and message-size limitations in a common mathematical framework [11]. A swarm-intelligence–guided clustering method is presented that uses repository-level sub-swarms and constraint-aware fitness shaping,

Table 3. Main control parameters of the swarm-based clustering procedure

Parameter	Default	Role in the algorithm
Population size P	40	Number of candidate partitions per iteration
Cognitive factor c_1	1.8	Strength of individual memory component
Social factor c_2	1.6	Strength of neighborhood influence
Inertia weight w	0.7	Balance between exploration and exploitation
Comm. weight λ	0.3	Penalty for high communication cost
Max iterations $T_{\max}$	200	Stopping condition on global loop

Table 4. Considered variants of swarm-intelligence mechanisms

Variant	Swarm model	Encoding of solution	Local update rule
SI-PSO	Particle swarm	Cluster centroids	Velocity-based position update
SI-ACO	Ant colony	Cluster labels	Pheromone-guided path construction
SI-BA	Bat algorithm	Cluster centroids	Frequency and loudness adaptation
SI-GWO	Grey wolf	Cluster centroids	Hierarchical leader following
SI-HYB	Hybrid PSO–ACO	Mixed centroids/labels	Alternating PSO/ACO updates

along with coordination protocols that respect budgets by construction. Analytical considerations are discussed for stability, approximate stationarity, and constraint satisfaction under stochastic and asynchronous updates [12]. Finally, an evaluation methodology is outlined to compare communication-aware clustering behaviors across different network regimes and repository heterogeneity profiles.

2. System Model and Problem Formulation

Consider a collection of repositories indexed by $r \in \{1, \dots, R\}$, each holding a local dataset $\mathcal{D}_r = \{x_{r,i}\}_{i=1}^{n_r}$ with $x_{r,i}$ in an input space $\mathcal{X}_r$. The input spaces may differ across repositories due to sensor modality, feature engineering, or data schema. To support a shared clustering objective, each repository applies a local mapping $\phi_r : \mathcal{X}_r \to \mathbb{R}^d$ to embed records into a common d-dimensional representation. The mapping may be fixed, learned offline, or updated slowly relative to clustering [13]. The embedded vectors are denoted $y_{r,i} = \phi_r(x_{r,i})$. The goal is to partition the union of embedded points into K clusters represented by parameters $\Theta = \{\theta_k\}_{k=1}^{K}$, where θ_k may be a centroid in $\mathbb{R}^d$, a medoid index, or a parameter vector for a distributional model. For concreteness, the discussion emphasizes centroid-based clustering, but the communication-aware structure extends to other forms.

Let $z_{r,i} \in \{1, \dots, K\}$ denote the cluster assignment of $y_{r,i}$. A standard distortion objective for centroid-based clustering is

$$J(\Theta, Z) = \sum_{r=1}^{R} \sum_{i=1}^{n_r} \ell\big(y_{r,i}, \theta_{z_{r,i}}\big)\,, \tag{1}$$

where $\ell(\cdot, \cdot)$ is a loss such as squared Euclidean distance $|0y - \theta|0_2^2$ or a robust alternative [14]. In distributed settings, the assignments Z are computed locally given Θ, while the updates to Θ require information aggregated from all repositories. If synchronization were free, one could alternate between local assignment and global centroid updates. The central challenge considered here is that updating Θ requires messages exchanged over a network with constraints [15].

The communication fabric is modeled as a directed graph $G = (V, E)$ where $V = \{1, \dots, R\}$ and $(r, s) \in E$ indicates that repository r can send messages to repository s. Each edge has a capacity $b_{rs} \geq 0$ measured in bits per unit time, a latency parameter $\tau_{rs} \geq 0$, and a cost weight $\kappa_{rs} \geq 0$ that can represent monetary cost, energy consumption, or prioritization. Communication may be time-varying; this can be represented by allowing $b_{rs}(t)$ and $\tau_{rs}(t)$ to depend on a discrete time index t, but much of the structure can be explained in the time-invariant case and extended by replacing constants with random or adversarial sequences.

Clustering in this environment is not only about finding Θ and Z that minimize distortion, but also about selecting an update protocol that respects budgets. The protocol defines which messages are transmitted, when they are transmitted, and at what resolution. If $\mathcal{M}_{rs}(t)$ denotes the message transmitted from

Table 5. Communication models and constraints across distributed sites

Model	Bandwidth cap (Mbps)	Avg. latency (ms)	Synchronization scheme
CM-1	10	80	Fully synchronous
CM-2	50	40	Synchronous with local buffering
CM-3	5	120	Asynchronous, best-effort delivery
CM-4	100	20	Hierarchical master–worker
CM-5	20	60	Gossip-based partial synchronization

Table 6. Clustering quality comparison under identical communication constraints

Method	NMI ↑	ARI ↑	Silhouette ↑
k-means (centralized)	0.71	0.54	0.41
Distributed k-means	0.67	0.50	0.38
Local-only clustering	0.59	0.42	0.30
Proposed SI-PSO	0.79	0.63	0.47
Proposed SI-HYB	**0.82**	**0.66**	**0.50**

r to s at time t, then its size in bits is $|\mathcal{M}_{rs}(t)|$. The aggregate communication in round t can be expressed as [16]

$$C(t) = \sum_{(r,s)\in E} \kappa_{rs}\,|\mathcal{M}_{rs}(t)|. \tag{2}$$

A communication-aware clustering problem can then be seen as jointly choosing (Θ, Z) and a protocol Π that determines $\{\mathcal{M}_{rs}(t)\}$ so as to reduce distortion while controlling communication. One way to capture this coupling is to optimize a constrained objective in which $J(\Theta, Z)$ is minimized subject to constraints on $|\mathcal{M}_{rs}(t)|$ and on cumulative costs, while recognizing that the achievable (Θ, Z) depend on the protocol's ability to coordinate. Another way is to incorporate communication costs as penalties in an augmented objective. The approach adopted here uses both perspectives: explicit constraints represent hard system limits, while soft penalties express preference among feasible solutions.

A key modeling decision concerns the granularity at which solutions are represented [17]. Directly encoding all assignments Z is impractical for metaheuristic search because Z scales with the dataset size. Instead, candidate solutions are represented by cluster parameters Θ, while assignments are computed locally by nearest-representative rules [18]. This produces a lower-dimensional search space and naturally matches distributed evaluation: given Θ, each repository can compute its local contribution to the distortion and the statistics needed to update Θ. The remaining issue is how repositories coordinate to evolve Θ under budgets. Swarm intelligence methods provide a mechanism: a population of candidate Θ values is maintained, fitness is evaluated in a distributed manner, and population members are updated using stochastic rules that depend on locally best and globally best candidates inferred via constrained communication [19].

The formulation also accommodates repository heterogeneity in scale. Some repositories may have many points and others few, suggesting weighted aggregation. Define $N = \sum_r n_r$ and weights $w_r = n_r/N$ or alternative importance weights reflecting reliability or priority [20]. Local distortion can be written as

$$J(\Theta) = \sum_{r=1}^{R} w_r\, J_r(\Theta), \tag{3}$$

$$J_r(\Theta) = \sum_{i=1}^{n_r} \ell\Big(y_{r,i}, \theta_{\arg\min_k \ell(y_{r,i},\theta_k)}\Big). \tag{4}$$

This representation makes clear that evaluating $J(\Theta)$ decomposes across repositories, while updating Θ requires coordination. The rest of the paper develops communication-aware constraints for such coordination and a swarm-guided method for searching Θ that is compatible with those constraints [21].

Table 7. Communication overhead per method on the SocialTrace dataset

Method	Messages / iteration	Total transferred data (MB)
Distributed k-means	4 500	920
Local-only clustering	0	0
Naïve global PSO	6 200	1 430
Proposed SI-PSO	3 100	610
Proposed SI-HYB	3 400	650

Table 8. Sensitivity of clustering to swarm population size

Population size P	NMI ↑	Silhouette ↑	Comm. data (MB) ↓
20	0.74	0.43	420
40	0.79	0.47	610
60	0.81	0.49	790
80	0.82	0.50	990
100	0.82	0.50	1 210

3. Communication-Aware Constraint Modeling

Communication-aware constraints arise from physical link capacities, latency sensitivity of the application, energy limits on devices, and policies governing what information may be transmitted. These constraints can be represented at different levels. At the per-message level, there may be a maximum size due to packetization or an upper bound on the dimensionality of transmitted summaries [22]. At the per-round level, there may be a budget on total transmitted bits. At longer horizons, there may be cumulative quotas or battery constraints [23]. A useful modeling approach is to represent all such limits as constraints on a set of communication variables that are themselves functions of the protocol and of the solution representation.

