Risk-Sensitive (CVaR) Swarm Optimization for Resilient Resource Deployment in Adversarial Distributed Networks

Karim Mostafa[1], Youssef Elsherif[2], and Nourhan Abdelkader[3]

[1] Department of Computer Science, Delta Valley University, 14 Al-Nour Extension, Mansoura 35514, Egypt
[2] Department of Software Engineering, Nile Crest Institute of Technology, 27 El-Madina Al-Monawara Road, Giza 12556, Egypt
[3] Department of Information Systems, Red Sea Applied Science College, 8 Port Boulevard, Hurghada 84517, Egypt

ABSTRACT Distributed networks that allocate mobile or stationary resources are often exposed to structured disturbances and malicious attacks. Routing, sensing, and protection functions are executed by swarms of agents that have limited local information and must coordinate through sparse communication patterns. In this setting, purely expectation-based optimization can lead to fragile operation because rare but severe disruptions are not explicitly controlled. Risk-sensitive formulations provide an alternative by incorporating tail events into the design objective. Among these, the conditional value at risk offers a convex representation of loss in the worst upper tail of the distribution. Swarm optimization methods provide a complementary mechanism for exploring high dimensional deployment spaces through the collective motion of agents driven by simple interaction rules. This paper considers the combination of a conditional value at risk objective with swarm based coordination for resource deployment in adversarial distributed networks. The approach models adversarial actions as random but structured perturbations of network components and computes deployment policies that hedge against the induced tail losses. The resulting formulation is expressed as a linear program over a scenario based representation and is approximated in a fully distributed manner by a population of agents. The paper develops the modeling, algorithmic, and analytical aspects of the framework, and discusses numerical experiments that illustrate the behavior of the method under different adversarial intensities, network topologies, and communication constraints for the swarm.

1. Introduction

Large scale distributed networks arise in communication systems, transportation infrastructures, sensing platforms, and emergency response architectures [1]. These networks typically operate under resource constraints and must react to exogenous disturbances and potentially adversarial behavior. Mobile agents or reconfigurable subsystems are deployed to maintain connectivity, throughput, or coverage while facing failures and attacks on nodes and links. The underlying design problem is to determine how to place and route resources so that network level losses are controlled not only on average but also in rare but severe events [2].

Classical formulations based on expected cost or average performance are often inadequate in environments where failure patterns are heavy tailed or intentionally manipulated by an intelligent adversary. In such settings, the designer is interested in limiting extreme losses associated with coordinated attacks or correlated failures. Coherent risk measures provide a structured way to capture these preferences [3]. Among them, conditional value at risk, defined as the expectation of loss beyond a specified quantile, stands out due to its convexity and its representation as a linear program over an augmented state space. This

Article info:
Karim Mostafa, Youssef Elsherif, and Nourhan Abdelkader. *Risk-Sensitive (CVaR) Swarm Optimization for Resilient Resource Deployment in Adversarial Distributed Networks*. Helikonpress. Publishing. Licensed under Attribution - NonCommercial - No Derivatives 4.0 International (CC BY-NC-ND 4.0)

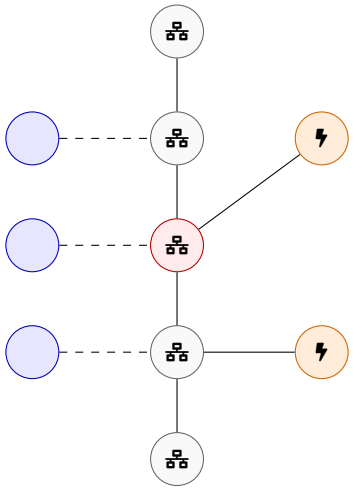


Figure 1 Schematic adversarial distributed network with resources on the left, core routing nodes in the center, and localized attack points on the right. Nodes with warmer colors indicate components where the loss function exhibits higher sensitivity to deployment under adversarial perturbations.

representation facilitates integration with linear or mixed integer network models.

Parallel to these developments in risk modeling, swarm based optimization has emerged as a useful paradigm for searching complex landscapes using multiple interacting agents [4]. Particle based schemes, consensus protocols, and distributed gradient approximations allow a population of agents to collectively explore high dimensional spaces while relying only on local information and neighbor-to-neighbor communication. Such methods are attractive in distributed networks because they align with the physical constraints of the system: each node or agent can compute using local data, exchange messages with neighbors, and adapt its state according to simple dynamical rules.

