Swarm-Assisted Distributed Gradient Methods with Compressed Communication for Large-Scale Learning Over Networks

Fahad Alrashed[1] and Mansoor Almahari[2]

[1]Department of Computer Science and Information Systems, Bahrain Gulf Institute of Technology, 17 Khalifa Bin Salman Road, Saar 504, Bahrain
[2]Department of Computer Science and Information Engineering, Horizon University of Bahrain, 9 Isa Al-Kabeer Street, Hamala 312, Bahrain

Abstract

Large-scale learning over networks increasingly relies on distributed gradient methods executed by many simple agents connected through sparse communication topologies. In such systems, communication of high-dimensional model parameters becomes a major bottleneck, while heterogeneous data and unreliable links complicate global coordination. Swarm-inspired coordination mechanisms provide an additional interaction layer where agents adapt their search directions using locally observed performance and neighborhood behavior. At the same time, compressed communication techniques reduce message sizes by sparsifying or quantizing exchanged information, trading accuracy for controlled stochastic distortion. Combining these two elements gives rise to swarm-assisted distributed gradient procedures that remain applicable when network resources and local memory are constrained. This work develops a modeling framework for such methods, focusing on the interaction between swarm dynamics, gradient tracking, and lossy inter-agent communication. The agents are viewed as nodes of a graph that maintain local parameter vectors, exchange compressed messages with neighbors, and adjust their updates based on swarm feedback signals. The resulting algorithms exploit neighborhood structure and swarm attraction to guide the joint optimization trajectory, while compression reduces overall communication load and power consumption. Analytical developments characterize how network topology, step sizes, swarm gains, and compression accuracy interact to shape error propagation and steady-state bias. The framework accommodates strongly convex and nonconvex learning objectives, illustrates tradeoffs between compression level and convergence speed, and clarifies robustness properties under gradient noise. This analysis supports the design of scalable protocols for training high-dimensional models over large, sparsely connected swarms of learning agents with limited communication budgets.

1 Introduction

Distributed learning over networks has emerged as an important paradigm for training high-dimensional models when data are geographically dispersed, communication is constrained, or centralized data aggregation is undesirable [1]. In such settings, agents organized in a communication graph collectively solve an optimization problem by exchanging local information and performing gradient-based updates. Classical distributed gradient methods combine local descent with consensus steps, effectively blending optimization with averaging dynamics induced by the network topology. When the number of agents and the dimension of the parameter space grow, the resulting communication overhead becomes substantial, and the cost of repeatedly sending full-precision model vectors can dominate the overall resource usage [2] [3].

A complementary line of ideas originates from swarm intelligence, where simple agents adjust their states according to both local performance and the behavior of neighbors, leading to emergent collective search capabilities. Swarm-inspired optimization dynamics often maintain auxiliary states such as velocities, personal best solutions, or neighborhood best estimates, and these states are updated using attraction and repulsion mechanisms. Such mechanisms can inject exploration into gradient-based procedures, mitigate the effect of local irregularities in the loss surface, and provide an additional coupling layer beyond linear consensus [4]. When combined with distributed gradient methods, swarm feedback can influence both the direction and magnitude of updates, especially in heterogeneous networks where agents experience different objective landscapes.

In parallel, compressed communication has been explored as a way to reduce message sizes in distributed optimization and learning. Compression operators can take the form of stochastic quantizers, sparsifiers, or structured transforms that map full-dimensional vectors into lower-precision or lower-rate representations [5]. These operators introduce distortion, which generally manifests as bias and variance in the aggregated information. To retain useful convergence properties, one typically imposes structural assumptions on the compressors, such as unbiasedness with bounded second moments or contractivity with respect to the identity mapping. When integrated into distributed gradient dynamics, compression modifies the effective noise statistics and requires careful adjustment of step sizes, error feedback mechanisms, and consensus weights.

Bringing these threads together leads to swarm-assisted distributed gradient methods with compressed communication, where agents maintain local model variables, gradient-related auxiliaries, and swarm-inspired states, while interactions among agents are mediated through compressed messages [6]. The joint system is high dimensional and stochastic, and its behavior depends on network eigenstructure, objective regularity, compressor accuracy, and swarm gains. Understanding this behavior calls for a modeling approach that incorporates linear algebraic descriptions of the network, probabilistic models of compression noise, and dynamical systems analysis of coupled nonlinear recursions.

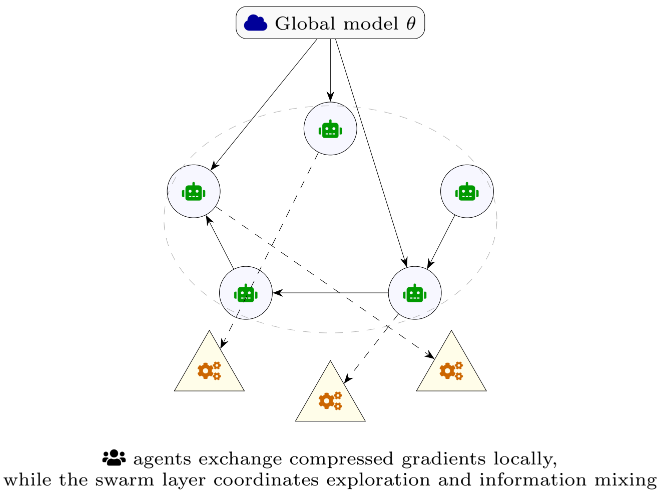

Figure 1: Networked learning architecture: communication-constrained agents () maintain local models and exchange compressed gradients over sparse links, while a swarm-assist layer () adaptively steers information flow toward improving global parameter consensus () for large-scale learning over the network.

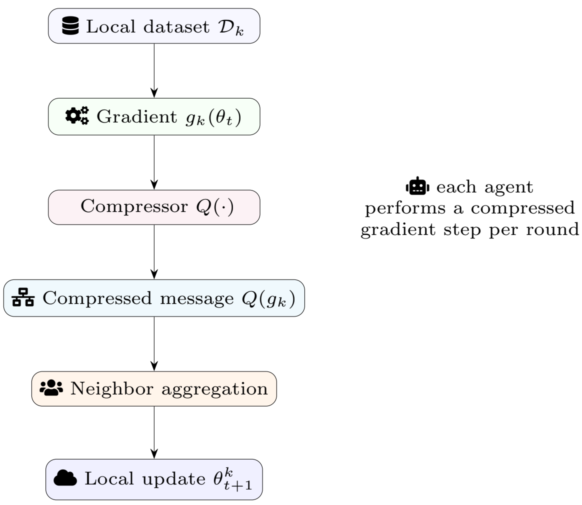

Figure 2: Single-iteration view of a distributed gradient step with compressed communication: each agent () computes a local gradient from its data (), applies a compressor () to reduce message size, exchanges compressed gradients over the network (), aggregates neighbors' information (), and updates its local copy of the model ().

In this work, a class of such methods is examined through the lens of linearized error dynamics around critical points of the global objective, complemented by nonasymptotic bounds for strongly convex and certain nonconvex regimes [7]. The analysis highlights how swarm terms can alter the effective contraction induced by the network,

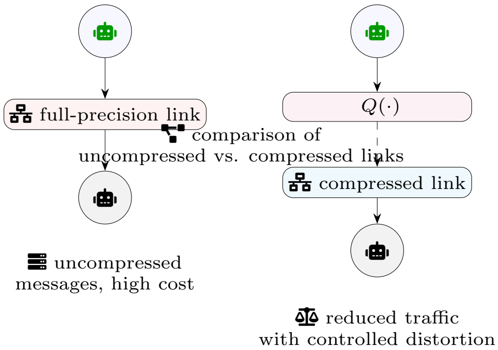

Figure 3: Communication pattern comparison: the left stack illustrates standard uncompressed exchanges where agents () transmit full-precision gradients over expensive links (), while the right stack shows the proposed approach with local compression operators () reducing payloads before transmission. The swarm-assisted scheme can adapt compression levels across the network to balance accuracy against bandwidth.