Let $t \in \{0, 1, \dots, T{-}1\}$ index clustering rounds. For each edge (r, s) and round t, define a decision variable $m_{rs}(t)$ representing the bit-length of the message sent from r to s. The protocol determines $m_{rs}(t)$ implicitly via the representation of messages, but treating $m_{rs}(t)$ explicitly allows constraints to be expressed directly. A basic per-edge capacity constraint is

$$m_{rs}(t) \le b_{rs}(t)\,\Delta t, \tag{5}$$

where Δt is the duration of a round [24]. If link availability is intermittent, then $b_{rs}(t) = 0$ for unavailable rounds, and the constraint naturally prevents sending. A per-node uplink budget can be expressed as

$$\sum_{s:(r,s)\in E} m_{rs}(t) \le \bar{b}_r(t), \tag{6}$$

where $\bar{b}_r(t)$ reflects shared bottlenecks such as a single radio interface. A global budget on weighted cost can be represented as

$$\sum_{(r,s)\in E} \kappa_{rs}\, m_{rs}(t) \le \bar{C}(t). \tag{7}$$

Latency sensitivity can be modeled by limiting the number of hops or by penalizing delayed information [25]. One representation uses an age-of-information variable $a_r(t)$ indicating the staleness of the global or neighborhood-best swarm state at repository r. The protocol and network jointly determine $a_r(t)$, but constraints can limit maximum staleness:

$$a_r(t) \le \bar{a}_r. \tag{8}$$

Alternatively, latency can be incorporated by weighting messages by their delay, recognizing that sending on a high-latency link may be less useful in fast-changing scenarios [26]. If $\delta_{rs}(t)$ is the realized delay of messages on (r, s), one can define an effective cost

$$C_{\text{eff}}(t) = \sum_{(r,s)\in E} \kappa_{rs}\, m_{rs}(t)\, g(\delta_{rs}(t))\,, \tag{9}$$

Table 9. Ablation of communication-aware components in the objective function

Configuration	Comm.-aware term	NMI ↑	Comm. data (MB) ↓
A1	disabled	0.82	1 450
A2	local penalty only	0.81	980
A3	global penalty only	0.80	870
A4	hierarchical penalty	0.82	720
A5 (proposed)	local + global	**0.82**	**650**

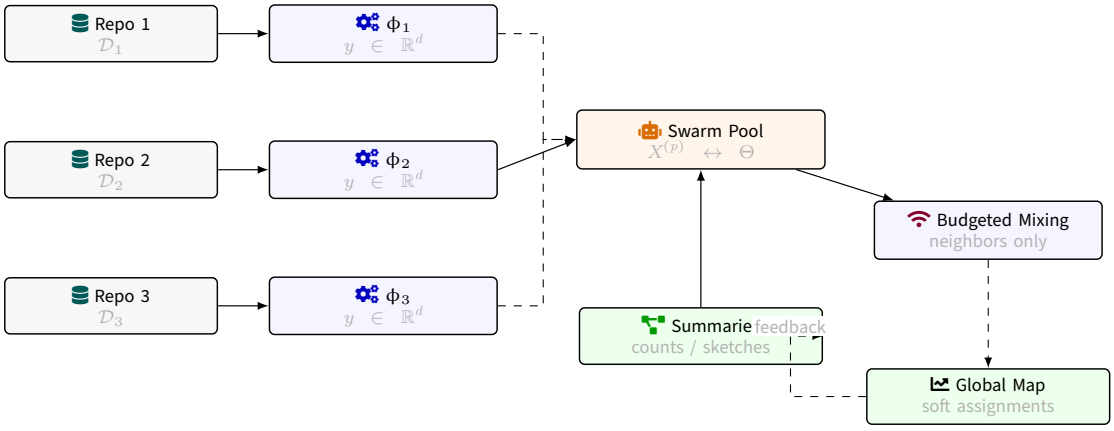


Figure 1. Distributed architecture: repositories embed local data into a shared space, evaluate swarm candidates using local summaries, and exchange only budgeted neighborhood information to refine a global clustering map.

where g is increasing, for example $g(u) = 1 + \rho u$ for a linear penalty. Such modeling supports applications where outdated centroid information is less valuable.

Message-size modeling requires connecting $m_{rs}(t)$ to the clustering representation. If a repository transmits K centroids in $\mathbb{R}^d$ with q-bit quantization per coordinate, then a naive representation yields $m_{rs}(t) \approx Kdq$ bits, plus overhead for framing. If centroids are transmitted sparsely or as differences from prior values, message sizes can be smaller and depend on the magnitude and sparsity of changes [27]. If sketches are used, such as random projections or coresets, then message size depends on sketch dimension and encoding. These relations can be captured by a function $m_{rs}(t) = \psi_{rs}(\Xi_{rs}(t))$, where $\Xi_{rs}(t)$ denotes the transmitted content and ψ_{rs} accounts for encoding. While the algorithm can treat ψ_{rs} abstractly, explicit forms are helpful for selecting parameters such as q and sketch dimension.

Policy constraints can be represented similarly. If direct transmission of raw $y_{r,i}$ is prohibited, messages must be computed via an allowable mechanism $\mathcal{A}_r$ that maps local data to summaries. If privacy constraints require randomized mechanisms, one can restrict messages to belong to a family $\mathcal{F}_r$ of distributions parameterized by a privacy level. Without adopting any single privacy formalism, the optimization can include a constraint that message-generating transformations must satisfy a divergence bound relative to a baseline mechanism, which indirectly limits leakage [28]. For instance, if $P_{rs}(t)$ is the distribution of messages from r to s at time t under the protocol, and $P^0_{rs}(t)$ is a reference distribution, then a constraint of the form

$$D\Big(P_{rs}(t) \,|0\, P^0_{rs}(t)\Big) \le \epsilon_{rs} \tag{10}$$

restricts how informative messages can be about individual records, where D is a divergence such as Kullback–Leibler or Rényi divergence. In practice, such constraints translate into limits on precision, on sketch dimension, or on the number of transmitted sufficient statistics [29].

With constraints expressed, the communication-aware clustering objective can be written as a constrained optimization problem over Θ and protocol variables. A

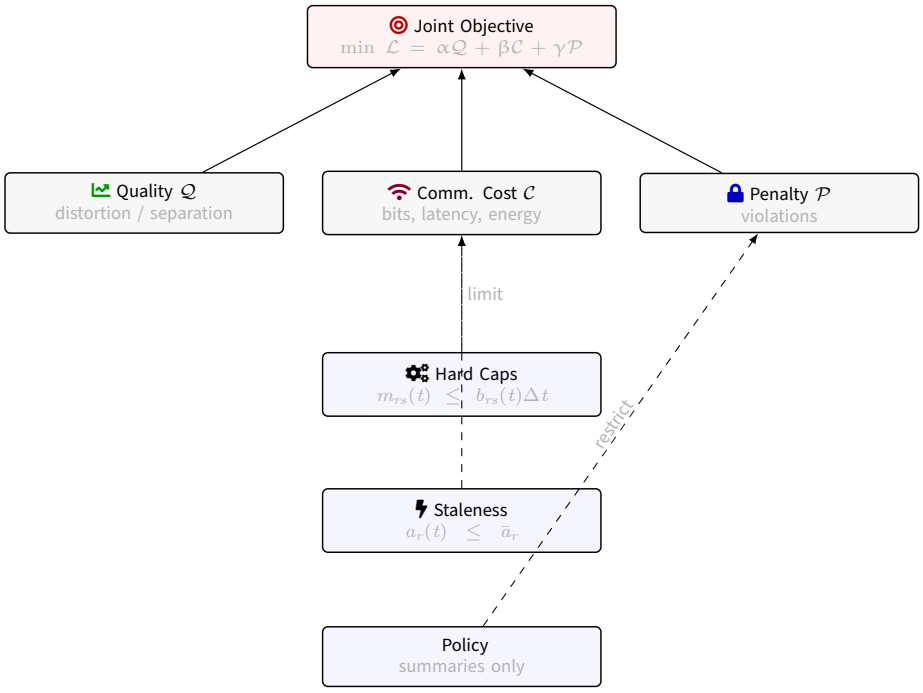


Figure 2. Objective coupling: clustering quality is balanced against communication cost, while constraint penalties encode cap exceedance, staleness limits, and admissible-message policies within a single optimization target.

representative formulation is

$$\begin{aligned} \min_{\Theta,\Pi} \quad & J(\Theta) + \lambda\,\mathbb{E}\left[\sum_{t=0}^{T-1}\sum_{(r,s)\in E} \kappa_{rs}\, m_{rs}(t)\right] \\ \text{s.t.} \quad & m_{rs}(t) \le b_{rs}(t)\,\Delta t \quad \forall (r,s),\ t, \\ & \sum_{s:(r,s)\in E} m_{rs}(t) \le \bar{b}_r(t) \quad \forall r,\ t, \\ & a_r(t) \le \bar{a}_r \quad \forall r,\ t, \\ & \mathcal{M}_{rs}(t) \sim P_{rs}(t) \in \mathcal{F}_r \quad \forall (r,s),\ t. \end{aligned} \tag{11}$$

The expectation captures randomness from network delays, randomized sketches, or swarm stochasticity [30]. The penalty parameter λ controls the relative weight of communication costs when the constraints admit multiple feasible protocols.