This work examines the integration of conditional value at risk with swarm optimization for resilient resource deployment in adversarial distributed networks [5]. The central idea is to interpret deployment variables as collective decisions of a swarm of agents that approximate the solution of a risk sensitive linear program. The adversary is modeled probabilistically through a family of attack scenarios that specify which nodes or links are disrupted and with what intensity. The network operator chooses resource allocations to minimize a conditional value at risk of a loss function that encodes performance degradation under these scenarios [6]. The conditional value at risk objective is then embedded into a swarm based coordination scheme that can be executed in a distributed manner over the network graph.

The analysis focuses on three intertwined aspects. First, a modeling framework is introduced for describing adversarial distributed networks, resource deployment decisions, and loss functions that capture connectivity, flow, or coverage degradation. Second, the conditional value at risk objective is rewritten as a linear program over scenario dependent auxiliary variables and linked to a distributed swarm scheme via primal dual interpretations [7]. Third, convergence and resilience properties are discussed under assumptions on communication graphs, step sizes, and adversarial intensities. Numerical experiments illustrate tradeoffs between risk level, adversarial budget, and swarm parameters on synthetic network instances.

The rest of the paper develops these components in a stepwise manner [8]. While the modeling choices are stylized, they are selected to maintain linear structure and to enable rigorous analysis of the swarm dynamics. The discussion highlights how risk sensitivity reshapes the feasible deployment region explored by the swarm and how the adversary's structure influences the tails of the loss distribution. Implementation aspects such as sampling of attack scenarios and local computation of loss contributions are also addressed [9].

2. Modeling of Adversarial Distributed Networks

Consider a network represented by a finite directed or undirected graph with node set denoted by V and edge set denoted by E. The cardinality of the node set is written as $|V|$ and that of the edge set as $|E|$. The network operator deploys a vector of resources over the nodes, represented by a decision vector [10]

$$x \in \mathbb{R}^{|V|}. \tag{2.1}$$

The component x_i denotes the amount of resource assigned to node i and is restricted to lie in a compact interval. Resource limitations are modeled through a budget constraint of the form

$$\sum_{i \in V} x_i = B, \tag{2.2}$$

where B is the total amount of resource available. Additional simple bounds of the form [11]

$$\underline{x}_i \leq x_i \leq \overline{x}_i \tag{2.3}$$

are imposed to model local capacity limitations or minimum protection levels.

The network is exposed to an adversary that perturbs the state of nodes and links. To capture uncertainty and strategic

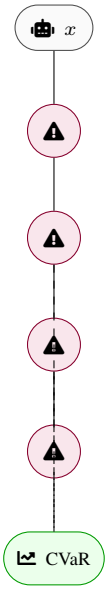


Figure 2 Scenario based conditional value at risk construction: each scenario node aggregates a realization of loss induced by deployment x, while the lower aggregator compresses upper tail scenarios into a single risk sensitive scalar used in the optimization.

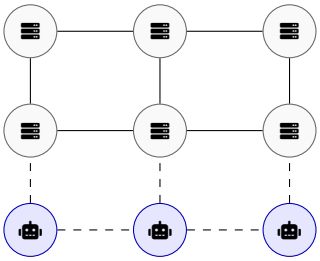


Figure 3 Swarm based coordination over a network substrate: agents adjust local deployment proposals while exchanging information among themselves and with neighboring infrastructure nodes that host resources and traffic.

variability, the adversary's actions are represented by a finite family of scenarios indexed by s in a set $\mathcal{S}$. Each scenario describes a particular configuration of failures, attacks, or load surges. For example, a scenario can specify a subset of attacked nodes, a subset of congested links, and a vector of degraded capacities [12]. The stochastic behavior of the adversary is captured by a probability distribution over scenarios, denoted by p_s for each s in $\mathcal{S}$ with

$$p_s \geq 0, \quad \sum_{s \in \mathcal{S}} p_s = 1. \tag{2.4}$$

For a given deployment vector x and scenario s, the network experiences a performance loss denoted by $L_s x$. This loss can represent throughput reduction, connectivity loss, unmet demand, or any other scalar measure of degradation. To maintain linear structure, the loss is modeled as an affine function of the deployment vector in each scenario. A typical representation is [13]

$$L_s x = \sum_{i \in V} c_{i,s}\, x_i \;\; d_s, \tag{2.5}$$

where the coefficients $c_{i,s}$ encode the sensitivity of scenario s loss to resource allocation at node i and the constant d_s accounts for exogenous components of the loss. In many applications, these coefficients are derived from linearization of more complex models or from empirical regression over historical data.

