RESEARCH PAPER

Resilient Swarm Consensus and Distributed Optimization Under Byzantine Adversaries in Large-Scale Networked Systems

Brian Mwangi[1] and Kevin Otieno[2]

[1]Mount Kenya Institute of Technology, Department of Computer Science, Nyeri–Nyahururu Road, Nyeri, Kenya
[2]Coastal University of Digital Innovation, Department of Information Systems, Mombasa–Malindi Highway, Mombasa, Kenya

Abstract

Swarm robotic systems and distributed sensor networks are increasingly deployed in environments where communication is unreliable and node integrity cannot be guaranteed. Many practical platforms rely on local averaging and gradient-based update rules to achieve coordination, yet these mechanisms are vulnerable to malicious or faulty agents that deviate arbitrarily from prescribed protocols. Classical consensus and distributed optimization approaches therefore do not directly apply when adversaries can inject arbitrary signals, manipulate local memories, or corrupt messages in transit. There is ongoing interest in algorithmic structures that retain linear update simplicity while tolerating such Byzantine behaviors in large-scale networks. This work investigates resilient consensus and distributed optimization in swarms subject to worst-case adversarial behavior under a fairly general linear interaction model. The focus is on maintaining coherence of the normal agents while adversaries attempt to disrupt convergence or steer the swarm to undesirable states. The paper develops a modeling framework that captures local filtering, truncated aggregation, and adversarial isolation through effective linear operators, and it interprets many existing resilient mechanisms through this lens. The analysis emphasizes how structural properties of the interaction graphs, such as redundancy and sparsity, combine with algorithmic parameters to determine achievable resilience levels and convergence rates in large-scale settings. The discussion covers both static and slowly time-varying networks and includes remarks on computational and communication complexity. The overall perspective highlights trade-offs between resilience, scalability, and performance for consensus and distributed optimization in adversarial environments.

1. Introduction

Networked systems composed of many simple agents are now used in surveillance, environmental monitoring, cooperative exploration, and infrastructure management (1). In such settings there is often no central controller with complete information. Instead, each agent exchanges state information with nearby peers and updates its own state according to a local rule. Over time this process is expected to drive all nonfaulty agents to agreement or to a common optimizer of a shared objective (2). The basic premise of distributed consensus and optimization is therefore that global coordination can emerge from repeated local averaging and gradient steps.

This premise is challenged when the network is exposed to failures or malicious attacks. An adversary may compromise a subset of nodes, alter their internal logic, and cause them to transmit arbitrary values (3). Unlike benign faults such as packet drops or delays, Byzantine behavior is not constrained by any predefined error model. A compromised node can behave correctly for long periods of time and then deviate in carefully chosen directions, possibly in a coordinated fashion with other compromised nodes. Classical convergence arguments that rely on linearity and stochasticity of the weight matrices no longer hold when some entries in the update rule correspond to arbitrary values (4).

The large population nature of swarms adds an additional dimension. The same mechanism that makes consensus scalable, namely the use of simple linear iteration with sparse local interactions, also makes it difficult to protect every component. Nodes may be low-cost and computationally limited, communication channels may be lossy, and cryptographic authentication may not be practical at every link (5). At the same time, coordinated adversaries may be able to strategically compromise a fraction of the network. In such circumstances, it is natural to ask what forms of linear or piecewise linear update rules can still guarantee agreement or approximate optimality for the nonfaulty agents.

One line of research imposes structural constraints on the underlying communication graph. For example, if each normal node has sufficiently many independent neighbors whose information can be cross-validated, then outlier rejection or median type filters can be used to attenuate adversarial influence (6). Another line of work imposes redundancy at the level of local objective functions, making it more difficult for an adversary to alter the global optimum. These ideas can be combined within a unified linear algebraic framework that models the evolution of the swarm state as the repeated application of state-dependent operators on the vector of agent variables.

The present work examines resilient swarm consensus and distributed optimization under Byzantine adversaries from such a linear operator perspective (7). The agents of interest are those that follow a prescribed update rule, called normal agents. A subset of agents may be adversarial and is allowed to send arbitrary messages to its neighbors, possibly dependent on the entire history of the system. The normal agents implement updates that combine linear averaging, gradient descent, and local filtering based on simple comparisons among neighbor values (8). Even though the filtering makes the update rule nonlinear in general, it can be represented as a selection among a finite set

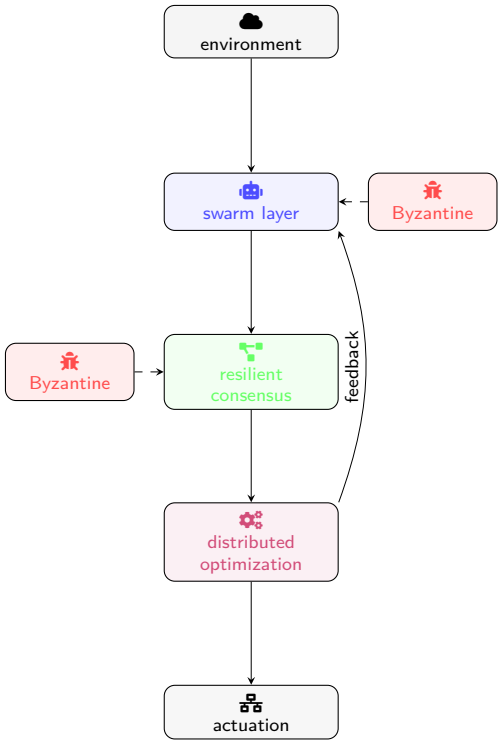


Figure 1. Layered view of the swarm, resilient consensus, and distributed optimization stack with localized Byzantine perturbations.

of linear operators that depend on the configuration of neighbor states.

The central questions addressed are the following. Under what structural conditions on the communication graph and the set of potentially adversarial nodes can normal agents reach approximate consensus in the presence of arbitrary adversarial inputs (9). How do these conditions change when the agents seek to collectively minimize a global objective that decomposes into private convex functions, while some agents are adversarial. How do the properties of the resulting dynamics scale with the network size. These questions are explored through linear modeling of the update rules, spectral properties of effective weight matrices, and contraction arguments. The overall aim is not to provide a single universal algorithm, but rather to clarify what types of linear update mechanisms can be made resilient in large-scale adversarial settings (10).