Because the protocol Π is itself complex, a direct optimization over Π is typically intractable. The swarm-guided method introduced later uses a restricted family of protocols in which messages consist of compressed swarm state and low-order sufficient statistics for centroid refinement [31]. The restriction makes $m_{rs}(t)$ predictable as a function of a small set of tunable parameters such as the number of shared particles, quantization levels, and mixing frequency. The constrained optimization then becomes a joint search over Θ and these parameters, with the swarm mechanism providing a way to explore Θ under fixed or adaptively adjusted communication settings. This framing also enables constraint-aware fitness shaping: infeasible candidates or candidates that require excessive communication to evaluate can be penalized, steering the swarm toward solutions compatible with the system limits.

4. Swarm-Intelligence–Guided Distributed Clustering Algorithm

Swarm intelligence methods maintain a population of candidate solutions and update them using stochastic rules inspired by collective behavior [32]. In the present context, each candidate solution corresponds to a cluster-parameter set Θ that defines a partitioning rule at each repository. The population-based nature of swarm methods is useful when objective evaluations are noisy or partial, which is common in distributed environments where repositories may be temporarily unavailable and where only compressed summaries can be exchanged. A further advantage is that different candidates can be evaluated using different subsets of repositories, enabling communication-aware sampling strategies [33].

Let there be P particles indexed by $p \in \{1, \ldots, P\}$. Particle p has a position vector $X^{(p)} \in \mathbb{R}^{Kd}$ representing concatenated centroids, and optionally a velocity vector $V^{(p)}$ if using particle swarm dynamics. The decoded centroids are $\theta_k^{(p)} \in \mathbb{R}^d$. Each repository maintains a local view of some subset of particles, which may be all particles or a smaller subset determined by budget. Denote by $\mathcal{P}_r(t)$ the set of particles known to repository r at time t. The repository evaluates local

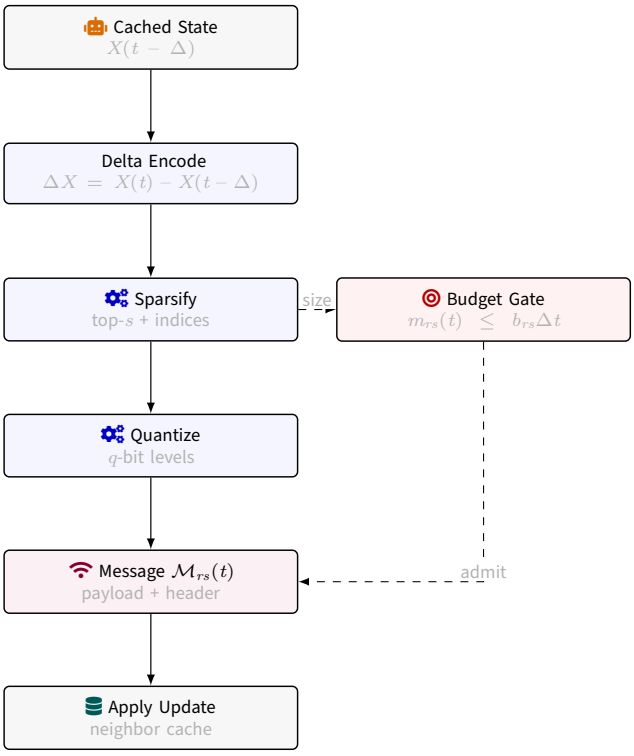


Figure 3. Message construction pipeline: particle updates are transmitted as sparse, quantized deltas and passed through a capacity gate, producing bounded-size packets for constrained links.

distortion contributions for these particles: [34]

$$\widehat{J}_r\Big(X^{(p)};t\Big) = \sum_{i=1}^{n_r} \ell\Big(y_{r,i}, \theta^{(p)}_{\arg\min_k \ell(y_{r,i},\theta_k)}\Big), \qquad (12)$$

possibly computed on a minibatch if n_r is large or if energy constraints apply. The global fitness for particle p ideally aggregates $\sum_r w_r \widehat{J}_r(X^{(p)};t)$, but communication constraints may prevent exact aggregation at every round. The method therefore uses a communication-aware approximation in which each particle is evaluated on a subset of repositories, and the contribution of non-participating repositories is either estimated using historical values or treated as missing with increased uncertainty.

Communication-aware fitness shaping is central [35]. For each particle p at time t, define an estimated distortion $\widetilde{J}(X^{(p)};t)$ computed from available repository contributions. Define also a communication accounting term $\widetilde{C}(p;t)$ representing the incremental communication required to evaluate and disseminate the particle update. A composite fitness can be defined as

$$F\Big(X^{(p)};t\Big) = \widetilde{J}\Big(X^{(p)};t\Big) + \lambda\,\widetilde{C}(p;t) + \mu\,\widetilde{\Phi}\Big(X^{(p)};t\Big), \qquad (13)$$

where $\widetilde{\Phi}$ is a constraint-violation measure capturing hard constraints such as per-round budgets or message admissibility. The parameter μ is selected large enough that infeasible behaviors are strongly discouraged relative to feasible alternatives. The specific form of $\widetilde{\Phi}$ depends on the protocol, but a common choice is a hinge penalty on budget exceedance.

The swarm update rule can be expressed in a particle-swarm style with constraint-aware modifications [36]. Let $P^{(p)}_{\text{best}}(t)$ denote the best position previously visited by particle p according to the composite fitness, and let $G_{\text{best}}(t)$ denote a best-known position within a neighborhood, which may be local to a repository or approximate global. A canonical update is

$$V^{(p)}(t+1) = \omega\, V^{(p)}(t) + c_1\, r_1(t)\,\big(P^{(p)}_{\text{best}}(t) - X^{(p)}(t)\big) + c_2\, r_2(t)\,\big(G_{\text{best}}(t) - X^{(p)}(t)\big) - \eta\,\nabla\widetilde{\Phi}\Big(X^{(p)}(t);t\Big), \qquad (14)$$

$$X^{(p)}(t+1) = X^{(p)}(t) + V^{(p)}(t+1), \qquad (15)$$

where ω is an inertia term, c_1 and c_2 are attraction coefficients, $r_1(t)$ and $r_2(t)$ are random vectors with i.i.d. components in $[0,1]$, and the final term introduces a drift away from constraint violations. When $\widetilde{\Phi}$ is non-differentiable, the drift can be replaced by a subgradient or by a projection step. The appearance of $\nabla\widetilde{\Phi}$ is not meant to force a fully gradient-based method; rather, it captures the idea that the swarm can incorporate constraint information directly rather than only through fitness ranking.

Distributed implementation requires specifying how

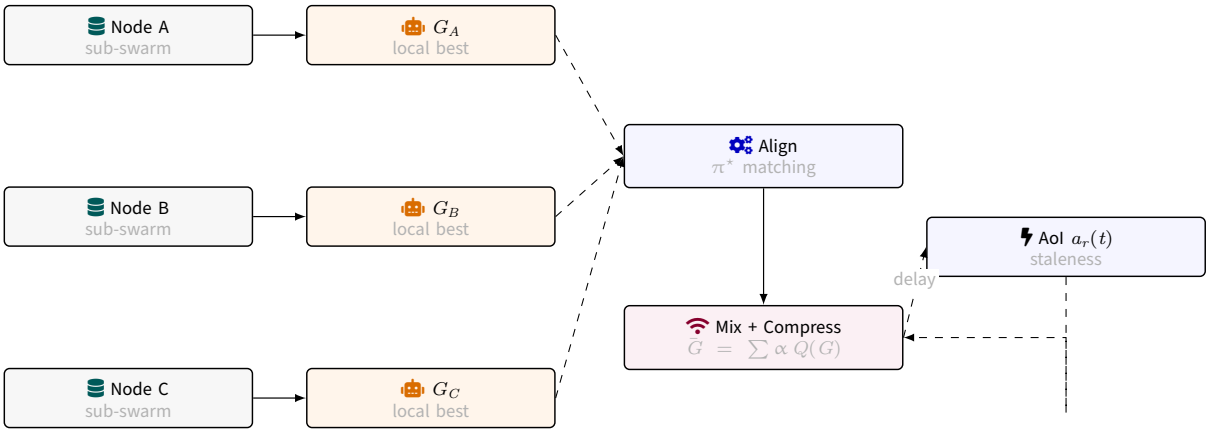


Figure 4. Neighborhood best formation: repositories exchange local best candidates, align centroid identities before mixing, and track staleness via an age-of-information variable that feeds back into subsequent mixing.