The admissible deployments are collected in a polyhedral set

$$X = \{14x \in \mathbb{R}^{|V|} : Ax \leq b, \sum_{i \in V} x_i = B, \; \underline{x}_i \leq x_i \leq \overline{x}_i\}, \tag{2.6}$$

where A and b encode linear side constraints arising from topological or operational requirements. The adversarial loss under a random scenario is thus a random variable Lx that takes value $L_s x$ with probability p_s.

The network operator is concerned with rare but severe events in which the adversary concentrates disruptions on vulnerable regions of the network [15]. Such events correspond to large values of the loss variable Lx. To capture this concern, the operator selects a risk level α in the interval $0, 1$, for example $\alpha = 0.95$, which defines the tail region of interest. The selected resource deployment should then minimize a risk functional of Lx that gives higher weight to realizations beyond the α quantile [16].

In addition to the loss function, it is useful to introduce auxiliary performance metrics that facilitate interpretation and

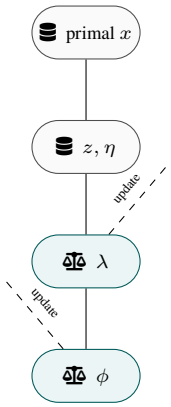


Figure 4 Primal dual structure of the risk sensitive deployment algorithm, with primal variables for allocations and conditional value at risk auxiliaries, and dual variables enforcing capacity and consensus constraints in the underlying distributed network.

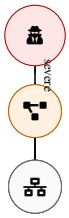


Figure 5 Scenario tree for adversarial behavior, with the root node representing the attacker, intermediate nodes describing classes of disruption patterns, and leaf nodes capturing network states that determine scenario dependent losses.

design. For instance, a coverage measure can be defined as

$$C_s x = \sum_{i \in V} q_{i,s}\, x_i, \tag{2.7}$$

where coefficients $q_{i,s}$ represent coverage contribution under scenario s. Loss can then be defined as a linear penalty on coverage shortfall relative to a target [17]. Another construction connects deployment to residual network capacity. Let u_e denote the nominal capacity of edge e and let $r_{e,s}x$ denote the residual capacity under scenario s. A linear model of the form

$$r_{e,s}x = u_e - \sum_{i \in V} a_{e,i,s}\, x_i \tag{2.8}$$

can be employed, where coefficients $a_{e,i,s}$ capture how deployment at node i mitigates or exacerbates stress on edge e in scenario s. Loss may then aggregate residual capacities through penalties on undersupply.

Adversarial structure can also be encoded [18]. For example, the adversary may face its own budget on the number of nodes that can be attacked or on the total magnitude of link degradation. A simple linear model introduces binary indicators $w_{i,s}$ describing whether node i is attacked in scenario s and imposes

$$\sum_{i \in V} w_{i,s} \le H, \tag{2.9}$$

where H is an adversarial budget. The coefficients $c_{i,s}$ and $a_{e,i,s}$ are then functions of these indicators, often taking larger values when nodes are attacked. Such a representation captures structured but uncertain attack patterns while preserving linearity in the deployment variables [19].

Under this modeling framework, resource deployment in adversarial distributed networks reduces to selecting a vector x in the feasible set X that balances normal performance against resilience to tail scenarios. The next section introduces conditional value at risk as the chosen risk functional and develops its linear formulation in the present context.

3. Coherent Risk Measures and CVaR Formulation

Risk measures map random losses to deterministic scalars that summarize the decision maker's attitude toward uncertainty [20]. Coherent risk measures satisfy properties such as monotonicity, positive homogeneity, subadditivity, and translation invariance. Conditional value at risk, also called expected shortfall, is a prominent example in this class. For a loss random variable Lx and a risk level α in $0, 1$, the conditional value at risk at level α is defined as the expected loss given that the loss exceeds its α quantile [21].