2. System Model and Problem Formulation

Consider a network of agents indexed by the set

$$V = \{1, \dots, n\}.$$

Each agent holds a state variable in a Euclidean space. For simplicity, the discussion uses scalar states and then comments on vector-valued extensions (11). Let

$$x \in \mathbb{R}^n$$

denote the stacked vector of agent states, where the entry associated with agent i is written

$$x_i \in \mathbb{R}.$$

The communication pattern among agents is described by a directed graph (12). When there is an edge from node j to node i, node i can receive information from node j. The set of in-neighbors of agent i is written as a fixed set

$$\mathcal{N}_i \subseteq V.$$

The self-index i may be included in this set for notational convenience (13).

In the absence of adversaries, a standard linear consensus update has the form

$$x = Wx,$$

where W is a nonnegative matrix with row sums equal to one. The entry w_{ij} is nonzero only when there is a communication link from node j to node i. In typical designs, the scalar w_{ij} is chosen according to a simple rule such as uniform weights over neighbors or weights based on local degree. The iteration is then a Markov chain in state space, and classical convergence results follow when the underlying graph is strongly connected and aperiodic (14).

In the presence of Byzantine agents, the update for a normal agent must be modified. Let $N \subseteq V$ denote the set of normal agents and let $B = V \setminus N$ denote the set of potentially adversarial agents. The adversarial agents may deviate arbitrarily from any prescribed protocol. For a normal agent i, the messages received from neighbors are no longer guaranteed to reflect their actual states (15). Instead, for each incoming neighbor j, the received value may be any real number. The update rule for a normal agent therefore takes the form of a function

$$x_i = F_i z_i,$$

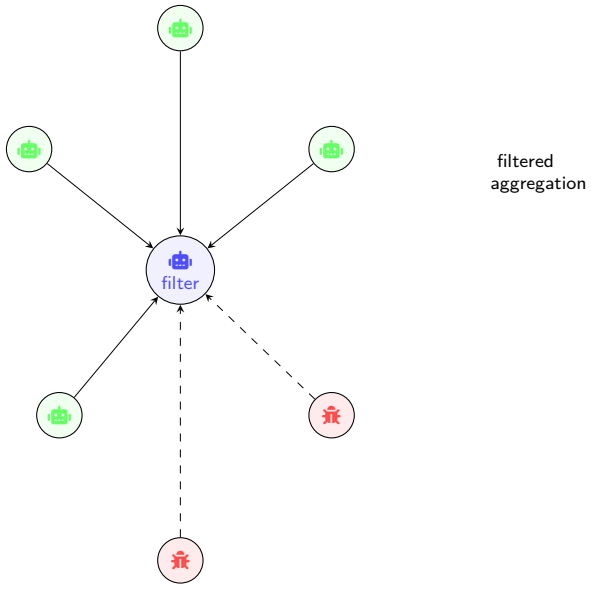


Figure 2. Local neighborhood of a normal agent performing filtered aggregation of neighbor states in the presence of Byzantine neighbors.

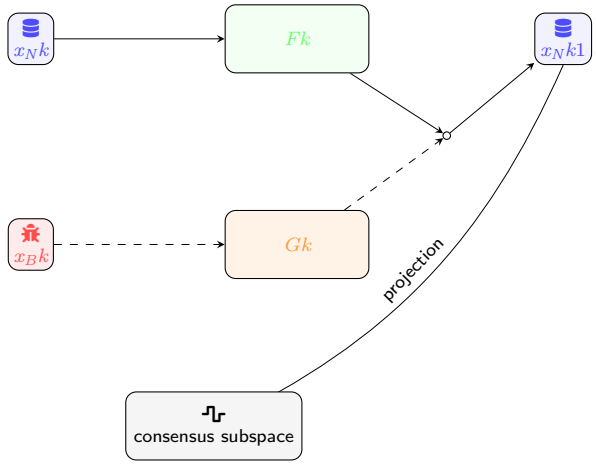


Figure 3. Block representation of normal-agent dynamics with linear operators mapping normal and Byzantine states and projection onto the consensus subspace.

where z_i is the vector of received values and may include arbitrary components due to adversaries.

A large class of resilient mechanisms relies on linear or piecewise linear operations on the received vector z_i. A typical design consists of discarding extreme values and then averaging the remaining entries using fixed coefficients (16). Such a mechanism can be represented as

$$x_i = w_i^\top \phi_i z_i,$$

where ϕ_i is a selection operator that reorders and possibly truncates the received vector, and where w_i is a fixed weight vector. For a given configuration of input values, the selection operator chooses a particular subset, and the resulting update is linear in the retained entries. The overall network transition for the normal states can be written as (17)

$$x_N = Mx x_N \;\; ux,$$

where x_N is the subvector of normal states, Mx is a matrix whose entries depend on which neighbors are discarded by the filters, and ux is a term that aggregates the influence of adversarial agents. Both Mx and ux vary with the current configuration of states because the filtering decisions depend on relative values.

The consensus problem in this context is to design update rules and impose structural conditions such that the normal agents approach a common value despite the presence of adversaries. Formally, for every initial condition of the normal states and for every sequence of adversarial actions, the aim is to achieve (18)

$$\lim_{k\to\infty} |x_i k - x_j k| = 0$$

for all normal agents i and j. The limiting value itself may depend on the initial condition and on the adversarial strategy, but it is typically constrained to lie in the convex hull of the initial