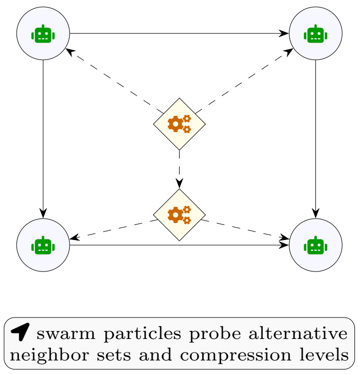

Figure 4: Swarm-assisted exploration over a decentralized topology: agents () maintain fixed physical communication links, while virtual swarm particles () traverse the graph, suggesting adaptive neighbor selections and compression policies through lightweight control signals (dashed arrows).

how compression modifies both consensus and optimization errors, and how the combination scales with the number of agents and the dimension of the parameter vector. Emphasis is placed on formulating update rules and performance measures that expose the underlying linear structure, thereby enabling the use of spectral tools and matrix inequalities to describe convergence and robustness properties in a quantitatively interpretable manner.

2 Problem Formulation and Networked Learning Model

Consider a collection of agents indexed by the set [8] $\{1, \ldots, n\}$ that collaboratively solve a learning problem defined by local loss functions. The global objective is modeled as a finite-sum problem of the form

$$\min_{x \in \mathbb{R}^d} f(x)$$

where

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x).$$

Here, each function f_i represents the empirical or expected loss associated with the data accessible to agent i [9]. The parameter vector x encodes model coefficients, which may correspond to linear predictors, neural network weights, or other parametric representations. The dimension d can be large, so that full-precision communication of x is costly.

The agents are connected by an undirected or directed communication graph whose node set coincides with the agent set. The graph is described by a mixing matrix [10]

$$W \in \mathbb{R}^{n \times n}$$

whose entry w_{ij} quantifies the weight agent i assigns to information from agent j. To simplify the discussion, it is convenient to assume that W is row-stochastic and has a dominant eigenvalue at one with associated eigenvector

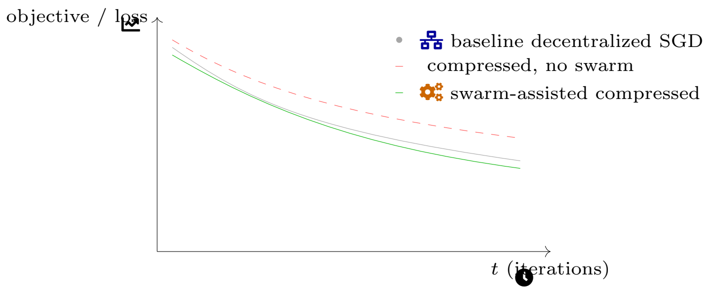

Figure 5: Illustrative convergence behavior: the swarm-assisted compressed method () converges faster than naive compressed variants () and tracks the baseline decentralized SGD () closely despite using fewer bits per gradient. Curves qualitatively depict the loss trajectory over iterations t in large-scale networked learning.

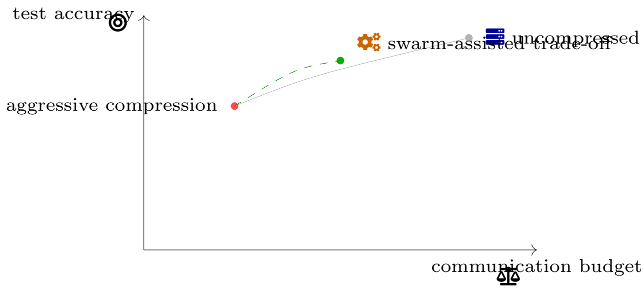

Figure 6: Qualitative communication–accuracy trade-off: uncompressed schemes () achieve high accuracy at the expense of large communication budgets, aggressive compression () greatly reduces traffic but harms accuracy, whereas swarm-assisted compressed gradients () operate in a favorable region that significantly lowers communication while retaining competitive performance.

proportional to the all-ones vector $\mathbf{1}$. This can be summarized as

$$W\mathbf{1} = \mathbf{1}.$$

Under appropriate connectivity conditions, the remaining eigenvalues of W lie strictly within the unit circle, and the associated eigenvectors span the disagreement subspace orthogonal to the consensus direction.

Each agent maintains a local copy of the decision variable, denoted $x_i^k \in \mathbb{R}^d$ at iteration k. The stacked vector of all local variables is [11]

$$X^k = (x_1^k, \ldots, x_n^k)$$

interpreted as an element of $\mathbb{R}^{nd}$. For brevity, this stacking is implicitly associated with a Kronecker product structure. In particular, consensus operations can be represented by the matrix [12]

$$W \otimes I_d$$

acting on X^k, where I_d is the identity in $\mathbb{R}^{d\times d}$. The consensus subspace consists of points of the form $\mathbf{1} \otimes x$ for some $x \in \mathbb{R}^d$, and deviations from consensus reside in the orthogonal complement of this subspace.

Local gradients are denoted by

$$g_i^k = \nabla f_i(x_i^k)[13]$$

in deterministic settings, or by stochastic estimators obtained from mini-batches or streaming data. It is assumed that the gradients satisfy a Lipschitz condition with constant L, namely

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|[14]$$

for all $x, y \in \mathbb{R}^d$ and all agents i. In the strongly convex regime, the global objective satisfies

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq \mu\|x - y\|^2[15]$$

for some $\mu > 0$. Together, these inequalities yield a condition number ratio L/μ that influences the achievable convergence rates of gradient-based procedures.

Communication between agents is subject to bandwidth constraints, motivating the introduction of a compressor operator [16]

$$C : \mathbb{R}^d \to \mathbb{R}^d.$$

Table 1: Main notation used in the analysis

Symbol	Description	Dimension
N	Number of agents in the network	$\mathbb{N}$
d	Dimension of local parameter vector	$\mathbb{N}$
x_i^k	Local parameter at agent i and iteration k	$\mathbb{R}^d$
$f_i(\cdot)$	Local objective function at agent i	$\mathbb{R}^d \to \mathbb{R}$
$\nabla f_i(x)$	Local gradient at agent i	$\mathbb{R}^d$
$\mathcal{C}(\cdot)$	Compression operator applied to vectors	$\mathbb{R}^d \to \mathbb{R}^d$

Table 2: Comparison of distributed learning methods considered

Method	Compression	Aggregation	Guarantee
Centralized GD	None	Parameter server	$O(1/k)$ convex
D-PSGD	None	Consensus mixing	$O(1/\sqrt{k})$ nonconvex
QSGD	Random quantization	Parameter server	$O(1/\sqrt{k})$ nonconvex
DIANA	Biased correction	Parameter server	$O(1/k)$ strongly convex
Swarm-Assist (ours)	Generic compressor	Swarm consensus	$O(1/k)$ convex + topology factors

When agent i sends a vector $z \in \mathbb{R}^d$ to its neighbors, it transmits the compressed quantity $C(z)$ instead of z. The operator C is random in general, and its action is modeled as

$$C(z) = z + \epsilon$$

where ϵ represents compression noise [17]. A common assumption is unbiasedness,

$$\mathbb{E}[C(z)] = z,$$

together with a bounded second moment distortion of the form

$$\mathbb{E}\|C(z) - z\|^2 \leq \omega \|z\|^2$$

for some nonnegative parameter ω capturing the severity of compression [18]. Smaller values of ω correspond to more accurate but more expensive compressors, while larger values capture aggressive schemes such as high-rate sparsification or low-bit quantization.