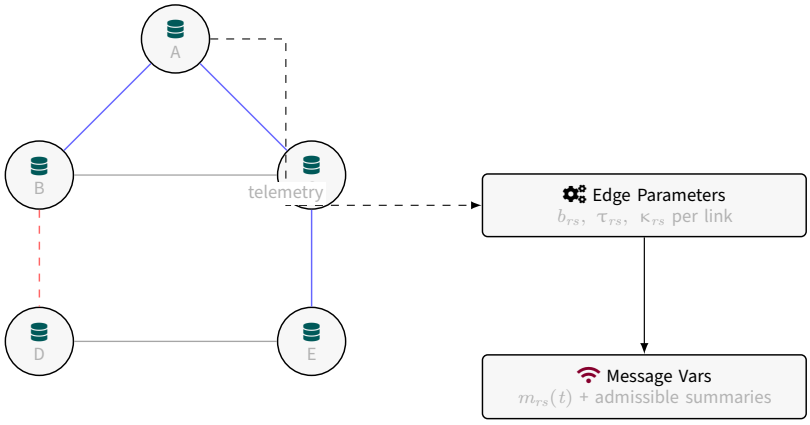


Figure 5. Communication graph abstraction: repositories connect through links with capacity, latency, and weighted cost; message-length variables and admissible summaries are constrained per link and used for budget-aware coordination.

$G_{\text{best}}(t)$ is computed. A fully global best requires aggregating fitness values across all repositories and broadcasting the corresponding particle position, which may be too expensive [37]. The proposed approach uses a multi-swarm structure. Each repository runs a local sub-swarm over a subset of particles and computes a repository-local best $G_r(t)$. Periodically, repositories exchange compressed representations of their local best candidates with neighbors, and use a consensus-like mixing rule to form a neighborhood best that approximates a global best over time [38]. If $\mathcal{N}_r$ denotes the outgoing or incoming neighbor set of repository r, a mixing update can be expressed as

$$\bar{G}_r(t+1) = \sum_{s \in \mathcal{N}_r \cup \{r\}} \alpha_{rs}(t)\, Q(G_s(t))\,, \qquad (16)$$

where $\alpha_{rs}(t)$ are weights that sum to one and respect connectivity, and $Q(\cdot)$ is a compression operator that quantizes or sparsifies the candidate position to meet message budgets. The operator Q can be unbiased in expectation to reduce systematic drift, but biased compression is also possible if it substantially reduces message size. The neighborhood best used in the particle update is then $G_{\text{best}}(t) = \bar{G}_r(t)$ for particles updated at repository r.

A key challenge is that mixing candidate positions by averaging is not always meaningful for clustering centroids, because permutation symmetries can cause centroid identities to swap between particles. To address this, the method incorporates a centroid-alignment step before mixing [39]. Given two centroid sets Θ and Θ', an alignment permutation π is computed to minimize matching cost, such as

$$\pi^\star = \arg\min_{\pi \in \mathcal{S}_K} \sum_{k=1}^{K} \left| 0\theta_k - \theta'_{\pi(k)} \right| 0_2^2, \qquad (17)$$

where $\mathcal{S}_K$ is the set of permutations. The aligned representation reorders centroids to a canonical or reference ordering, enabling sensible averaging and comparison [40]. In distributed settings, exact permutation search can be costly for large K, so repositories can use approximate matching based on greedy assignment or on low-dimensional signatures of centroids, which can be exchanged at small cost.

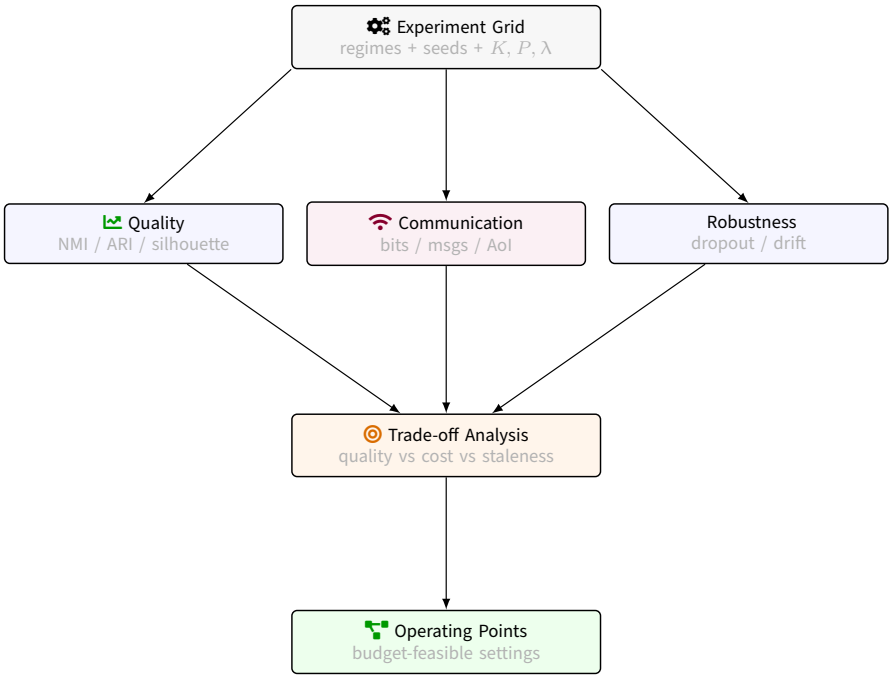


Figure 6. Evaluation workflow: experiments report clustering quality, communication consumption (including staleness), and robustness under failures or drift, summarized as feasible operating points across network regimes.

Constraint handling is integrated at multiple layers. At the swarm-update layer, constraint violation influences fitness and can add drift terms [41]. At the communication layer, the protocol enforces per-round budgets by controlling how many particles and which components are exchanged. For example, a repository may share only the best particle and a small number of promising alternatives, or it may share only a subset of centroid coordinates with the largest changes since the last exchange. At the evaluation layer, repositories can choose minibatch sizes adaptively so that compute and communication budgets are balanced, which is important when evaluation itself is expensive due to large n_r [42].

The algorithm also supports hybridization with ant-inspired or bee-inspired mechanisms by treating particle dissemination as an exploration-exploitation trade-off under communication constraints. For instance, repositories can allocate limited message budget to disseminate either exploitative information, such as the current best candidate, or exploratory information, such as diverse candidates that maintain population diversity across the network. Diversity is relevant because limited communication can cause premature homogenization around suboptimal solutions favored by a subset of repositories [43]. A simple diversity-preserving mechanism is to maintain multiple neighborhood bests corresponding to different regions of the search space, and to occasionally reinitialize a fraction of particles near underrepresented candidates using locally constructed perturbations. Communication-aware reinitialization can be guided by measuring disagreement between repositories' local best solutions, which indicates that the network is sampling different modes of the objective.

An important practical detail is the representation of candidate centroids [44]. Because Kd can be large, transmitting full positions at each mixing step may violate budgets. The method therefore uses incremental coding [45]. If $X^{(p)}(t)$ is the current particle position and $X^{(p)}(t–\Delta)$ is a previously shared position, then the message can transmit $\Delta X^{(p)} = X^{(p)}(t)–X^{(p)}(t–\Delta)$ after sparsification and quantization. If $S(\cdot)$ selects the top-magnitude components and $Q_q(\cdot)$ quantizes to q bits, then a transmitted update has the form $Q_q(S(\Delta X^{(p)}))$ along with indices. The resulting message size depends on the sparsity level and index encoding, and can be tuned to fit $m_{rs}(t)$ constraints. This incremental approach aligns with the observation that centroid updates often become smaller as the swarm approaches a stable region.

Overall, the swarm-intelligence–guided method is defined by the interaction of distributed evaluation, constraint-aware fitness shaping, and budgeted information exchange. The next section formalizes the protocol-level choices and shows how they map to the constraint variables introduced earlier [46].