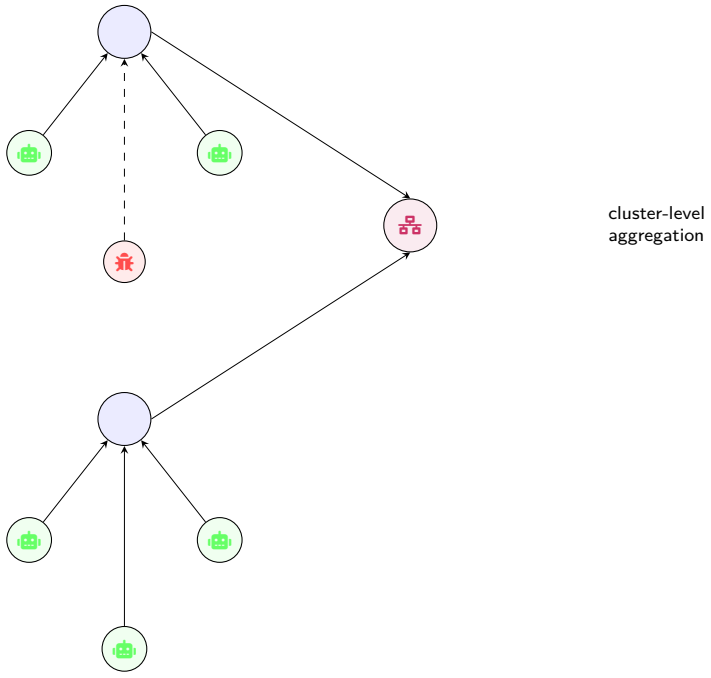


Figure 4. Hierarchical clustering of agents with resilient intra-cluster aggregation and inter-cluster coordination among protected representatives.

normal values when certain filtering schemes are employed.

Distributed optimization introduces an additional structure. Each agent i is associated with a private cost function f_i defined on a common decision space. The collection of normal agents seeks to minimize the aggregate objective (19)

$$fx = \sum_{i \in N} f_i x,$$

while agent states remain synchronized. This leads to update rules that combine consensus averaging with gradient descent. A prototypical form for a normal agent is (20)

$$x_i = v_i - \gamma g_i,$$

where v_i is an aggregated consensus variable constructed from neighbor information, g_i is a subgradient or gradient of the local function at the current estimate, and γ is a stepsize parameter. In the presence of adversaries, the consensus variable v_i is constructed through filtering operations on messages from neighbors, while the gradient is computed locally. The global dynamics of the normal agents then combine the previously described piecewise linear consensus operator with a gradient step.

The formal optimization problem of interest is to design such local update rules so that the normal agents asymptotically agree on a common state that lies near the minimizer of the aggregate objective, under bounding assumptions on gradients and structural assumptions on the graph. The notion of approximate optimality is often used because adversarial agents may bias the achievable equilibrium away from the true minimizer (21). The analysis in subsequent sections relies on linear algebraic properties of the effective consensus operators and on standard convex analysis for the gradient components.

3. Byzantine-Resilient Consensus Mechanisms

The core challenge in resilient consensus is to limit the influence of adversarial signals while preserving enough averaging among normal agents to achieve agreement. A common design pattern introduces a local filter that discards extreme neighbor values before averaging (22). Consider a normal agent i with received vector z_i. For a given integer parameter r, the agent sorts the received values and removes up to r largest and r smallest ones. The remaining entries are then combined linearly. When the number of adversarial neighbors is bounded by r, the filter ensures that all remaining values originate from normal neighbors or from the agent itself (23).

The filtered averaging step for agent i can be expressed as

$$x_i = \sum_{j \in S_i z_i} a_{ij} z_{ij},$$

where $S_i z_i$ is the retained index set after filtering and where a_{ij} are fixed nonnegative weights with unit sum over the retained set. For each configuration of received values, the set $S_i z_i$ is a subset of neighbors that depends on the ordering of the entries in z_i. The resulting update is linear in the retained entries but nonlinear in the full vector of states. From a network perspective, at each time step the state transition for the normal agents can be written as

$$x_N = Akx_N,$$

where Ak is a row stochastic matrix whose nonzero entries correspond to the dynamically selected neighbor sets for each agent (24). The entries of Ak depend on the current state configuration and on the adversarial actions through their effect on the sorting and filtering.

Despite this state dependence, one can derive convergence by showing that the sequence of matrices Ak belongs to a com-

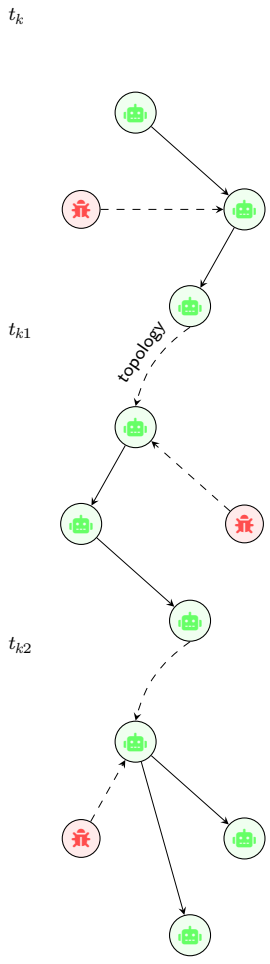


Figure 5. Snapshots of time-varying local neighborhoods showing evolution of connectivity and Byzantine placement across successive iterations.

pact set of stochastic matrices that share certain structural properties. In particular, each row of Ak assigns a fixed minimum weight to at least one normal neighbor in every sufficiently large subset (25). The interaction graph induced by the nonzero entries of these matrices must satisfy redundancy conditions that guarantee that influences from normal nodes percolate through the network even when adversarial nodes are present. When such connectivity properties hold over time, it is possible to show that the product of matrices Ak converges to a rank one matrix, leading to consensus among normal agents.

A useful viewpoint is to partition the state into normal and adversarial components (26). Let the state be expressed as

$$x = \begin{bmatrix} x_N \\ x_B \end{bmatrix},$$

where x_B collects whatever values the adversarial agents choose to adopt. The filtered consensus dynamics of the normal agents can then be written as

$$x_N = Fkx_N \;\; Gkx_B,$$

where Fk and Gk are matrices that capture, respectively, the internal averaging among normal agents and the residual influence of adversarial values after filtering (27). When the filter parameters satisfy suitable bounds relative to the number of adversarial neighbors, it is possible to design algorithms such that each row of Gk has bounded norm and each row of Fk has a dominant diagonal entry. This induces a contraction of the disagreement among normal agents, with the adversarial term acting as a bounded disturbance.