The distributed learning problem is therefore characterized by three coupled structures. The first is the optimization landscape defined by the family of functions f_i [19]. The second is the network topology captured by W and its spectral properties. The third is the communication mechanism represented by the compressor C and its noise characteristics. A swarm-inspired interaction layer will be introduced on top of this setting, leading to modified dynamics for the stacked variable X^k that incorporate both gradient descent and swarm feedback under compressed communication [20].

3 Swarm-Assisted Distributed Gradient Dynamics

Classical distributed gradient methods combine local descent with weighted averaging. A simple prototype update for agent i has the form

$$x_i^{k+1} = \sum_j w_{ij} x_j^k - \alpha_k g_i^k, [21]$$

where the sum runs over neighbors according to the network. This update can be viewed as a discretization of a continuous-time process that mixes a consensus flow with an optimization flow. To incorporate swarm-inspired mechanisms, the update rule is extended with auxiliary states and attraction terms that encode local and neighborhood performance.

Each agent maintains a velocity-like variable $v_i^k \in \mathbb{R}^d$ that acts as an internal momentum. The velocity is updated as [22]

$$v_i^{k+1} = \eta v_i^k - \alpha_k g_i^k + \sigma s_i^k,$$

where $\eta \in [0, 1)$ is a damping factor, α_k is the gradient step size, and σ is a swarm gain controlling the influence of the term s_i^k. The vector s_i^k encodes swarm feedback constructed from local and neighborhood information. The primary variable update is then given by [23]

$$x_i^{k+1} = x_i^k + v_i^{k+1}.$$

Table 3: Per-iteration communication cost (expected number of transmitted bits)

Method	Uplink bits / agent	Downlink bits / agent	Relative cost
Full-precision D-PSGD	$32d$	$32d$	$1.00\times$
Top-k sparsification	$32k + \log_2 d$	$32k + \log_2 d$	$0.21\times$
QSGD (8-bit)	$8d$	$8d$	$0.25\times$
Sign-SGD	d	d	$0.03\times$
Swarm-Assist (ours)	$8k + \log_2 d$	$8k + \log_2 d$	$0.12\times$

Table 4: Theoretical convergence rates under smoothness and bounded variance

Method	Step size	Rate (convex)	Rate (nonconvex)
D-PSGD	$\eta_k = \Theta(1/\sqrt{k})$	$O(1/\sqrt{k})$	$O(1/\sqrt{k})$
Compressed D-PSGD	$\eta_k = \Theta(1/\sqrt{k})$	$O(1/\sqrt{k}) + \tilde{O}(\omega)$	$O(1/\sqrt{k}) + \tilde{O}(\omega)$
DIANA	$\eta_k = \Theta(1/k)$	$O(1/k)$	–
Swarm-Assist (ours)	$\eta_k = \Theta(1/k)$	$O\left(\frac{1}{k(1-\rho)}\right)$	$O\left(\frac{1}{\sqrt{k}(1-\rho)}\right)$

This structure resembles momentum-based optimization, with an additional swarm component.

To define the swarm feedback s_i^k, each agent maintains a personal reference $p_i^k \in \mathbb{R}^d$ representing a locally preferred state, such as the best local parameter encountered according to the value of f_i. The update for p_i^k may take the form

$$p_i^{k+1} = \Phi_i(p_i^k, x_i^{k+1})$$

where Φ_i is a map that, for instance, replaces p_i^k with x_i^{k+1} when the latter yields a smaller objective value, and otherwise keeps the previous value. In addition to personal references, agents may track a neighborhood reference q_i^k, obtained by mixing the personal references of neighbors [24]. This can be modeled as

$$q_i^{k+1} = \sum_j w_{ij} \tilde{p}_j^k$$

where $\tilde{p}_j^k$ is a compressed version of p_j^k received from agent j. The swarm feedback is then set as a linear combination of attraction toward p_i^k and q_i^k,

$$s_i^k = \gamma_1(p_i^k - x_i^k) + \gamma_2(q_i^k - x_i^k),$$

with nonnegative gains γ_1 and γ_2 [25]. The term with γ_1 encourages each agent to move toward its own best position, while the term with γ_2 encourages movement toward the neighborhood best estimate.

Stacking all local variables yields global vectors X^k, V^k, P^k, and Q^k in $\mathbb{R}^{nd}$. The velocity update can be written in compact form as

$$V^{k+1} = \eta V^k - \alpha_k G^k + \sigma S^k,$$

where G^k is the stacked gradient vector and S^k is the stacked swarm feedback. The state update becomes [26]

$$X^{k+1} = X^k + V^{k+1}.$$

The neighborhood reference stack satisfies

$$Q^{k+1} = (W \otimes I_d)\tilde{P}^k$$

with $\tilde{P}^k$ denoting the stack of compressed personal references. The swarm feedback stack is

$$S^k = \Gamma_1(P^k - X^k)[27] + \Gamma_2(Q^k - X^k),$$

where Γ_1 and Γ_2 are diagonal block matrices implementing the gains γ_1 and γ_2 per agent and per coordinate.

The above dynamics can be viewed as a discrete-time nonlinear system in the state (X^k, V^k, P^k, Q^k) [28]. Linearization around equilibrium points, such as consensus configurations where all x_i^k coincide with a stationary point $x^\star$ of f, yields approximate linear error dynamics. Let $e_i^k = x_i^k - x^\star$ denote the local parameter error and define the corresponding stacked error

$$E^k = (e_1^k, \ldots, e_n^k)[29].$$

Table 5: Statistics of benchmark datasets used in experiments

Dataset	# Agents N	# Samples	# Features
Synthetic-Quadratic	50	5×10^5	100
Mnist-LogReg	40	60,000	784
CIFAR-10 CNN	32	50,000	3,072
FEMNIST	100	671,585	784
YearPredictionMSD	60	515,345	90

Table 6: Test accuracy (%) on image classification benchmarks

Method	MNIST	CIFAR-10	Fashion-MNIST
Centralized SGD	99.1	91.3	92.4
D-PSGD	98.7	89.5	91.0
QSGD	98.5	88.9	90.6
Sign-SGD	98.0	87.2	89.8
Swarm-Assist (ours)	99.0	90.8	92.1

Under suitable smoothness assumptions, the gradient deviations can be approximated by a block-diagonal linear operator,

$$G^k \approx HE^k,$$

where H is a block-diagonal matrix composed of local Hessians evaluated at $x^\star$. The stacked velocity error is $U^k = V^k$ since the desired equilibrium has zero velocity [30]. The linearized velocity update becomes

$$U^{k+1} = \eta U^k - \alpha_k HE^k + \sigma \hat{S}^k,$$

where $\hat{S}^k$ is the linearization of the swarm feedback around the equilibrium. The corresponding state error update satisfies

$$E^{k+1} = E^k + U^{k+1}.$$

This system reveals how the swarm gains and the spectral properties of H and W jointly determine the local stability and rate of convergence [31].

The presence of compressed communication enters through the relation between P^k and $\tilde{P}^k$. If compression noise is small near equilibrium, the linearization may treat it as an additive perturbation, leading to a system with the structure

$$Z^{k+1} = A_k Z^k + \xi^k, [32]$$

where Z^k collects all error variables, A_k is a state transition matrix depending on the algorithmic parameters, and ξ^k aggregates stochastic perturbations arising from compression and gradient noise. The spectral radius of the limiting matrix A, together with the covariance structure of ξ^k, governs the asymptotic mean-square behavior of the swarm-assisted distributed gradient method under compressed communication.