5. Theoretical and Computational Analysis

The behavior of a swarm-guided clustering method under communication constraints is shaped by nonconvexity, stochastic evaluation, compression bias, and asynchrony. A full global optimality analysis is gen-

erally not available for such metaheuristics, but useful guarantees can still be discussed at the level of stability, boundedness, approximate stationarity, and constraint adherence. This section provides analytical considerations that help interpret algorithm behavior and guide parameter choices, focusing on what can be asserted under explicit assumptions [47].

A first point concerns the objective being optimized. Even if the underlying distortion $J(\Theta)$ is well-defined, the algorithm operates on an estimated fitness $F(X^{(p)}; t)$ that depends on partial repository participation and on stale or compressed information. It is therefore helpful to interpret the swarm dynamics as optimizing a time-varying surrogate objective. Let $\mathcal{R}_p(t)$ denote the set of repositories contributing to particle p's evaluation at time t, and let $\widehat{J}_{\mathcal{R}_p(t)}(X^{(p)}; t) = \sum_{r \in \mathcal{R}_p(t)} w_r \widehat{J}_r(X^{(p)}; t)$ be the partial distortion. Then the expected surrogate can be written as [48]

$$\mathbb{E}\Big[\widehat{J}_{\mathcal{R}_p(t)}(X^{(p)}; t)\Big] = \sum_{r=1}^{R} w_r \, \pi_{r,p}(t) \, J_r(X^{(p)}) + \text{noise terms}, \tag{18}$$

where $\pi_{r,p}(t) = \mathbb{P}(r \in \mathcal{R}_p(t))$ is the participation probability and the noise terms capture minibatching and compression. If participation probabilities are bounded away from zero over time for all repositories, and if the noise is unbiased or bounded, then the surrogate remains coupled to the true objective. When some repositories rarely participate due to persistent link limits, the surrogate can drift, and the algorithm effectively optimizes a reweighted objective. This observation motivates protocol designs that enforce minimum participation rates or that periodically allocate budget to underrepresented repositories [49].

Stability of particle swarm dynamics typically requires bounding velocities and preventing divergence in high-dimensional spaces. In centralized settings, sufficient conditions involve inertia ω and acceleration coefficients c_1, c_2 that keep the expected velocity bounded [50]. In the distributed setting, compression and staleness add perturbations. One useful approach is to introduce explicit clipping or damping. If a componentwise clip operator $\mathrm{clip}_{\nu}(\cdot)$ limits velocity magnitude to ν, then replacing $V^{(p)}(t+1)$ with $\mathrm{clip}_{\nu}(V^{(p)}(t+1))$ ensures bounded steps. Such bounding makes the induced process a Markov chain on a compact set if positions are also projected onto a bounded domain, which supports arguments about existence of invariant measures and long-run stability even when the chain is not reversible [51].

Constraint adherence can be analyzed at the protocol level and at the optimization level. Protocol-level constraints such as $m_{rs}(t) \le b_{rs}(t)\Delta t$ can be enforced by construction by choosing message formats whose worst-case size is below the budget. For example, if each mixing step transmits at most s nonzero coordinate updates per particle with q-bit quantization and with index coding overhead γ bits per coordinate, then a conservative bound is

$$m_{rs}(t) \le s\,(q + \gamma) + \text{framing overhead}, \tag{19}$$

and selecting s and q to satisfy the inequality guarantees feasibility regardless of particle values [52]. This approach avoids the need for penalty-based enforcement for hard link budgets, and moves the primary optimization trade-off to selecting s and the frequency of mixing.

Optimization-level constraints include staleness bounds and policy constraints. Staleness can be bounded by limiting the number of rounds between mixing events for each repository [53]. If repository r participates in a mixing round at least once every H_r rounds, and if delays are bounded by $\delta_{\max}$, then an age bound is

$$a_r(t) \le H_r + \delta_{\max}, \tag{20}$$

assuming age counts in rounds. This kind of bound is simple but informative: achieving smaller a_r requires more frequent mixing, which increases communication. Policy constraints on message admissibility can also be enforced by restricting message generators [54]. If messages are always produced by a fixed sketching and quantization pipeline with parameters that satisfy the policy, then admissibility is automatic. The remaining question is how these restrictions affect optimization.

Compression introduces bias and variance [55]. If Q is an unbiased compressor with $\mathbb{E}[Q(u)] = u$ and bounded second moment $\mathbb{E}|0Q(u) - u|0_2^2 \le \beta|0u|0_2^2$, then mixing updates behave like noisy consensus steps. Under connectivity assumptions on the communication graph over time, such noisy consensus can still bring repository estimates of best solutions closer in expectation, with an error floor that depends on β and on mixing frequency. Biased compressors can lead to systematic drift, but may still be acceptable if the bias is small relative to the scale of improvement in distortion and if periodic full-precision synchronization is feasible within long-horizon budgets [56].

The swarm dynamics can be related to stochastic approximation by considering a smoothed objective and interpreting the swarm's sampling of candidate solutions as approximating gradients or descent directions in expectation. While particle swarm updates are not gradient descent, one can still analyze expected movement toward attractors defined by best-known positions. If $G_{\mathrm{best}}(t)$ converges to a stable point $G^{\star}$ and particles' personal bests also converge, then the velocity recursion resembles a linear stochastic system whose stability can be assessed by examining eigenvalues of the associated linear operator, perturbed by random coefficients. In practice, the presence of constraint penalties and noise complicates the picture, but

the qualitative implication is that moderate inertia and bounded acceleration coefficients help prevent oscillation, especially when delays cause $G_{\text{best}}(t)$ to lag behind the current landscape.

Computational complexity separates into local evaluation cost and communication cost [57]. Local evaluation of a particle position requires computing distances between local points and K centroids. If computed exactly, the cost at repository r is $O(n_r K d)$ operations per particle per evaluation. With minibatching of size b_r, the cost becomes $O(b_r K d)$ and can be tuned to energy and latency [58]. If each repository evaluates $|\mathcal{P}_r(t)|$ particles per round, then compute per round is $O(|\mathcal{P}_r(t)|\, b_r K d)$. Communication depends on how many particles are shared and at what resolution. If each mixing event shares P_{share} particle positions and each position is encoded with s sparse updates, then per-edge communication per event is approximately $O(P_{\text{share}}\, s\,(q+\gamma))$ bits. The total depends on the number of active edges and the mixing schedule.

A further complexity consideration is centroid alignment [59]. Exact alignment via permutation optimization scales poorly with K, but approximate alignment can be amortized. One approach is to compute low-dimensional signatures of centroids, such as random projections, and match based on signatures, which reduces the cost of matching. The signature exchange itself consumes communication, but if signatures are much smaller than full centroid sets, the net cost can be favorable [60]. Alignment is most important during mixing and less important during local evaluation, so it can be performed only at mixing rounds.

Theoretical analysis also benefits from recognizing that clustering objectives often have symmetries and degeneracies [61]. For example, centroid permutations yield the same partition, and multiple centroid configurations can yield similar distortion. In distributed settings, different repositories may prefer slightly different centroid placements due to local data distributions. Communication-aware penalties can then encourage compromise solutions that reduce the need for frequent updates [62]. This effect can be formalized by introducing a regularizer that penalizes rapid centroid movement, which correlates with message size under incremental coding. If $\Theta(t)$ denotes the swarm's representative solution at time t, a temporal smoothness term can be

$$R_{\text{temp}}(t) = \sum_{k=1}^{K} \left|0\theta_k(t) - \theta_k(t-1)\right| 0_2^2, \tag{21}$$

and including $\alpha R_{\text{temp}}(t)$ in fitness implicitly reduces communication by promoting smaller updates. The regularizer is not universally desirable because it may slow adaptation to nonstationary data, but it provides a principled knob when communication is the binding constraint [63].

Overall, the analysis suggests that enforceable constraints are best handled by protocol design, while softer trade-offs can be incorporated via penalties that connect centroid dynamics to message size. Stability can be supported by bounding updates and by selecting mixing frequencies that keep staleness within acceptable limits. The next section translates these considerations into an evaluation methodology for comparing communication-aware clustering behavior across system regimes [64].

6. Evaluation Methodology and Discussion

Evaluating communication-aware clustering requires measuring both clustering quality and the resources consumed to achieve it. Because distributed repositories can differ widely in data distribution, feature mapping quality, and network connectivity, evaluation should cover multiple regimes rather than focusing on a single benchmark. This section describes an evaluation methodology and discusses the kinds of outcomes that can be expected, emphasizing how to interpret trade-offs rather than asserting any fixed numerical performance [65].