The contraction effect can be seen by defining the disagreement vector as

$$y = Hx_N,$$

where H is a matrix that projects onto the subspace orthogonal to the consensus subspace (28). For example, H may subtract the average of the normal states. The corresponding dynamics satisfy

$$y = HFkH^{\dagger}y \;\; HGkx_B,$$

where $H^{\dagger}$ is any right inverse on the relevant subspace. Under connectivity and redundancy assumptions, the induced matrix $HFkH^{\dagger}$ has a spectral radius strictly less than one in an appropriate norm, uniformly over k. This yields exponential decay of the disagreement among normal agents when the disturbance term $HGkx_B$ is bounded. If adversarial agents can choose arbitrary unbounded values, the disturbance can be unbounded, but their contribution is attenuated by the filtering and does not

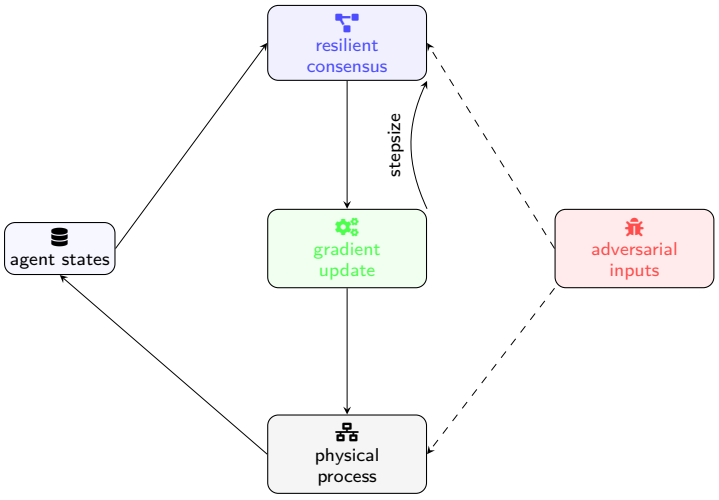


Figure 6. Closed-loop interaction between resilient consensus, gradient-based optimization, and the underlying physical process under adversarial perturbations.

Table 1. Roles in the swarm consensus and optimization architecture.

role	description	typical state	key constraint
normal agent	protocol-following node	consensus / optimizer variable	bounded update, local information
Byzantine agent	arbitrarily deviating node	arbitrary signal in $\mathbb{R}$	unconstrained behavior
environment	exogenous field or task	reference trajectory or field	indirectly observed
aggregator	local filtering entity	weighted neighbor aggregate	limited neighborhood, local memory
plant	physical process	system output	governed by dynamics

prevent the normal states from remaining within a bounded region (29).

The large-scale nature of swarm systems raises questions about how the redundancy conditions scale with the number of nodes. In sparse graphs with bounded degree, a fixed number of adversarial nodes per neighborhood represents a diminishing fraction of the overall network as the size grows. Under appropriate assumptions, the filtering rules can maintain resilience for networks of increasing size, provided that the local neighborhoods remain sufficiently large relative to the maximum number of adversarial neighbors (30). The analysis then relies on per-node conditions rather than on global fractions of adversaries, leading to resilience guarantees that scale in a structurally transparent manner.

4. Distributed Optimization with Adversarial Agents

When agents seek not only consensus but also minimization of a shared objective, the update rule is enriched by local gradient information. For each normal agent i, the local objective f_i is assumed to be convex and differentiable, with gradients that are Lipschitz continuous and bounded over the domain of interest. The aggregate objective over normal agents is (31)

$$fx = \sum_{i \in N} f_i x,$$

and the goal is to approximate a minimizer of this function while preserving resilience to adversarial agents.

A basic resilient optimization scheme combines a filtered consensus step with a gradient descent step. Let the current estimate for agent i be x_i. The agent first performs a filtered consensus operation using neighbor messages to construct an intermediate variable v_i. This variable is formed by discarding up to a prescribed number of extreme neighbor estimates and averaging the remainder, in a manner consistent with the consensus mechanisms previously described (32). The agent then computes the gradient g_i of its local function at the intermediate estimate and updates its state according to

$$x_i = v_i - \gamma g_i,$$

where γ is a stepsize that may be constant or time varying.

In vector form, stacking the normal agents' estimates into x_N, the overall dynamics can be written as

$$x_N = Ckx_N - \gamma dx_N,$$

where Ck is a stochastic matrix capturing the filtered consensus step and where dx_N is the stacked gradient vector evaluated at agent-wise intermediate points. The dependence of Ck on the state and the presence of adversarial agents leads to a piecewise

Table 2. Graph and linear operator symbols in the system model.

symbol	meaning	type	constraint
V	set of agents	finite index set	fixed cardinality n
$\mathcal{N}_i$	in-neighborhood of agent i	subset of V	bounded degree
N	set of normal agents	subset of V	$N \cap B = \emptyset$
B	set of Byzantine agents	subset of V	unknown internal logic
W	nominal weight matrix	row-stochastic matrix	sparsity matches graph
Ak	filtered weight matrix	time-varying matrix	state-dependent entries

Table 3. Linear and nonlinear components of the normal-agent update rule.

component	mapping	interpretation	effect on dynamics
Fk	$x_N \mapsto Fkx_N$	normal-to-normal mixing	contracts disagreement
Gk	$x_B \mapsto Gkx_B$	Byzantine injection	bounded disturbance
H	$x_N \mapsto Hx_N$	projection to disagreement	removes average mode
Qk	$x_N \mapsto Qkx_N$	consensus operator	ergodic product
rx_N	$x_N \mapsto rx_N$	gradient perturbation	biases equilibrium

linear operator plus a nonlinear gradient term. The analysis of convergence combines techniques from resilient consensus with those from classical distributed gradient descent.

A common assumption in convex optimization is that the aggregate objective possesses at least one minimizer and that the gradients are bounded over the domain explored by the iterates (33). Under such assumptions, one can establish that the disagreement among normal agents decreases due to the contracting effect of the filtered consensus step. Simultaneously, the average of the normal agent states approximately follows a gradient descent trajectory on the aggregate objective, perturbed by terms that arise from the discrepancies between local and average gradients and from adversarial influence. When constant stepsizes are used, the iterates can be shown to approach a neighborhood of the minimizer whose radius is proportional to the stepsize and to the strength of the adversarial disturbance (34). When diminishing stepsizes are used, the neighborhood shrinks, and convergence to a point in the set of minimizers can be obtained under additional regularity assumptions.