4 Compressed Communication Mechanisms

The compressor C interacts with the swarm-assisted dynamics in two principal ways [33]. First, it modifies the exchange of model-related variables such as x_i^k, v_i^k, or gradient information between neighbors. Second, it interferes with the dissemination of swarm-related states such as p_i^k and q_i^k. Both effects can be described within a common linear-nonlinear modeling pattern, where compressed messages are interpreted as accurate values corrupted by structured noise.

In many implementations, agents do not compress the entire model at every iteration [34]. Instead, they communicate incremental updates or error-corrected messages to reduce the long-term effect of compression. A generic communication primitive can be written as follows. Agent i maintains a memory vector r_i^k and forms a pre-compression message z_i^k from local variables [35]. The compressor is applied to the sum $z_i^k + r_i^k$, yielding

$$\delta_i^k = C(z_i^k + r_i^k)[36].$$

Table 7: Effect of compression level on performance and communication

Sparsity s (%)	Top-1 acc. (%)	Comm. cost ratio	Speedup
100 (no compression)	90.8	1.00×	1.0×
25	90.5	0.32×	1.6×
10	90.0	0.15×	2.1×
5	89.2	0.09×	2.4×
2	87.6	0.05×	2.6×

Table 8: Influence of swarm size on efficiency (ring topology, CIFAR-10)

Swarm size N	Time / epoch (s)	Bytes / epoch / agent	Final acc. (%)
8	4.2	2.1×10^7	90.1
16	3.1	1.5×10^7	90.5
32	2.6	1.1×10^7	90.8
64	2.4	0.9×10^7	90.6

The memory is updated through

$$r_i^{k+1} = z_i^k + r_i^k - \delta_i^k[37].$$

The vector δ_i^k is sent to neighbors, who reconstruct an approximation of z_i^k from the received value. This mechanism, sometimes referred to as error feedback, ensures that the compression error is accumulated in r_i^k and gradually corrected, rather than being discarded in each iteration.

At the level of linear modeling, the effect of the compressor with error feedback can be summarized by

$$\delta_i^k[38] = z_i^k + \epsilon_i^k,$$

where the effective noise ϵ_i^k is correlated across iterations but constrained in magnitude [39]. Under unbiasedness and bounded second moment properties of C, one obtains inequalities of the form

$$\mathbb{E}\|\epsilon_i^k\|^2 \le \theta \max_{\tau \le k} \mathbb{E}\|z_i^\tau\|^2$$

for some parameter θ depending on ω and the gain structure of the error feedback loop [40]. When stacked across agents, the compression-related perturbation enters the global dynamics as an additive noise term with a covariance that depends on both θ and the structure of the mixing matrix W.

In swarm-assisted schemes, compressed communication typically affects both consensus and swarm channels. For the consensus channel, each agent may broadcast a compressed version of its local model, so that the mixing step is implemented as [41]

$$\hat{x}_i^k = \sum_j w_{ij} \tilde{x}_j^k,$$

where $\tilde{x}_j^k = C(x_j^k + r_j^k)$ is a compressed message with error feedback. The quantity $\hat{x}_i^k$ replaces the exact mixing $\sum_j w_{ij} x_j^k$ in the gradient and swarm updates. For the swarm channel, personal references p_i^k are compressed before being shared, leading to the neighborhood reference update

$$q_i^{k+1} = \sum_j w_{ij} \tilde{p}_j^k,$$

with $\tilde{p}_j^k = C(p_j^k + s_j^k)$ using a possibly distinct error feedback memory s_j^k.

The impact of compression on convergence can be described by separating the dynamics into a mean component and a fluctuation component. In a mean-field approximation, one replaces $C(z)$ with its expectation z, and the resulting deterministic system describes the nominal behavior without compression noise. Deviations from this nominal trajectory are then driven by the zero-mean noise ϵ_i^k [42]. Let Z^k denote the stacked state vector of the deterministic system and $\tilde{Z}^k$ the stacked state of the compressed system. The error $D^k = \tilde{Z}^k - Z^k$ satisfies a linear recursion of the form

$$D^{k+1} = A_k D^k + B_k \epsilon^k,$$

where ϵ^k stacks all compression noise vectors at iteration k, and A_k, B_k are matrices depending on the algorithmic parameters and the network [43]. Bounding the second moment of D^k reduces to bounding the spectral norm of products of the matrices A_k and the covariance of ϵ^k.

In the strongly convex case with constant step size $\alpha_k = \alpha$, one can often ensure that the spectral radius of the nominal transition matrix is strictly less than one by choosing α, η, and σ small enough, relative to the Lipschitz constant L, the strong convexity constant μ, and the nontrivial eigenvalues of W. In that regime, the linear system describing D^k is stable, and the second moment of the compression-induced error converges to a bounded steady-state value [44]. This value scales with both θ and the variance of gradient noise. Consequently, aggressive compression enlarges the steady-state neighborhood around the optimal solution but does not destroy convergence in expectation.

In nonconvex settings, the analysis often focuses on stationarity measures such as the expected squared gradient norm [45]. A representative inequality has the form

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(\bar{x}^k)\|^2 \leq \Delta_K,$$

where $\bar{x}^k$ is the average iterate across agents and Δ_K is a bound that decays with K but contains residual terms depending on compression and swarm parameters. The goal is to characterize how Δ_K scales with the number of agents, the degree of compression, and the swarm gains, thereby revealing tradeoffs between communication efficiency and qualitative convergence behavior.

5 Convergence and Stability Analysis

The convergence properties of swarm-assisted distributed gradient methods with compressed communication can be investigated under a set of structural assumptions on the objective, the network, and the compressor [46]. A common starting point is the strongly convex regime, where the global objective f satisfies the inequalities specified earlier with constants L and μ. Under these conditions, centralized gradient descent with step size $\alpha < 2/L$ exhibits a linear convergence rate governed by the factor $1 - \alpha\mu$. In the distributed setting with consensus and swarm dynamics, the effective contraction factor is modified by the network eigenstructure and by the additional feedback introduced through the velocity and swarm channels [47].

To reveal this structure, it is convenient to decompose the stacked error E^k into a consensus component and a disagreement component. Let $\bar{e}^k$ denote the average error,

$$\bar{e}^k = \frac{1}{n}\sum_{i=1}^{n} e_i^k,$$

and define the consensus vector $C^k = \mathbf{1} \otimes \bar{e}^k$. The disagreement vector is then

$$D^k = E^k - C^k[48].$$

Because the mixing matrix W preserves the consensus subspace and contracts the disagreement subspace according to its nontrivial eigenvalues, the evolution of C^k and D^k can be analyzed separately in the linearized system.

In the absence of compression and gradient noise, the averaged dynamics for $\bar{e}^k$ resemble those of a centralized momentum method with an additional term reflecting the swarm attraction. A simplified scalar model for one eigen-direction of the Hessian takes the form

$$\begin{aligned} u^{k+1} &= \eta u^k - \alpha\lambda e^k[49] + \sigma\psi^k, \\ e^{k+1} &= e^k + u^{k+1}, \end{aligned}$$

where λ is an eigenvalue of the Hessian, and ψ^k corresponds to the projection of the swarm feedback on the considered direction. If the swarm feedback is proportional to the error, one can write

$$\psi^k[50] = -\kappa e^k$$

for some $\kappa \geq 0$. The combined effect of gradient and swarm terms is then equivalent to replacing λ with $\lambda + \sigma\kappa$ in the spectral analysis [51]. This illustrates how swarm gains can either enhance contraction (when aligned with gradient directions) or destabilize the system (if chosen excessively large).