A basic experimental setup specifies a set of repositories, their datasets, and a network model. Repository datasets can be constructed by partitioning a larger dataset by source, time, geography, or synthetic labels, or they can be drawn from naturally siloed corpora. Heterogeneity can be controlled by varying class proportions, feature noise, missingness, or the mapping ϕ_r [66]. The network model should include edge capacities and latencies, and can include time-varying availability to simulate intermittent connectivity. Communication cost weights κ_{rs} can be chosen to reflect asymmetric costs such as cellular uplink versus local wired links. A useful practice is to define at least two network regimes: one where bandwidth is ample but latency is high, and another where bandwidth is tight but latency is modest, because these regimes stress different aspects of the protocol [67].

Metrics for clustering quality depend on whether ground truth labels exist. When labels are available, external validity measures can be computed by comparing cluster assignments to labels via agreement scores. When labels are not available, internal measures based on cohesion and separation can be used, such as average within-cluster distortion and silhouette-like measures computed in the embedding space [68]. In all cases, it is important to separate the quality of the final clustering solution from the quality of intermediate solutions, because communication-aware methods may converge more slowly or may exhibit plateaus due to budgeted mixing. Therefore, quality should be tracked as a function of both rounds and total communicated bits.

Communication metrics should be reported at mul-

tiple granularities [69]. Per-round total bits and per-edge utilization reveal whether constraints are being met and whether bottlenecks are localized. Cumulative bits over time indicate the total resource cost. If the protocol includes quantization and sparsification, it is helpful to report effective message sizes, including index overhead [70]. Latency-related metrics include age-of-information for the best-known global solution at each repository and the distribution of delays in incorporating updates. Energy-related metrics, when relevant, can be approximated by weighting transmitted bits and local compute operations using device-specific coefficients.

Baselines should include purely local clustering, in which each repository clusters its own data independently, and a centralized or fully synchronized method as an upper bound when feasible [71]. Between these extremes, communication-reduced distributed baselines include periodic averaging of centroids, gossip-based consensus on sufficient statistics, and hierarchical aggregation through a coordinator. The swarm-guided method should be compared under equal communication budgets to these baselines, because raw performance without budget normalization can be misleading [72]. For example, a method that communicates more often may appear to converge faster, but may violate realistic constraints.

Interpreting outcomes requires attention to how communication constraints shape the solution. In tight-bandwidth regimes, methods that rely on frequent full-precision centroid exchange tend to be dominated by overhead, and their effective progress per communicated bit can be low [73]. Communication-aware swarm methods may show slower progress per round but better progress per bit, especially when incremental and sparse updates capture the most salient centroid movements. In high-latency regimes, asynchrony and staleness become more important, and methods that can tolerate delayed information without destabilizing updates can maintain steady improvement. Multi-swarm structures can help here because local swarms continue exploring using local fitness even when global mixing is delayed [74].

Repository heterogeneity often creates tension between local and global optima. If repositories have systematically different data distributions, a globally optimal clustering under the combined objective may place centroids in compromise positions that are not optimal for any single repository. Communication-aware penalties can amplify this effect by favoring solutions that reduce the need for coordination, potentially leading to slightly higher distortion but lower overhead [75]. Such trade-offs are not necessarily undesirable; they reflect the system-level objective. Evaluation should therefore present Pareto-like comparisons, plotting distortion against communication cost and against staleness, to reveal whether the method provides useful operating points.

Nonstationary data streams present additional considerations [76]. If repositories receive new data over time, the clustering objective changes and methods must adapt. Communication-aware regularization that penalizes centroid movement may hinder adaptation, while aggressive mixing may violate budgets. A practical evaluation can simulate drift by gradually changing class proportions or feature distributions at subsets of repositories [77]. The swarm-guided method can then be assessed on its ability to track changes while respecting budgets, for example by adapting mixing frequency or by reallocating communication to repositories experiencing drift. Interpretation should focus on responsiveness measured in rounds and in communicated bits, rather than on final distortion alone [78].

Sensitivity analysis is important because swarm methods have hyperparameters such as swarm size P, inertia ω, and acceleration coefficients. Under communication constraints, these parameters interact with protocol parameters such as sharing frequency and quantization level. For example, increasing swarm size may improve exploration but also increases the cost of sharing particles, unless sharing is restricted to a small subset [79]. Similarly, finer quantization improves fidelity but consumes more bits. A systematic evaluation varies these parameters within feasible ranges and identifies stable regions where performance does not degrade sharply. Such analysis is particularly valuable in distributed settings where deployment conditions may differ from test conditions [80].

Finally, evaluation should consider failure and dropout. Repositories may temporarily disconnect, and links may fail. A communication-aware method should degrade gracefully, continuing to improve based on available repositories and reintegrating returning repositories without large disruptive jumps [81]. Swarm methods can support this by maintaining diversity and by allowing returning repositories to resynchronize via compressed snapshots. Experiments can simulate dropout by randomly disabling nodes or links and measuring both quality and constraint adherence during and after outages. The interpretation should emphasize whether the method maintains bounded communication and whether it avoids catastrophic divergence in centroid values [82].

Overall, the evaluation methodology treats communication as a first-class metric alongside clustering quality. The expected outcome of a well-tuned communication-aware swarm method is not necessarily the lowest possible distortion at any cost, but rather a set of solutions that provide controllable trade-offs under explicit budgets [83]. The presence of multiple operating points is a feature in this setting because system requirements can change with workload, network conditions, and policy.

7. Conclusion

Clustering over distributed repositories requires coordinating cluster structure across nodes that may be connected by constrained, heterogeneous, and time-varying networks. When communication is treated purely as an implementation concern, clustering methods can produce solutions that are difficult to realize within bandwidth, latency, energy, or policy limits [84]. This paper presented a communication-aware perspective in which the clustering objective is coupled to the cost of exchanging information needed to evaluate and refine candidate solutions. A swarm-intelligence–guided approach was developed around this perspective by representing candidate cluster parameters as a population, evaluating fitness using decomposed local distortion contributions, and coordinating improvement through budgeted exchange of compressed swarm state and aggregated statistics.

The proposed framing emphasizes two complementary mechanisms [85]. Hard communication limits can be satisfied by protocol design that constrains message formats, mixing frequency, and the number of shared candidates, producing feasibility by construction. Softer preferences and trade-offs can be expressed by penalties that connect centroid movement and synchronization behavior to communication cost, shaping the swarm's search toward solutions that are easier to maintain under constrained coordination. The distributed swarm structure, in which repositories run local sub-swarms and periodically mix best candidates through compressed consensus, supports continued progress under asynchrony and partial participation, while providing knobs to control staleness and overhead [86].

Analytical considerations highlighted how partial participation effectively reweights the objective and why protocol design should ensure that repositories are not persistently excluded when global structure matters. The discussion also clarified how compression influences convergence behavior through noise and bias, and how stability can be supported through bounded updates and projection. Computational complexity was separated into local evaluation cost and communication cost, enabling parameter selection that respects both compute and network budgets [87].

An evaluation methodology was outlined to measure clustering quality alongside per-round and cumulative communication, with attention to different network regimes, repository heterogeneity, and nonstationarity. Interpreting results in this framework focuses on operating points that balance distortion reduction against coordination cost and timeliness, rather than on a single unconstrained optimum. Future refinements can explore richer message representations, adaptive budget allocation based on measured disagreement or drift, and tighter integration between representation learning and communication-aware clustering, while maintaining the central principle that communication is an explicit constraint shaping both the optimization problem and the algorithm used to solve it [88].

References

[1] W. Humpage and K. P. Wong, "z-plane analysis of networks of distributed and discrete systems," Electronics Letters, vol. 18, pp. 502–504, 6 1982.

[2] C. Qian and Z. Xie, "A distributed framework of student data early warning system," Highlights in Science, Engineering and Technology, vol. 56, pp. 375–379, 7 2023.

[3] M. R. Thompson, W. E. Johnston, J. Guojun, J. Lee, B. Tierney, and J. Terdiman, "Distributed health care imaging information systems," in SPIE Proceedings, vol. 3035, pp. 97–106, SPIE, 5 1997.

[4] P. Shi, "Distributed coordination and learning on multi-agent systems," in 2023 IEEE 21st Jubilee International Symposium on Intelligent Systems and Informatics (SISY), pp. 11–12, IEEE, 9 2023.

[5] H. Abdu, S. Nawaz, and B. Elenbogen, "A framework to support adaption decisions for dynamic distributed systems," in Proceedings. Tenth International Conference on Parallel and Distributed Systems, 2004. ICPADS 2004., vol. 1437, pp. 477–484, IEEE, 11 2004.

[6] T. Wu and N. Xie, "Ve4pd: a distributed information system for collaborative product development," in SPIE Proceedings, vol. 5605, pp. 36–46, SPIE, 11 2004.