From a linear algebraic perspective, it is useful to decompose the dynamics into a consensus component and an optimization component. Let the consensus operator on normal agents be written as (35)

$$Qkx_N,$$

where Qk is a stochastic matrix derived from the filtering mechanism. The optimization step for the normal agents can then be expressed as a perturbation

$$x_N = Qkx_N \; rx_N,$$

where rx_N collects the gradient descent contributions. If the consensus operator is strongly ergodic in the absence of gradients, meaning that repeated application of Qk drives all normal states to a common value, then the gradients can be incorporated as slowly varying inputs to this process (36). Under small stepsizes, the gradients change slowly between consensus operations, allowing the state to track an approximate descent trajectory of the aggregate objective.

A subtle issue arises because adversarial agents may not participate in the optimization in a consistent way. They may transmit arbitrary estimates that are unrelated to any meaningful local objective. Nevertheless, their influence on the normal agents is filtered through the consensus operator and can be bounded under suitable redundancy conditions (37). This boundedness allows one to treat adversarial contributions as disturbances in the gradient dynamics. In particular, if the adversarial inputs remain bounded, they induce a bounded bias in the gradient descent trajectory of the normal agents, and approximate optimality can still be achieved.

The presence of adversaries also affects the design of stepsize schedules (38). Large stepsizes may amplify the effect of adversarial disturbances and lead to oscillatory behavior or divergence. On the other hand, excessively small stepsizes slow down convergence, which may be undesirable in time-critical applications. A compromise is to use stepsizes that decay slowly over time, balancing the need for sufficient progress early on with the need to mitigate accumulated adversarial effects (39). The functional form of the stepsize schedule can be chosen using standard conditions from stochastic approximation, adapted to account for deterministic but arbitrary disturbances.

5. Scalability and Network-Level Effects

Resilient algorithms for consensus and optimization are often developed and analyzed on fixed network sizes. Swarm systems, however, may involve large and growing numbers of agents (40). It is therefore natural to consider how the properties of

Table 4. Key design and structural parameters affecting resilience and convergence.

parameter	type	typical choice	qualitative influence
filter radius r	integer	small relative to degree	controls Byzantine isolation
stepsize γ	scalar or schedule	constant or decaying	sets bias and convergence speed
degree lower bound	graph property	moderate integer	ensures redundancy
gradient bound	analytic bound	finite constant	limits perturbation energy
spectral gap	eigenvalue gap	positive constant	determines mixing rate

Table 5. Scaling tendencies of distributed optimization with increasing swarm size.

dimension	local cost	communication	convergence behavior
small swarm	few f_i terms	low message volume	fast mixing, low latency
medium swarm	tens of f_i terms	moderate volume	mild spectral slow-down
large swarm	many f_i terms	localized traffic	slower global mixing
very large swarm	clustered f_i terms	intra-cluster dense, inter-cluster sparse	relies on hierarchy

resilient mechanisms scale with the number of nodes. A central issue is the structure of local neighborhoods. If each agent communicates with a bounded number of neighbors, then as the network grows, each agent retains a local view of a small part of the network (41). Resilience in this regime relies on ensuring that each local neighborhood retains sufficient redundancy.

Consider a family of graphs indexed by the number of nodes, where each node has at least a fixed minimum in-degree. For a given bound on the number of adversarial neighbors per node, the filtering parameters can be chosen uniformly over the family so that each normal node discards enough extreme messages to eliminate all potential adversarial contributions. The consensus operator at each time step then yields a stochastic matrix whose rows assign nonzero weights only to normal neighbors after filtering (42). Under suitable connectivity conditions, such as joint strong connectivity over bounded time windows, these matrices induce ergodic products that drive normal states to agreement. Importantly, the size of the network does not directly enter into the local update rules, which depend only on the fixed parameters of the filter and on neighbor degrees.

The rate of convergence, however, can be influenced by network size and topology (43). For example, in sparse graphs with limited connectivity, information about distant nodes must propagate through multiple hops. The spectral gap of the effective consensus matrices often decreases as the network grows, leading to slower convergence. This effect is compounded when filtering mechanisms remove a fraction of neighbor information at each step (44). From a linear systems perspective, the second largest eigenvalue modulus of the average consensus operator plays a central role in determining convergence rates. In resilient settings, this eigenvalue is influenced not only by the graph structure but also by the placement of adversarial nodes and by the filtering pattern.

Distributed optimization introduces additional scaling considerations (45). Each normal agent maintains a local estimate and performs gradient evaluations. As the network grows, the communication and computation load per agent may remain bounded if the local neighborhood size remains bounded and the complexity of local objectives does not increase. However, the speed at which information about gradients at different parts of the network is aggregated may slow down. The effect of adversarial disturbances on the overall optimization trajectory may also become more intricate in large networks, especially when adversaries are placed strategically to obstruct information flow between distant regions (46).

One approach to managing scalability is to introduce hierarchical structures. Agents can be grouped into clusters that perform resilient consensus and optimization internally. Cluster representatives then interact with each other using similar resilient mechanisms (47). This leads to a multi-level system in which resilience is enforced at each layer. The linear operators describing the dynamics become block structured, with intra-cluster and inter-cluster components. Analyses of such systems rely on understanding the spectral properties of the block matrices and on ensuring that each level satisfies appropriate redundancy and connectivity conditions (48).