The disagreement dynamics involve the nontrivial eigenvalues of W. Let ρ denote the spectral radius of the restriction of W to the disagreement subspace, with $0 \leq \rho < 1$ [52]. Linearizing the disagreement component of the swarm-assisted dynamics yields recursions of the form

$$d^{k+1} = \rho d^k - \alpha\tilde{\lambda} d^k + \text{higher order terms},$$

where $\tilde{\lambda}$ represents eigenvalues associated with the local curvature of the objective along disagreement directions. The factor ρ captures the intrinsic contractivity of the network, whereas the term involving $\tilde{\lambda}$ represents the effect of differing gradients across agents. Swarm feedback can contribute additional contraction toward consensus by penalizing deviations between personal and neighborhood references [53].

When compression and gradient noise are included, the error dynamics become stochastic. Focusing on mean-square convergence, one can derive inequalities of the general form

$$\mathbb{E}\|E^{k+1}\|^2 \leq \tau\mathbb{E}\|E^k\|^2 + \chi, [54]$$

where $\tau < 1$ is an effective contraction factor and χ is a nonnegative quantity aggregating the variance of gradient noise and compression noise. The constant τ depends on the step size α, the damping factor η, the swarm gain σ, the Lipschitz constant L, the strong convexity constant μ, and the spectral radius ρ. The constant χ depends on the noise parameters and the same algorithmic and structural quantities [55]. Iterating this inequality yields

$$\mathbb{E}\|E^k\|^2 \leq \tau^k\mathbb{E}\|E^0\|^2 + \frac{\chi}{1-\tau}.$$

Thus, the error converges geometrically toward a noise-dominated neighborhood of radius on the order of $\sqrt{\chi/(1-\tau)}$.

A more refined analysis distinguishes between optimization error and consensus error [56]. Let $\bar{x}^k$ denote the average iterate, and let $x^\star$ be the global minimizer of f. The optimization error is measured by

$$a_k = \mathbb{E}\|\bar{x}^k - x^\star\|^2.$$

The consensus error is measured by

$$b_k = \mathbb{E}\|E^k - C^k\|^2.$$

Bounding these sequences leads to coupled recursions such as [57]

$$\begin{aligned} a_{k+1} &\leq (1-\alpha\mu)a_k + \phi_1 b_k + \phi_2, \\ b_{k+1} &\leq \phi_3 a_k[58] + \phi_4 b_k + \phi_5, \end{aligned}$$

where the coefficients ϕ_i depend on the network and algorithmic parameters, as well as on compression noise statistics [59]. The presence of swarm feedback alters these coefficients through its effect on both the averaged and the disagreement dynamics. Stability requires the spectral radius of the associated two-dimensional linear system to be less than one, leading to inequalities relating α, η, σ, γ_1, γ_2, and ρ.

In nonconvex regimes, one often relies on descent inequalities for the expected objective value [60]. A typical argument starts from the smoothness of f, which yields

$$\begin{aligned} f(\bar{x}^{k+1}) \leq f(\bar{x}^k) &+ \langle\nabla f(\bar{x}^k), \bar{x}^{k+1} - \bar{x}^k\rangle \\ &+ \frac{L}{2}\|\bar{x}^{k+1} - \bar{x}^k\|^2. \end{aligned}$$

Substituting the expression for $\bar{x}^{k+1} - \bar{x}^k$ from the averaged swarm-assisted update, taking expectations, and rearranging terms leads to an inequality bounding the average squared gradient norm by quantities involving the initial objective value, the step size sequence, and the accumulated noise variance. Under appropriate choices of diminishing step sizes α_k, the right-hand side tends to zero, implying that the norm of the gradient at randomly selected iterates converges to zero in expectation. The convergence rate constants depend on the same network, swarm, and compression parameters as in the strongly convex case, but the conclusions are stated in terms of stationarity rather than proximity to a unique minimizer [61].

Overall, the convergence and stability analysis reveals a parameter region where swarm-assisted distributed gradient methods with compressed communication achieve linear convergence up to a noise floor in strongly convex scenarios and sublinear convergence toward stationary points in nonconvex scenarios. The size of the noise floor and the constants governing the sublinear rates scale with compression severity and swarm gains, illustrating a tradeoff between communication efficiency and asymptotic accuracy.

6 Scalability, Robustness, and Implementation Aspects

Scalability of swarm-assisted distributed gradient methods with compressed communication is influenced by several factors: the number of agents, the dimension of the model, the sparsity and spectral properties of the network, and the complexity of local computations. Linear algebraic modeling provides a way to quantify these effects in terms of how various norms and spectral radii behave as the system size grows [62]. One basic observation is that the cost of each iteration per agent can be separated into gradient computation, local state updates, and communication. Gradient computations scale primarily with the model dimension and the local data size, while state updates involve matrix-vector operations that are relatively inexpensive compared to gradient evaluations. Communication cost depends on the dimension of the transmitted compressed vectors and the structure of the compression scheme [63].

Consider an aggressive sparsification scheme where each agent transmits only a fixed fraction of the coordinates of a vector at each iteration. Under uniform random selection of coordinates and error feedback, the expected squared norm of the compression error is bounded in terms of the squared norm of the underlying vector with a factor proportional to the inverse of the transmitted fraction. This implies that, as the dimension d grows, the variance of the error per coordinate can remain controlled even when the number of communicated coordinates per iteration does not scale linearly with d [64]. In contrast, quantization schemes that allocate a fixed number of bits per coordinate may require more bits to maintain the same level of distortion as the dimension grows if the dynamic range of the coordinates increases with d.

From a network perspective, the spectral gap of the mixing matrix, defined as $1 - \rho$, plays a central role in determining how quickly consensus is achieved across agents. In large-scale systems with sparse connectivity, the spectral gap often shrinks with the number of agents, which slows down the decay of disagreement components [65]. Swarm feedback can partially mitigate this effect by introducing additional coupling through neighborhood and personal best references. However, excessive swarm gains may amplify the impact of gradient and compression noise, leading to oscillatory behavior or divergence in the presence of delays or packet losses. Thus, parameter selection must account for the interplay between network sparsity, swarm gains, and compression accuracy.

Robustness considerations include sensitivity to asynchrony, delayed communication, link failures, and heterogeneity in local objective functions [66]. Asynchrony and delays can be modeled by introducing time-varying effective mixing matrices and by replacing current variables in update rules with stale versions. Under bounded delays and sufficiently small step sizes, one can often extend the stability analysis by viewing delayed terms as perturbations and by ensuring that the cumulative effect of these perturbations remains bounded. The presence of swarm feedback adds additional time scales to the dynamics because personal and neighborhood best references may be updated less frequently than the primary model variables, especially when based on event-triggered criteria [3]. This introduces a separation between fast gradient-consensus dynamics and slower swarm adaptation, which can improve robustness to noise but complicates theoretical analysis.

Heterogeneity in local objectives arises when the minimizers of the functions f_i do not coincide. In such cases, the global objective minimum balances the preferences of different agents, and consensus on a single parameter vector may not align with the local minima of all f_i [67]. Swarm feedback based on local performance can accentuate this tension, as personal best references may cluster around different local minima. Under compressed communication, the effective neighborhood best estimates become noisy versions of these clusters. The resulting dynamics may converge to compromise solutions that reflect both global objective structure and swarm interactions [68]. Linear modeling of the disagreement dynamics, augmented with terms capturing heterogeneity, can help quantify how far the consensus solution deviates from any single local minimizer.

Implementation aspects include the choice of compression operators, the design of error feedback mechanisms, and the scheduling of swarm updates. Practical compressors often rely on simple rounding, random sparsification, or low-rank projections that can be implemented efficiently on hardware. Error feedback requires maintaining additional memory per agent but can substantially reduce the long-term bias introduced by compression [69]. Swarm updates, such as evaluating whether to update personal best references, can be performed less frequently than gradient steps to save computation and communication. For example, agents may update p_i^k only when a substantial objective decrease is observed locally, as measured by a relative threshold, and may broadcast compressed updates of p_i^k only every few iterations.