[7] M. J. Cunningham, "Distributed or centralized hvac: a practical investigation into household heating systems," 4 2023.

[8] S. Hu, L. Hou, G. Chen, J. Weng, and J. Li, "Mobiquitous - reputation-based distributed knowledge sharing system in blockchain," in Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, pp. 476–481, ACM, 11 2018.

[9] R. Malik, S. Kim, X. Jin, C. Ramachandran, J. Han, I. Gupta, and K. Nahrstedt, "Mlr-index: An index structure for fast and scalable similarity search in high dimensions," in International Conference on Scientific and Statistical Database Management, pp. 167–184, Springer, 2009.

[10] A. R. Zimmerman and J. P. Lynch, "Distributed model updating in smart wireless monitoring systems," in Structures Congress 2008, pp. 1–8, American Society of Civil Engineers, 10 2008.

[11] S. T. Jones, K. Gabert, and T. D. Tarman, "Evaluating emulation-based models of distributed computing systems," 8 2017.

[12] J. K. Bennighof and L. Meirovitch, "Active vibration control of a distributed system with moving support," Journal of Vibration and Acoustics, vol. 110, pp. 246–253, 4 1988.

[13] L. Yang, W. Tong, Z. Li, and T. Wu, "Authentication scheme for distributed industrial control system terminals," in Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering, pp. 1005–1012, ACM, 10 2021.

[14] A. R. del Nozal, L. Orihuela, and P. Millán, "Distributed estimation design for lti systems: a linear quadratic approach," International Journal of Systems Science, vol. 50, pp. 2703–2714, 10 2019.

[15] X. Yan, B. Chen, and Z. Hu, "Distributed estimation for interconnected dynamic systems under binary sensors," IEEE Sensors Journal, vol. 22, pp. 13153–13161, 7 2022.

[16] S. Liu and F. Oggier, An Overview of Coding for Distributed Storage Systems, pp. 363–383. Springer International Publishing, 1 2018.

[17] M. D. DeMerchant, A. W. Brown, X. Bao, and T. W. Bremner, "Automated system for distributed sensing," in SPIE Proceedings, vol. 3330, pp. 315–322, SPIE, 7 1998.

[18] S. K. Rahimi and F. S. Haug, Distributed Database Management Systems - The J2EE Starter Kit. Wiley, 7 2010.

[19] D. Hart, M. E. Tudoreanu, and E. Kraemer, "Agents - mobile agents for monitoring distributed systems," in Proceedings of the fifth international conference on Autonomous agents, pp. 232–233, ACM, 5 2001.

[20] C. W. Myles, C. Ebner, and P. A. Fedders, "Dynamics of a system of randomly distributed spins with multipolar interactions: Application to dipolar systems," Physical Review B, vol. 14, pp. 1–12, 7 1976.

[21] S. Chatterjee and J. K. Strosnider, "Distributed pipeline scheduling: A framework for distributed, heterogeneous real-time system design," The Computer Journal, vol. 38, pp. 271–285, 1 1995.

[22] X. Li, J. Yang, and J. Li, Design of Distributed Test System for Satellites, pp. 97–106. Germany: Springer Singapore, 5 2018.

[23] Y. Wu, M. Zhao, A. Haeberlen, W. Zhou, and B. T. Loo, "Sigcomm - diagnosing missing events in distributed systems with negative provenance," in Proceedings of the 2014 ACM conference on SIGCOMM, vol. 44, pp. 383–394, ACM, 8 2014.

[24] W. A. Pike, J. Bruce, B. Baddeley, D. M. Best, L. Franklin, R. May, D. M. Rice, R. M. Riensche, and K. Younkin, "The scalable reasoning system: lightweight visualization for distributed analytics," Information Visualization, vol. 8, pp. 71–84, 1 2009.

[25] V. Savchenko, O. Matsko, V. Vorobiova, Y. Kizyak, and V. Duhanets, "Assessment of performance of a distributed information system based on time profile," Eastern-European Journal of Enterprise Technologies, vol. 6, pp. 44–52, 11 2017.

[26] J. Dale and D. Dwyer, "Calibration and display of distributed aperture sensor systems," in SPIE Proceedings, vol. 6559, pp. 65590G–65590G, SPIE, 4 2007.

[27] E. Pagani and G. P. Rossi, "A set of multicast primitives for fault tolerant distributed systems," Journal of High Speed Networks, vol. 4, pp. 299–316, 1 1995.

[28] 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, IEEE, 2006.

[29] E. Jul, A. P. Black, D. Lea, R. E. Filman, and I. Welch, ECOOP Workshops - Panel: The Next 700 Distributed Object Systems, pp. 208–212. Germany: Springer Berlin Heidelberg, 7 2002.

[30] B. Shen, W. Huang, and X. Li, "Design and construction of distributed javascript parsing system," International Journal of Interdisciplinary Telecommunications and Networking, vol. 6, pp. 1–14, 10 2014.

[31] R. Wiebe, L. N. Virgin, and S. M. Spottswood, "Stochastic interrogation of competing responses in a nonlinear distributed system," Nonlinear Dynamics, vol. 79, pp. 607–615, 9 2014.

[32] L. Jin, Y. Tao, C. Wen, Y. Hong-yan, and W. Zhong-hai, "The research of the schemes of regional distributed energy system," MATEC Web of Conferences, vol. 175, pp. 04007–, 7 2018.

[33] P. Ribino and C. Lodato, "A distributed fuzzy system for dangerous events real-time alerting," Journal of Ambient Intelligence and Humanized Computing, vol. 10, pp. 4263–4282, 11 2018.

[34] E. Zerrik and M. Ouzahra, "Output stabilisation for distributed semilinear systems," IET Control Theory & Applications, vol. 1, pp. 838–843, 5 2007.

[35] W. Lohmiller and J.-J. E. Slotine, "Contraction analysis of nonlinear distributed systems," International Journal of Control, vol. 78, pp. 678–688, 6 2005.

[36] A. Bartoli, G. Dini, and L. Lopriore, "Single address space implementation in distributed systems," Concurrency: Practice and Experience, vol. 12, pp. 251–280, 4 2000.

[37] T. Srinivasan, R. Chandrasekar, V. Vijaykumar, V. Mahadevan, A. Meyyappan, and A. Manikandan, "Localized tree change multicast protocol for mobile ad hoc networks," in 2006 International Conference on Wireless and Mobile Communications (ICWMC'06), pp. 44–44, IEEE, 2006.

[38] F. Faghih and B. Bonakdarpour, "Smt-based synthesis of distributed self-stabilizing systems," ACM Transactions on Autonomous and Adaptive Systems, vol. 10, pp. 21–26, 10 2015.

[39] C. Y. Guo and H. B. Wu, "Capacity optimization of energy storage unit in distributed generation system," Advanced Materials Research, vol. 608-609, pp. 1116–1119, 12 2012.

[40] X. An, X. Yu, W. Song, L. Han, T. Yang, Z. Li, and Z. Su, "A software-defined distributed architecture for controlling unmanned swarm systems," Electronics, vol. 12, pp. 3739–3739, 9 2023.

[41] M. Hasanagić, T. Fabbri, P. G. Larsen, V. Bandur, P. W. V. Tran-Jørgensen, and J. Ouy, "Code generation for distributed embedded systems with vdm-rt," Design Automation for Embedded Systems, vol. 23, pp. 153–177, 11 2019.

[42] A. Bedrouni, R. Mittu, A. Boukhtouta, and J. Berger, Scalability and Performance Evaluation for the Coordination of Distributed Systems, pp. 37–52. Springer US, 8 2009.

[43] R. Chi, N. Lin, H. Zhang, and R. Zhang, Distributed Discrete-Time Adaptive ILC for Multi-Agent Systems, pp. 95–123. Springer Singapore, 3 2022.

[44] Y. Su, L. Wang, Z. Wang, Y. Liu, and X. Wang, "System error analysis and target localization of distributed pulse radars," Electronics Letters, vol. 60, 7 2024.

[45] X. Huang, Y. Yang, and G. A. Taylor, "Service restoration of distribution systems under distributed generation scenarios," CSEE Journal of Power and Energy Systems, vol. 2, pp. 43–50, 9 2016.

[46] T. Liu, Y. Lv, and G. Wen, "Fully distributed containment control of nonlinear multi-agent systems," in 2022 4th International Conference on Data-driven Optimization of Complex Systems (DOCS), pp. 1–5, IEEE, 10 2022.

[47] R. Chandrasekar and T. Srinivasan, "An improved probabilistic ant based clustering for distributed databases," in Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI, pp. 2701–2706, 2007.