Another aspect of scalability is robustness to dynamic changes in the network. In practical swarm deployments, links may appear or disappear as agents move or as communication conditions change. Adversaries may also join or leave the network over time (49). Resilient mechanisms must therefore be designed to accommodate time-varying graphs and potentially time-varying sets of adversarial agents. The linear operator viewpoint remains useful in this context. At each time step, the consensus and filtering operations yield a matrix that reflects the current connectivity and adversarial presence (50). The key requirement is that the sequence of such matrices satisfies conditions that guarantee convergence, such as joint connectivity and uniform ergodicity. When these conditions are satisfied

Table 6. Hierarchical view of layers in a scalable resilient swarm architecture.

layer	main entities	function	resilience handle
agent layer	normal and Byzantine agents	local sensing and actuation	neighbor-based filtering
cluster layer	group heads	regional aggregation	redundancy across clusters
backbone layer	representative links	long-range coordination	sparse trusted paths
plant layer	physical subsystems	task execution	indirect feedback to network

Table 7. Contrasting benign and adversarial operating regimes and associated design responses.

aspect	benign regime	adversarial regime	design response
topology	slowly varying	intermittently disrupted	robust connectivity margin
messages	unbiased noise	targeted perturbations	extreme-value filtering
gradients	consistent across agents	partially corrupted	bounded-step updates
timing	mildly asynchronous	sporadic blocking	tolerance to delays

uniformly over the network evolution, the resilient properties of the algorithms can be preserved even in large-scale and time-varying environments.

6. Numerical Illustration and Practical Considerations

Although the focus of this discussion is on modeling and analysis, it is useful to consider how the described mechanisms behave in typical scenarios. Suppose a network of agents is arranged on a ring or grid, with each agent communicating with a fixed number of neighbors (51). A subset of agents, chosen according to some pattern, acts as adversaries and transmits values designed to disrupt consensus or to bias the optimization process. Normal agents implement a filtered averaging rule with a fixed parameter that limits the number of extreme neighbors that can be discarded. They then incorporate gradient information for distributed optimization tasks (52).

The behavior of such a system can be understood qualitatively through the linear operator framework. The filtered consensus step at each time corresponds to a stochastic matrix whose nonzero entries are determined by the retained neighbor sets. Over time, the composition of these matrices governs how information about normal initial states and local gradients diffuses through the network (53). Adversarial nodes influence the process through the terms that involve their states, but that influence is constrained by the filtering and by the limited number of adversarial neighbors per normal node.

Implementation on resource-constrained agents introduces several practical considerations. First, the sorting and filtering operations require some computational effort and memory (54). However, in many swarm applications, the number of neighbors per agent is not large, and the overhead of sorting a small vector at each iteration is manageable. Second, the design of weights for the averaging step must account for potential asymmetries in the communication graph. Uniform weights over retained neighbors are simple to implement but may not exploit structure that could improve convergence rates. More sophisticated weight choices can be employed if agents have access to local degree information or other graph metrics (55).

Communication constraints also play an important role. In wireless swarms, links may be unreliable, and communication may consume significant energy. Resilient algorithms must operate under packet losses and delays (56). One strategy is to design update rules that rely only on the most recently received values from neighbors, without requiring synchronized rounds. The resulting asynchronous dynamics can still be captured by time-varying linear operators, albeit with more complex dependence on the communication pattern. The analysis then extends to products of random or deterministic matrices that reflect the combined effect of filtering, asynchrony, and adversarial behavior (57).

Another practical dimension is the selection of algorithm parameters, such as filtering thresholds and stepsizes. Conservative parameter choices may enhance resilience by reducing the influence of adversarial nodes, but they may also slow convergence or increase sensitivity to measurement noise. Conversely, aggressive parameter choices may accelerate convergence in benign conditions but compromise resilience when adversaries are present (58). Tuning such parameters in large-scale deployments can be challenging, especially when the distribution or number of adversarial nodes is not known a priori. Adaptive schemes that adjust parameters based on observed disagreement or performance metrics may offer a way to balance these trade-offs in real time.

Finally, integration with higher-level control objectives must be considered (59). In many swarm systems, consensus or distributed optimization is not an end in itself but a component of a larger control architecture. For example, agents may use consensus to agree on a reference trajectory, which is then tracked using local controllers. Resilient consensus mechanisms must therefore interface with these local controllers in a way that preserves stability and performance. The linear model of con-

Table 8. Implementation-oriented resource considerations for resilient swarm operation.

resource	per-agent load	dominant contributor	mitigation option
computation	sorting and gradients	filtering and local descent	lightweight sorting, partial updates
communication	neighbor messages	state broadcasts	sparse schedules, compression
memory	neighbor buffers	recent message history	limited window storage
energy	radio and compute	wireless interface	duty cycling, adaptive rates

sensus dynamics can be integrated into system-level analyses that account for the interaction between estimation, optimization, and control layers in the presence of adversarial behavior (60).

7. Conclusion

Large-scale swarm systems and distributed networks face challenges when some agents behave in a Byzantine manner. Classical consensus and optimization algorithms, while efficient in benign settings, are not designed to tolerate arbitrary deviations from protocol. The linear operator viewpoint on resilient consensus and distributed optimization provides a framework for understanding how local filtering, weighted averaging, and gradient steps interact in the presence of adversaries (61). By modeling the dynamics of normal agents through state-dependent stochastic matrices and disturbance terms arising from adversarial actions, one can derive conditions under which normal agents achieve agreement or approximate optimality.

The study highlights the importance of redundancy in local neighborhoods and of structural properties of the communication graph. When each normal agent has sufficiently many neighbors and the number of adversarial neighbors per agent is bounded, filtering mechanisms can effectively limit adversarial influence (62). The resulting consensus operators retain contraction properties that are essential for convergence. Combining these operators with gradient-based updates leads to distributed optimization schemes that remain functional under adversarial behavior, although with performance that depends on stepsizes, disturbance bounds, and objective properties.

Scalability considerations show that resilient mechanisms can be designed using local rules that do not explicitly depend on network size, relying instead on per-node conditions and structural graph properties (63). Hierarchical architectures, time-varying graphs, and asynchronous communication patterns can be incorporated within the same linear modeling framework. Practical deployment involves balancing computational and communication overheads with resilience and performance, as well as integrating resilient consensus and optimization modules into broader control architectures.

The linear formulation of resilient swarm consensus and distributed optimization under Byzantine adversaries offers a structured way to analyze and design algorithms for large-scale networked systems. It clarifies how structural graph conditions, local algorithm parameters, and adversarial capabilities jointly determine the behavior of the system, and it provides a basis for further refinement and specialization in application-specific contexts (64).

References

[1] J. G. McCarron, M. D. Lee, and C. Wilson, "The endothelium solves problems that endothelial cells do not know exist," *Trends in pharmacological sciences*, vol. 38, pp. 322–338, 2 2017.