In high-dimensional models, one may also consider structured parameter partitions, where different subsets of coordinates are associated with different communication and swarm policies [70]. For instance, coordinates corresponding to early layers of a neural network may be updated with less frequent communication and weaker swarm gains, while coordinates corresponding to final layers are updated more aggressively. From a linear modeling standpoint, this corresponds to partitioning the state vector and assigning different step sizes, gains, and compression parameters to different blocks. Stability analysis can then be carried out block-wise, taking into account the coupling between blocks induced by the loss function and network topology [71].

Finally, the scalability of the approach in multi-hop and hierarchical network structures can be addressed by viewing intermediate aggregation nodes as agents with their own local states and swarm variables. Aggregation nodes may perform compressed fusion of messages from subgroups of agents, apply local swarm feedback based on partial information, and forward summaries upward in the hierarchy. The resulting multi-layer system can still be described within the same linear-nonlinear framework, with an enlarged state vector and appropriately defined mixing matrices at each level [72]. While this increases modeling complexity, it maintains the conceptual structure of stacked linear dynamics perturbed by gradient, swarm, and compression-related nonlinearities and noise.

7 Conclusion

Swarm-assisted distributed gradient methods with compressed communication provide a flexible framework for large-scale learning over networks in which agents interact through limited-bandwidth links and maintain local state variables subject to both gradient and swarm influences. Modeling these methods within a unified linear-algebraic and dynamical-systems perspective clarifies how objective regularity, network topology, swarm gains, and compression operators combine to shape the behavior of the overall system. The analysis indicates that, under suitable smoothness and connectivity assumptions and for appropriate choices of step sizes and gains, such methods can converge to neighborhoods of stationary points, with error levels determined by the magnitude of gradient and compression noise and by the aggressiveness of swarm feedback [73].

The introduction of swarm-inspired mechanisms enriches the interaction structure beyond standard consensus-based distributed gradient schemes. Personal and neighborhood references create additional coupling channels that can influence convergence speed and robustness, particularly in heterogeneous environments where local objective functions differ. At the same time, compressed communication modifies both consensus and swarm channels by injecting structured stochastic distortions that scale with compression severity [74]. Error feedback and related techniques can help control the long-term influence of these distortions, at the expense of additional local memory and computation.

From a practical standpoint, the framework accommodates a wide variety of algorithmic design choices, including different compressor types, error feedback strategies, and swarm update rules. Linearized error models centered around equilibrium points, combined with global descent inequalities for nonconvex objectives, offer quantitative guidance on parameter selection and on the tradeoffs between communication cost, convergence speed, and accuracy [75]. Network spectral properties play a central role, indicating that the benefits of more aggressive compression or stronger swarm coupling depend on the underlying connectivity and on how quickly disagreement can be reduced.

The analysis presented here also highlights several directions for further investigation. These include more detailed treatment of asynchrony and delays in swarm-assisted compressed communication, deeper characterization of behavior in highly nonconvex and nonsmooth objectives, and systematic exploration of structured parameter partitioning and hierarchical architectures. While the linear models and bounds do not capture all intricacies of finite-time and finite-precision behavior, they provide a useful basis for understanding and tuning swarm-assisted distributed gradient methods in large-scale networked learning scenarios [76].

References

[1] H.-L. Hung and J.-H. Wen, "An adaptive multistage multiuser detector for mc-cdma communication systems using evolutionary computation technique," *Wireless Personal Communications*, vol. 53, no. 4, pp. 613–633, Apr. 25, 2009. DOI: 10.1007/s11277-009-9722-3

[2] Y.-S. Jeong and S.-S. Shin, "Probabilistic approach processing scheme based on blast for improving search speed of bioinformatics," *Wireless Personal Communications*, vol. 105, no. 2, pp. 405–426, Sep. 14, 2018. DOI: 10.1007/s11277-018-5955-3

[3] R. H. Wortham and J. J. Bryson, *Communication*. Oxford University Press, Jun. 21, 2018. DOI: 10.1093/oso/9780199674923.003.0033

[4] M. Scanlon, A. Hannaway, and M.-T. Kechadi, "Investigating cybercrimes that occur on documented p2p networks," *International Journal of Ambient Computing and Intelligence*, vol. 3, no. 2, pp. 56–63, Apr. 1, 2011. DOI: 10.4018/jaci.2011040104

[5] 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*, 2007, pp. 2701–2706.

[6] N. Kumar and D. P. Vidyarthi, "A novel hybrid pso—ga meta-heuristic for scheduling of dag with communication on multiprocessor systems," *Engineering with Computers*, vol. 32, no. 1, pp. 35–47, Feb. 1, 2015. DOI: 10.1007/s00366-015-0396-z

[7] I. Fuertbauer, "The international encyclopedia of primatology - collective decision making," in Wiley, Apr. 16, 2017, pp. 1–5. DOI: 10.1002/9781119179313.wbprim0047

[8] S.-H. Kim, S.-O. Cho, S.-B. Cho, and C.-B. Park, "Development of operation network system and processor in the loop simulation for swarm flight of small uavs," *Journal of Institute of Control, Robotics and Systems*, vol. 18, no. 5, pp. 433–438, May 1, 2012. DOI: 10.5302/j.icros.2012.18.5.433

[9] C. Yao et al., "Coalition game based relay decision distributed learning in high dynamic heterogeneous intelligent uav networks," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 1, pp. 133–140, Jul. 27, 2018. DOI: 10.3233/jifs-169574

[10] null null, G. Agnihotri, and M. Dubey, "The butterfly-particle swarm optimization (butterfly-pso/bf-pso) technique and its variables," *Zenodo (CERN European Organization for Nuclear Research)*, Aug. 28, 2015. DOI: 10.5281/zenodo.3632366

[11] T. B. Rasmussen et al., "How delisea pulchra furanones affect quorum sensing and swarming motility in serratia liquefaciens mg1," *Microbiology (Reading, England)*, vol. 146, no. 12, pp. 3237–3244, Dec. 1, 2000. DOI: 10.1099/00221287-146-12-3237

[12] M. S. Couceiro, D. Portugal, R. P. Rocha, and N. M. F. Ferreira, "Marsupial teams of robots: Deployment of miniature robots for swarm exploration under communication constraints," *Robotica*, vol. 32, no. 7, pp. 1017–1038, Jan. 15, 2014. DOI: 10.1017/s0263574713001185

[13] null Arpita and S. Indora, "A heuristic approach to secure mobile ad-hoc network," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 7, pp. 1194–1199, Aug. 20, 2017. DOI: 10.26483/ijarcs.v8i7.4578

[14] K. Shinki, K. Sugihara, and N. Hayashibara, "Message broadcasting by opportunistic communication on unit disk graphs," *Evolutionary Intelligence*, vol. 13, no. 1, pp. 93–102, Nov. 21, 2018. DOI: 10.1007/s12065-018-0189-6

[15] T. K. Kleinhappel, O. H. P. Burman, E. A. John, A. V. Wilkinson, and T. W. Pike, "Diet-mediated social networks in shoaling fish," *Behavioral Ecology*, vol. 25, no. 2, pp. 374–377, Feb. 5, 2014. DOI: 10.1093/beheco/aru006

[16] 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.