[48] U. Vora, "Architecting a sds for microservices-based distributed edge computing systems," in 2024 11th International Conference on Software Defined Systems (SDS), pp. 67–71, IEEE, 12 2024.

[49] S. H. Shaw and J. S. Dorrler, "A distributed reusable-sorbent oil recovery system," International Oil Spill Conference Proceedings, vol. 1977, pp. 247–250, 3 1977.

[50] P. A. Fomitchov, Y. K. Kim, A. K. Kromine, S. Krishnaswamy, J. D. Achenbach, and I. M. Daniel, "Distributed photoacoustic system for cure monitoring of composites," in SPIE Proceedings, vol. 4335, pp. 323–329, SPIE, 7 2001.

[51] T. Cruz, J. Proenca, P. Simões, M. Aubigny, M. Ouedraogo, A. Graziano, and L. A. Maglaras, "A distributed ids for industrial control systems," International Journal of Cyber Warfare and Terrorism, vol. 4, pp. 1–22, 4 2014.

[52] V. Korotkikh and G. Korotkikh, "On principles in engineering of distributed computing systems," Soft Computing, vol. 12, pp. 201–206, 5 2007.

[53] V. Subramonian and C. Gill, "Fine-grained composition of distributed sensor system infrastructure," in IEEE MTT-S International Microwave Symposium Digest, 2005., vol. 1, pp. 377–380, IEEE, 6 2005.

[54] T. Islam, S. Bagchi, and R. Eigenmann, "Reliable and efficient distributed checkpointing system for grid environments," Journal of Grid Computing, vol. 12, pp. 593–613, 5 2014.

[55] R. van Engelen, M. Govindaraju, N. Koziris, and K. Psarris, "Session details: Distributed systems and grid computing (dsgc)," in Proceedings of the 2005 ACM symposium on Applied computing, ACM, 3 2005.

[56] S. Yu, Y. Zhou, and L. Wang, "Quantum-enabled distributed transient stability assessment of power systems," in 2024 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 593–599, IEEE, 9 2024.

[57] P. Reynolds, J. L. Wiener, J. C. Mogul, M. A. Shah, C. Killian, and A. Vahdat, "Experiences with pip: finding unexpected behavior in distributed systems," in Proceedings of the twentieth ACM symposium on Operating systems principles, pp. 1–2, ACM, 10 2005.

[58] A. Fagiolini, G. Dini, F. Massa, L. Pallottino, and A. Bicchi, "Distributed misbehavior monitors for socially organized autonomous systems," The International Journal of Robotics Research, vol. 43, pp. 2145–2182, 11 2024.

[59] M. Perry, Socially Distributed Cognition in Loosely Coupled Systems, pp. 19–41. Springer International Publishing, 3 2017.

[60] K. Leung, "Distvid : an examination of a distributed video streaming system," 6 2021.

[61] L. Silverberg and S. Park, "Globally optimal maneuver of distributed systems," in 30th Structures, Structural Dynamics and Materials Conference, American Institute of Aeronautics and Astronautics, 4 1989.

[62] Y. Zheng, Y. Shi, Y. Luo, and Z. Gan, "Coordination control for distributed energy based on multi-agent system," Chinese Science Bulletin, vol. 62, pp. 3711–3718, 9 2017.

[63] I. Alhajri and S. T. Taqvi, "Mathematical modeling and analysis of distributed energy systems for a refinery in kuwait.," ACS omega, vol. 6, pp. 19778–19788, 7 2021.

[64] I. Colonnelli and M. Aldinucci, "Workflow models for heterogeneous distributed systems," 5 2022.

[65] Z. Zuchao, J. Zhenshan, X. Bingjia, and W. Yong, "Upgrade of the distributed time-keeping and synchronization system for east," Plasma Science and Technology, vol. 15, pp. 1076–1080, 10 2013.

[66] R. Chandrasekar, R. Suresh, and S. Ponnambalam, "Evaluating an obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs," in 2006 International Conference on Advanced Computing and Communications, pp. 628–629, IEEE, 2006.

[67] J. He, M. A. Hiltunen, M. Rajagopalan, and R. D. Schlichting, "Middleware - providing qos customization in distributed object systems," Lecture Notes in Computer Science, pp. 351–372, 10 2001.

[68] F. Tian, W. Yu, J. Fu, W. Gu, and J. Gu, "Distributed optimization of multiagent systems subject to inequality constraints," IEEE transactions on cybernetics, vol. 51, pp. 2232–2241, 3 2021.

[69] S. B. Sarsilmaz and T. Yucelen, "A distributed control approach for heterogeneous linear multiagent systems," International Journal of Control, vol. 94, pp. 1402–1414, 8 2019.

[70] Q. Cai, J. Cao, G. Xu, and N. Zhu, "Distributed recommendation systems: Survey and research directions," ACM Transactions on Information Systems, vol. 43, pp. 1–38, 11 2024.

[71] A. G. Tartakovsky and V. V. Veeravalli, "Asymptotically optimal quickest change detection in distributed sensor systems," Sequential Analysis, vol. 27, pp. 441–475, 11 2008.

[72] G. Wei, H. Zhang, L. Liu, R. He, and D. Tang, "Optimization of the recovery time of pyramid code in distributed storage system," Journal of Physics: Conference Series, vol. 1634, pp. 012075–, 9 2020.

[73] O. Pei, Z. Li, W. Liu, and H. Du, "Cipae - a novel scalable distributed database system," in Proceedings of the 2020 International Conference on Computers, Information Processing and Advanced Education, pp. 31–36, ACM, 10 2020.

[74] J. Shen, Positive Systems with Distributed Delays, pp. 51–59. Springer Singapore, 3 2017.

[75] G. Chen and F. L. Lewis, "Distributed tracking control for networked mechanical systems," Asian Journal of Control, vol. 14, pp. 1459–1469, 5 2012.

[76] J. Jiang, Z. Liu, and W. Gao, Maintenance and Reliability in Distributed Energy Resource System, pp. 187–211. United States: Springer International Publishing, 4 2023.

[77] T. Ridge, "Popl - verifying distributed systems: the operational approach," in Proceedings of the 36th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, vol. 44, pp. 429–440, ACM, 1 2009.

[78] A. D. Kshemkalyani and M. Singhal, Distributed Computing: Principles, Algorithms, and Systems - Distributed Computing: Principles, Algorithms, and Systems. Cambridge University Press, 4 2008.

[79] K. Jung, "Oneos: a distributed operating system for the internet of things," in Proceedings of the 22nd International Middleware Conference: Doctoral Symposium, pp. 12–14, ACM, 12 2021.

[80] Z. Chen, P. Liu, Y. Chen, W. Zhan, G. Interdonato, and S. Buzzi, "Distributed learning-based beamforming codebooks for unevenly distributed users in mmwave massive mimo system," in 2024 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6, IEEE, 4 2024.

[81] B. Ding, L. Xie, and W. Cai, "Distributed model predictive control for constrained linear systems," International Journal of Robust and Nonlinear Control, vol. 20, pp. 1285–1298, 6 2010.

[82] G. Li, C. Zhou, J. Li, and B. Guo, "Maintaining data freshness in distributed cyber-physical systems," IEEE Transactions on Computers, vol. 68, pp. 1077–1090, 7 2019.

[83] C. Sirocchi and A. Bogliolo, Distributed Averaging for Accuracy Prediction in Networked Systems, pp. 130–145. Germany: Springer Nature Switzerland, 4 2024.

[84] X. Liu, Z. Li, G. Zhang, and N. Xie, "Measures of uncertainty for a distributed fully fuzzy information system," International Journal of General Systems, vol. 48, pp. 625–655, 4 2019.

[85] R. Malik, C. Ramachandran, I. Gupta, and K. Nahrstedt, "Samera: a scalable and memory-efficient feature extraction algorithm for short 3d video segments.," in IMMERSCOM, p. 18, 2009.

[86] C. Vogt, L. Wolf, R. G. Herrtwich, and H. Wittig, "Heirat—quality-of-service management for distributed multimedia systems," Multimedia Systems, vol. 6, pp. 152–166, 5 1998.

[87] G. Wang, Y. Liu, Y. Zhang, J. Yan, and S. Xie, "Distributed online optimization for integrated energy systems: A multi-agent system consensus approach," International Journal of Adaptive Control and Signal Processing, vol. 38, pp. 3401–3421, 7 2024.

[88] T.-Y. Yen and W. Wolf, "Sensitivity-driven co-synthesis of distributed embedded systems," in Proceedings of the Eighth International Symposium on System Synthesis, pp. 4–9, IEEE Comput. Soc. Press, 11 2002.