[2] L. Zhang, F. Yang, and A. Z. Elsherbeni, "On the use of random variables in particle swarm optimizations: A comparative study of gaussian and uniform distributions," *Journal of Electromagnetic Waves and Applications*, vol. 23, pp. 711–721, 1 2009.

[3] G. Iacca, "Distributed optimization in wireless sensor networks: an island-model framework.," *Soft Computing*, vol. 17, pp. 2257–2277, 8 2013.

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

[5] R. Sagban, K. R. Ku-Mahamud, and M. S. A. Bakar, "Nature-inspired parameter controllers for aco-based reactive search," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 11, pp. 109–117, 9 2015.

[6] W. J. Gutjahr, "Recent trends in metaheuristics for stochastic combinatorial optimization," *Open Computer Science*, vol. 1, pp. 58–66, 1 2011.

[7] P. Chamoso and F. D. la Prieta, "Swarm-based smart city platform: A traffic application," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 4, pp. 89–98, 5 2015.

[8] H. Wu, S. Qu, D. Xu, and C. Chen, "Precise localization and formation control of swarm robots via wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2014, 11 2014.

[9] T. F. Ghanem, W. S. Elkilani, and H. M. Abdul-kader, "A hybrid approach for efficient anomaly detection using metaheuristic methods.," *Journal of advanced research*, vol. 6, pp. 609–619, 3 2014.

[10] X. Liang, N. Lin, H. Chen, and W. Liu, "Intelligent image segment for material composition detection," *MATEC Web of Conferences*, vol. 100, pp. 04026–, 3 2017.

[11] A. Gupta, "Network management: Current trends and future perspectives," *Journal of Network and Systems Management*, vol. 14, pp. 483–491, 10 2006.

[12] N. Kumari and A. S. Sandhu, "Enhanced quality of service based routing protocol using hybrid ant colony optimization and particle swarm optimization," *International Journal of Computer Applications Technology and Research*, vol. 4, pp. 618–622, 8 2015.

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

[14] P. Zhang, H. Liu, and D. Yanhui, "Dynamic bee colony algorithm based on multi-species co-evolution," *Applied Intelligence*, vol. 40, pp. 427–440, 9 2013.

[15] H. Williams and M. Bishop, "Stochastic diffusion search: A comparison of swarm intelligence parameter estimation algorithms with ransac," *Algorithms*, vol. 7, pp. 206–228, 5 2014.

[16] S. Doss and J. Jeyaraj, "Securing mobile ant agent using chinese remainder theorem," *Journal of Computer Science*, vol. 9, pp. 1211–1216, 9 2013.

[17] S. Misra and P. Agarwal, "Bio-inspired group mobility model for mobile ad hoc networks based on bird-flocking behavior," *Soft Computing*, vol. 16, pp. 437–450, 5 2011.

[18] Z. H. Ali, H. A. Ali, and M. M. Badawy, "A new proposed the internet of things (iot) virtualization framework based on sensor-as-a-service concept," *Wireless Personal Communications*, vol. 97, pp. 1419–1443, 7 2017.

[19] S.-H. Chang and S. Wan, "A novel study on ant-based clustering for paddy rice image classification," *Arabian Journal of Geosciences*, vol. 8, pp. 6305–6316, 9 2014.

[20] Y. Chen and V. K. Dubey, "Ultra-wideband source localization using a particle-swarm-optimized capon estimator from a frequency-dependent channel modeling viewpoint," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, pp. 1854–1866, 7 2005.

[21] H. Wang, Z. Cui, H. Sun, S. Rahnamayan, and X.-S. Yang, "Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism," *Soft Computing*, vol. 21, pp. 5325–5339, 3 2016.

[22] L.-Y. Chuang, S.-W. Tsai, and C.-H. Yang, "Fuzzy adaptive catfish particle swarm optimization," *Artificial Intelligence Research*, vol. 1, pp. 149–170, 9 2012.

[23] 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.

[24] Z. Li, L. He, and Y. Li, "A novel multiobjective particle swarm optimization algorithm for signed network community detection," *Applied Intelligence*, vol. 44, pp. 621–633, 10 2015.

[25] Y. Wang, R. Tao, and B.-Z. Li, "Using the multi-living agent concept to investigate complex information systems," *Science in China Series F: Information Sciences*, vol. 52, pp. 1–17, 10 2008.

[26] R. de Oliveira Albuquerque, L. J. G. Villalba, and T. hoon Kim, "Gtrust: Group extension for trust models in distributed systems," *International Journal of Distributed Sensor Networks*, vol. 10, pp. 872842–, 2 2014.

[27] A. Ali, S. Siddharth, Z. Syed, and N. El-Sheimy, "Swarm optimization-based magnetometer calibration for personal handheld devices," *Sensors*, vol. 12, pp. 12455–12472, 9 2012.

[28] W. Rehan, S. Fischer, and M. Rehan, "A critical review of surveys emphasizing on routing in wireless sensor networks—an anatomization under general survey design framework," *Sensors (Basel, Switzerland)*, vol. 17, pp. 1713–, 7 2017.

[29] U. Singhal and S. Jain, "An analysis of swarm intelligence based load balancing algorithms in a cloud computing environment," *International Journal of Hybrid Information Technology*, vol. 8, pp. 249–256, 1 2015.

[30] C. L. G. Pinzon, H. E. E. Cuchango, and G. A. Prieto, "Marco de desarrollo algorítmico de inteligencia de enjambres aplicada en almacenes," *Visión electrónica*, vol. 9, pp. 194–205, 11 2015.

[31] S. R. Chowdhury, D. Mukherjee, and H. Saha, "Fpga based maximum power point tracker of partially shaded solar photovoltaic arrays using modified adaptive perceptive particle swarm optimization," *International Journal on Smart Sensing and Intelligent Systems*, vol. 2, pp. 661–675, 1 2009.

[32] 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.

[33] C. A. C. Coello, "Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored," *Frontiers of Computer Science in China*, vol. 3, pp. 18–30, 3 2009.

[34] D. N. Miku and P. Gulia, "Improve performance of load balancing using artificial bee colony in grid computing," *International Journal of Computer Applications*, vol. 86, pp. 1–5, 1 2014.