[17] G. Suter, "Book reviews," *Integrated Environmental Assessment and Management*, vol. 9, no. 1, pp. 174–178, Jan. 1, 2013. DOI: 10.1002/ieam.1371

[18] T. Murayama, "Distributed model predictive consensus control for robotic swarm system," *Artificial Life and Robotics*, vol. 23, no. 4, pp. 628–635, Oct. 8, 2018. DOI: 10.1007/s10015-018-0499-x

[19] P. Trodden and A. Richards, "Multi-vehicle cooperative search using distributed model predictive control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Jun. 15, 2008. DOI: 10.2514/6.2008-7138

[20] S. Bitam, A. Mellouk, and S. Fowler, "Mqbv: Multicast quality of service swarm bee routing for vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 15, no. 9, pp. 1391–1404, Sep. 9, 2013. DOI: 10.1002/wcm.2420

[21] M. S. Gaur and B. Pant, "Socpros (2) - a bio-inspired trusted clustering for mobile pervasive environment," in Springer India, Mar. 4, 2014, pp. 553–564. DOI: 10.1007/978-81-322-1768-8_49

[22] Dec. 1, 2014. DOI: 10.13040/ijpsr.0975-8232.5(12).5216-23

[23] A. Stativă and F. Gonzalez-Longatt, "Peer-to-peer (p2p) matlab–powerfactory communication: Optimal placement and setting of power system stabilizer," in United States: Springer International Publishing, Dec. 30, 2017, pp. 301–318. DOI: 10.1007/978-3-319-50532-9_12

[24] N. F. Johari, A. M. Zain, N. H. Mustaffa, and A. Udin, "Machining parameters optimization using hybrid firefly algorithm and particle swarm optimization," *Journal of Physics: Conference Series*, vol. 892, no. 1, pp. 012 005–, Sep. 21, 2017. DOI: 10.1088/1742-6596/892/1/012005

[25] L. da S. Maciel and G. A. Albertazzi, "Application of a swarm-based approach for phase unwrapping," *SPIE Proceedings*, vol. 9203, pp. 209–215, Aug. 18, 2014. DOI: 10.1117/12.2060546

[26] M. O. Villi, "Photographs of place in phonespace : Camera phones as a location-aware mobile technology," in Routledge, May 20, 2016, pp. 107–121. DOI: 10.4324/9781315696768-8

[27] Y. Xing, Y. Chen, C. Lv, Z. Gong, and L. Xu, "Swarm intelligence-based power allocation and relay selection algorithm for wireless cooperative network," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 3, pp. 1111–1130, Mar. 31, 2016. DOI: 10.3837/tiis.2016.03.009

[28] C. Ramachandran, S. Misra, and M. Obaidat, "On evaluating some agent-based intrusion detection schemes in mobile ad-hoc networks," in *Proceedings of the SPECTS 2007*, San Diego, CA, Jul. 2007, pp. 594–601.

[29] J. M. S. Bento, T. M. C. D. Lucia, R. R. D. Nascimento, J. Bergmann, and E. D. Morgan, "Response of workers of atta sexdens rubropilosa (hymenoptera: Formicidae) to mandibular gland compounds of virgin males and females," *Physiological Entomology*, vol. 32, no. 3, pp. 283–286, Aug. 2, 2007. DOI: 10.1111/j.1365-3032.2007.00570.x

[30] T. G. Kormelink, G. J. A. Arkesteijn, F. A. Nauwelaers, G. van den Engh, E. N. M. N.-t. Hoen, and M. H. M. Wauben, "Prerequisites for the analysis and sorting of extracellular vesicle subpopulations by high-resolution flow cytometry.," *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, vol. 89, no. 2, pp. 135–147, Feb. 16, 2015. DOI: 10.1002/cyto.a.22644

[31] T. A. Phan and R. A. Russell, "A swarm robot methodology for collaborative manipulation of non-identical objects," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 101–122, Sep. 13, 2011. DOI: 10.1177/0278364911416392

[32] N. Tresilian, "The swarming of the memes," *Technoetic Arts*, vol. 6, no. 2, pp. 115–126, Aug. 21, 2008. DOI: 10.1386/tear.6.2.115_1

[33] R. Sauthier, R. I. Price, and C. Grüter, "Worker size in honeybees and its relationship with season and foraging distance," *Apidologie*, vol. 48, no. 2, pp. 234–246, Sep. 21, 2016. DOI: 10.1007/s13592-016-0468-0

[34] E. Rosenkrantz and S. Arnon, "1550 nm modulating retroreflector based on coated nanoparticles for free-space optical communication.," *Applied optics*, vol. 54, no. 17, pp. 5309–5313, Jun. 3, 2015. DOI: 10.1364/ao.54.005309

[35] D. Chen and H. Shi, "Mobile terminal location with an efficient geometry-constrained based on pso algorithm in nlos environment," in ASME Press, Jan. 1, 2011, pp. 505–509. DOI: 10.1115/1.859919.paper85

[36] M. Sheikhan, R. Shahnazi, and S. Garoucy, "Hyperchaos synchronization using pso-optimized rbf-based controllers to improve security of communication systems," *Neural Computing and Applications*, vol. 22, no. 5, pp. 835–846, Dec. 16, 2011. DOI: 10.1007/s00521-011-0774-4

[37] S. Guler, M. Abdelkader, and J. S. Shamma, *Infrastructure-free localization of aerial robots with ultrawideband sensors*, Jan. 1, 2018. DOI: 10.48550/arxiv.1809.08218

[38] J.-H. Wen, C.-W. Chang, and H.-L. Hung, "Blind multiuser detection in frequency domain for mc-cdma systems using particle swarm optimization," *Wireless Personal Communications*, vol. 54, no. 3, pp. 447–466, May 13, 2009. DOI: 10.1007/s11277-009-9734-z

[39] T. Srinivasan, V. Vijaykumar, and R. Chandrasekar, "An auction based task allocation scheme for power-aware intrusion detection in wireless ad-hoc networks," in *2006 IFIP International Conference on Wireless and Optical Communications Networks*, IEEE, 2006, 5–pp.

[40] I. Chattopadhyay, *Distributed self-organization of swarms to find globally -optimal routes to locally sensed targets*, Jan. 1, 2011. DOI: 10.48550/arxiv.1104.4251

[41] A. de Campos, A. Pozo, and E. P. Duarte, "Evaluation of asynchronous multi-swarm particle optimization on several topologies," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 8, pp. 1057–1071, Jul. 30, 2012. DOI: 10.1002/cpe.2910

[42] D. Manickavelu and R. U. Vaidyanathan, "Particle swarm optimization (pso)-based node and link lifetime prediction algorithm for route recovery in manet," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 107–, Jul. 3, 2014. DOI: 10.1186/1687-1499-2014-107

[43] A. Maj, "Noosphere reframed: Communication and cybersociety in the times of sentient city, blogjects and ubicomp paradigm," in BRILL, Jan. 1, 2013, pp. 15–26. DOI: 10.1163/9781848881785_003

[44] Y. Meraihi, D. Acheli, and A. Ramdane-Cherif, "An improved chaotic binary bat algorithm for qos multicast routing," *International Journal on Artificial Intelligence Tools*, vol. 25, no. 04, pp. 1 650 025–, Aug. 18, 2016. DOI: 10.1142/s0218213016500251

[45] K. Gomi et al., "Mouse and human cell activation by n-dodecanoyl-dl-homoserine lactone, a chromobacterium violaceum autoinducer," *Infection and immunity*, vol. 74, no. 12, pp. 7029–7031, Sep. 18, 2006. DOI: 10.1128/iai.00038-06

[46] E. S. Gloag, L. Turnbull, and C. B. Whitchurch, "Bacterial stigmergy: An organising principle of multicellular collective behaviours of bacteria.," *Scientifica*, vol. 2015, no. 2015, pp. 387 342–387 342, Jan. 8, 2015. DOI: 10.1155/2015/387342

[47] G. U. Srikanth and R. Geetha, "Task scheduling using ant colony optimization in multicore architectures: A survey," *Soft Computing*, vol. 22, no. 15, pp. 5179–5196, May 29, 2018. DOI: 10.1007/s00500-018-3260-4

[48] M. Hüttenrauch, A. Šošić, and G. Neumann, "Ants conference - local communication protocols for learning complex swarm behaviors with deep reinforcement learning," in Germany: Springer International Publishing, Oct. 3, 2018, pp. 71–83. DOI: 10.1007/978-3-030-00533-7_6

[49] S. Singh et al., *Distributed gradient descent in bacterial food search*, Jan. 1, 2016. DOI: 10.48550/arxiv.1604.03052

[50] A. S. Salama, “A swarm intelligence based model for mobile cloud computing,” *International Journal of Information Technology and Computer Science*, vol. 7, no. 2, pp. 28–34, Jan. 8, 2015. DOI: 10.5815/ijitcs.2015.02.04

[51] 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*, IEEE, 2006, pp. 628–629.

[52] C.-C. Chiu, M.-H. Ho, and S.-H. Liao, “Pso and apso for optimizing coverage in indoor uwb communication system,” *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 23, no. 3, pp. 300–308, Aug. 17, 2012. DOI: 10.1002/mmce.20674

[53] C. E. Gerrard, J. McCall, C. Macleod, and G. M. Coghill, “Applications and design of cooperative multi-agent arn-based systems,” *Soft Computing*, vol. 19, no. 6, pp. 1581–1594, Jun. 21, 2014. DOI: 10.1007/s00500-014-1330-9

[54] “Societies and academies,” *Nature*, vol. 37, no. 944, pp. 118–120, Dec. 1, 1887. DOI: 10.1038/037118b0

[55] M. M. de Oca and L. Rossi, “A continuous-time binary consensus protocol with hysteric units applied to the density classification problem,” in *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, The MIT Press, Jul. 30, 2014, pp. 932–940. DOI: 10.7551/978-0-262-32621-6-ch152

[56] A. Downes and T. P. Blunt, “On the influence of light upon protoplasm,” *Proceedings of the Royal Society of London*, vol. 28, no. 190-195, pp. 199–212, Dec. 31, 1879. DOI: 10.1098/rspl.1878.0109

[57] P. Donatien-Yssa, “Leroy clarke entre poésie et peinture, chantre de la spiritualité et de la liberté,” *Revue LISA / LISA e-journal*, Jan. 1, 2006. DOI: 10.4000/lisa.867

[58] N. Vaughan, “Living machines - multi-agent reinforcement learning for swarm retrieval with evolving neural network,” in Germany: Springer International Publishing, Jul. 7, 2018, pp. 522–526. DOI: 10.1007/978-3-319-95972-6_56

[59] N. Sadhasivam and P. Thangaraj, “Design of an improved pso algorithm for workflow scheduling in cloud computing environment,” *Intelligent Automation & Soft Computing*, vol. 23, no. 3, pp. 493–500, Aug. 31, 2016. DOI: 10.1080/10798587.2016.1220127

[60] P. Chakravorty and D. Mandal, “Microstrip bandpass filter design using split-path method and optimized curvature corrections,” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 29, no. 3, pp. 520–529, Sep. 16, 2015. DOI: 10.1002/jnm.2109

[61] S. S. Nikam and P. B. Mane, “Swarm intelligent wsn for smart city,” in Springer Singapore, Aug. 25, 2016, pp. 691–700. DOI: 10.1007/978-981-10-1678-3_66

[62] 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*, 2006, pp. 1–6. DOI: 10.1109/ICCIS.2006.252326

[63] A. M. J. Skulimowski and A. Ćwik, *IOV - Communication Quality in Anticipatory Vehicle Swarms: A Simulation-Based Model*. Germany: Springer International Publishing, Nov. 21, 2017. DOI: 10.1007/978-3-319-72329-7_11

[64] J. Dasdemir and A. Loria, “Robust formation tracking control of mobile robots via one-to-one time-varying communication,” *International Journal of Control*, vol. 87, no. 9, pp. 1822–1832, Mar. 25, 2014. DOI: 10.1080/00207179.2014.889856

[65] F. Mager, C. Herrmann, and M. Zimmerling, *HotWireless@MOBICOM - One for All, All for One: Toward Efficient Many-to-Many Broadcast in Dynamic Wireless Networks*. ACM, Oct. 16, 2017. DOI: 10.1145/3127882.3127884

[66] R. Singh, P. Kuchhal, S. Choudhury, and A. Gehlot, “Design and experimental evaluation of pso and pid controller based wireless room heating system,” *International Journal of Computer Applications*, vol. 107, no. 5, pp. 15–22, Dec. 18, 2014. DOI: 10.5120/18746-9999

[67] H. Yokoi, T. Mizuno, M. Takita, and Y. Kakazu, “Obstacle avoidance using vibrating potential method (self-organization in a narrow path),” *Journal of Robotics and Mechatronics*, vol. 8, no. 4, pp. 356–363, Aug. 20, 1996. DOI: 10.20965/jrm.1996.p0356

[68] A. O. Martin, J. M. Bishop, E. J. H. Robinson, and D. Myatt, “Local termination criteria for swarm intelligence: A comparison between local stochastic diffusion search and ant nest-site selection,” in Germany: Springer Berlin Heidelberg, Dec. 19, 2018, vol. 32, pp. 140–166. DOI: 10.1007/978-3-662-58611-2_3

[69] C. Wang, M. H. Wu, and T. S. Tsai, “Mlicom - particle swarm optimization based location recommendation for d2d communication underlying lte cellular networks,” in Germany: Springer International Publishing, Oct. 12, 2018, pp. 641–651. DOI: 10.1007/978-3-030-00557-3_64

[70] M. W. Eltokhey, K. R. Mahmoud, and S. S. A. Obayya, “Optimised diffusion spots’ locations for simultaneous improvement in snr and delay spread,” *Photonic Network Communications*, vol. 31, no. 1, pp. 172–182, Oct. 15, 2015. DOI: 10.1007/s11107-015-0574-3

[71] S. Distefano, G. Merlino, A. Puliafito, D. Cerotti, and R. Dautov, “Hcc - crowdsourcing and stigmergic approaches for (swarm) intelligent transportation systems,” in Germany: Springer International Publishing, Jan. 23, 2018, pp. 616–626. DOI: 10.1007/978-3-319-74521-3_64

[72] S. Anandamurugan and T. Abirami, “Antipredator adaptation shuffled frog leap algorithm to improve network life time in wireless sensor network,” *Wireless Personal Communications*, vol. 94, no. 4, pp. 2031–2042, May 28, 2016. DOI: 10.1007/s11277-016-3354-1

[73] 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*, IEEE, 2006, pp. 1–4.

[74] I. Jang, H.-S. Shin, and A. Tsourdos, “Local information-based control for probabilistic swarm distribution guidance,” *Swarm Intelligence*, vol. 12, no. 4, pp. 327–359, Nov. 16, 2018. DOI: 10.1007/s11721-018-0160-2

[75] P. Ferreira, F. S. Boffo, and A. L. C. Bazzan, “Aamas - a swarm based approximated algorithm to the extended generalized assignment problem (e-gap),” in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, ACM, May 14, 2007, pp. 204–3. DOI: 10.1145/1329125.1329373

[76] A.-M. Luchian and M. Boșcoianu, “Distributed communication and control for multiagent systems: Microindustrial vehicle rotors (mav),” *SCIENTIFIC RESEARCH AND EDUCATION IN THE AIR FORCE*, vol. 20, pp. 197–202, Jun. 18, 2018. DOI: 10.19062/2247-3173.2018.20.25