[35] S. Wang, Y. Zhang, Y. Li, W.-J. Jia, F.-Y. Liu, M.-M. Yang, and Y. Zhang, "Single slice based detection for alzheimer's disease via wavelet entropy and multilayer perceptron trained by biogeography-based optimization," *Multimedia Tools and Applications*, vol. 77, pp. 10393–10417, 12 2016.

[36] H.-S. Park and N.-H. Tran, "Development of a biology inspired manufacturing system for machining transmission cases," *International Journal of Automotive Technology*, vol. 14, pp. 233–240, 3 2013.

[37] S. R. Katteda, C. N. Raju, and M. L. Bai, "Feature extraction for image classification and analysis with ant colony optimization using fuzzy logic approach," *Signal & Image Processing : An International Journal*, vol. 2, pp. 137–143, 12 2011.

[38] H. Hamidi and A. Kamankesh, "An approach to intelligent traffic management system using a multi-agent system," *International Journal of Intelligent Transportation Systems Research*, vol. 16, pp. 112–124, 5 2017.

[39] H. A. A. Al-Rawi and K.-L. A. Yau, "Routing in distributed cognitive radio networks: A survey," *Wireless Personal Communications*, vol. 69, pp. 1983–2020, 5 2012.

[40] A. Gorbenko and V. Y. Popov, "Programming for modular reconfigurable robots," *Programming and Computer Software*, vol. 38, pp. 13–23, 2 2012.

[41] F. Naghibi, M. R. Delavar, and B. C. Pijanowski, "Urban growth modeling using cellular automata with multi-temporal remote sensing images calibrated by the artificial bee colony optimization algorithm," *Sensors (Basel, Switzerland)*, vol. 16, pp. 2122–, 12 2016.

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

[43] J. Kennedy and R. C. Eberhart, "Icnn - particle swarm optimization," *Swarm Intelligence*, vol. 1, pp. 33–57, 8 2007.

[44] F. Miao, H.-S. Park, C. Kim, and S. Ahn, "Swarm intelligence based on modified pso algorithm for the optimization of axial-flow pump impeller," *Journal of Mechanical Science and Technology*, vol. 29, pp. 4867–4876, 12 2015.

[45] J. Jayaperiya, S. S. Kumar, S. Jayendra, and S. Maha, "Fuzzy logic ant colony based routing for mobile ad-hoc networks towards improved quality of services," *International Journal of Science and Research (IJSR)*, vol. 5, pp. 777–782, 1 2016.

[46] O. Yilmaz, "Oscillatory synchronization model of attention to moving objects," *Neural networks : the official journal of the International Neural Network Society*, vol. 29, pp. 20–36, 2 2012.

[47] K.-W. Huang, J.-L. Chen, C.-S. Yang, and C.-W. Tsai, "A memetic particle swarm optimization algorithm for solving the dna fragment assembly problem," *Neural Computing and Applications*, vol. 26, pp. 495–506, 7 2014.

[48] Z. Salim, "New time-varying fuzzy sets based on a pso midpoint of the universe of discourse," *International Journal of Automation and Computing*, vol. 13, pp. 392–400, 6 2016.

[49] A. Canovas, J. Lloret, E. M. Macías, and A. Suárez, "Web spider defense technique in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, pp. 348606–, 7 2014.

[50] V. Mangat and R. Vig, "An algorithm for mining usable rules using a holistic swarm based approach," *Journal of Computer Science*, vol. 10, pp. 585–592, 4 2014.

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

[52] H.-C. Kuo, J.-R. Chang, and C.-H. Liu, "Particle swarm optimization for global optimization problems," *Journal of Marine Science and Technology*, vol. 14, pp. 6–, 12 2006.

[53] L. Vanneschi, D. Codecasa, and G. Mauri, "A comparative study of four parallel and distributed pso methods," *New Generation Computing*, vol. 29, pp. 129–161, 5 2011.

[54] N. S. Tung, S. Chakravorty, and H. S. Bhullar, "Gravity local search inspired particle swarm algorithm for economic power dispatch planning problem in small scale system," *International Journal of Grid and Distributed Computing*, vol. 9, pp. 111–124, 5 2016.

[55] S. M. Sangve and U. V. Kulkarni, "Anomaly based improved network intrusion detection system using clustering techniques," *International Journal of Advanced Research in Computer Science*, vol. 8, pp. 808–815, 8 2017.

[56] Q. Yuan, Y. Guan, B. Hong, and X. Meng, "Multi-robot task allocation using cnp combines with neural network," *Neural Computing and Applications*, vol. 23, pp. 1909–1914, 10 2012.

[57] J.-S. Wang, S. xia Li, and J. di Song, "Cuckoo search algorithm based on repeat-cycle asymptotic self-learning and self-evolving disturbance for function optimization," *Computational intelligence and neuroscience*, vol. 2015, pp. 374873–374873, 8 2015.

[58] I. Montalvo, J. Izquierdo, R. Pérez-García, and M. Herrera, "Water distribution system computer-aided design by agent swarm optimization," *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, pp. 433–448, 3 2014.

[59] J. Cannady, "An adaptive neural swarm approach for intrusion defense in ad hoc networks," *SPIE Proceedings*, vol. 8059, pp. 190–197, 5 2011.

[60] A. Shklarsh, A. Finkelshtein, G. Ariel, O. Kalisman, C. J. Ingham, and E. Ben-Jacob, "Collective navigation of cargo-carrying swarms," *Interface focus*, vol. 2, pp. 786–798, 8 2012.

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

[62] A. Boulmakoul and M. Mandar, "Fuzzy ant colony paradigm for virtual pedestrian simulation," *The Open Operational Research Journal*, vol. 5, pp. 19–29, 7 2011.

[63] H. Kaur and S. Sharma, "Performance evaluation of re-routing based hybrid aco-pso based routing algorithm for manets," *International Journal of Computer Applications*, vol. 142, pp. 12–15, 5 2016.

[64] W. Deng, H. Chen, and H. Li, "A novel hybrid intelligence algorithm for solving combinatorial optimization problems," *Journal of Computing Science and Engineering*, vol. 8, pp. 199–206, 12 2014.