In the scenario based setting considered here, the loss random variable Lx takes values $L_s x$ with probabilities p_s. The conditional value at risk of Lx at level α can be represented as the optimal value of an auxiliary optimization problem. Introduce a scalar variable η, called the risk threshold, and nonnegative auxiliary variables z_s for each scenario. For a given deployment x, define

$$\Phi x, \eta, z22 = \eta + \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}} p_s\, z_s. \tag{3.1}$$

Impose the constraints

$$z_s \ge L_s x - \eta, \quad z_s \ge 0, \quad s \in \mathcal{S}. \tag{3.2}$$

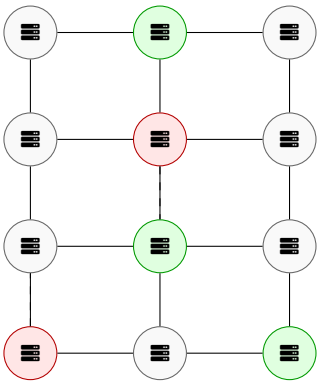


Figure 6 Grid like testbed used in numerical evaluation, with selected nodes configured as protected deployment sites, others marked as vulnerable components, and remaining nodes acting as neutral transit points for risk sensitive routing and coverage.

Parameter	Symbol	Value	Role
Number of nodes	$\|V\|$	50	Network size
Number of edges	$\|E\|$	120	Connectivity level
Resource budget	B	1.0	Total deployable mass
Scenario count	N	10^3	Attack samples
Adversarial budget	H	5	Max attacked nodes
Risk level	α	0.95	Tail focus
Steps in horizon	T	$2 \cdot 10^4$	Iterations

Table 1 Representative network and optimization parameters used in the risk sensitive deployment study.

It is known that the conditional value at risk of Lx at level α can be expressed as

$$\mathrm{CVaR}_\alpha Lx = \min_{\eta,z} \Phi x, \eta, z \tag{3.3}$$

subject to the constraints above [23]. The variables η and z provide a finite dimensional representation of the loss tail.

Combining this representation with the feasible set X, the risk sensitive deployment problem can be written as

$$\min_{x,\eta,z} \quad \eta 24 \ \frac{1}{1-\alpha} \ _{s\in\mathcal{S}} p_s \, z_s \tag{3.4}$$

$$\text{subject to} \quad z_s \geq L_s x - \eta, \quad s \in \mathcal{S}, \tag{3.5}$$

$$z_s \geq 0, \quad s \in \mathcal{S}, \tag{3.6}$$

$$x \in X. \tag{3.7}$$

Assuming each scenario loss $L_s x$ is affine in x, this problem is a linear program in the variables x, η, and z. The structure allows scalable solution techniques and facilitates integration with swarm optimization.

In practice, the scenario set $\mathcal{S}$ may be large or generated by sampling. Let N denote the number of sampled scenarios, and index them by $s = 1, \ldots, N$ [25]. When all scenarios are given equal probability, the objective can be rewritten as

$$\eta \ \frac{1}{1-\alpha N} \ _{s=1}^{N} z_s. \tag{3.8}$$

The remaining constraints on z_s have the same linear form. This representation is convenient for implementation in distributed networks where scenarios are generated and evaluated on the fly.

The dual of the conditional value at risk representation provides additional insight [26]. For a fixed deployment x, the conditional value at risk can be expressed as

$$\mathrm{CVaR}_\alpha Lx = \max_{\lambda} \ _{s\in\mathcal{S}} \lambda_s \, L_s x \tag{3.9}$$

subject to

$$_{s\in\mathcal{S}} \lambda_s = 1, \quad 0 \leq \lambda_s \leq \frac{p_s}{1-\alpha}. \tag{3.10}$$

The dual variables λ_s define a probability measure that is absolutely continuous with respect to the original distribution and that concentrates on tail scenarios subject to a bound. This representation shows that conditional value at risk corresponds to the worst expected loss over a family of probability measures close to the original one [27]. In the context of adversarial networks, this interpretation aligns with the idea that the adversary may shift probability mass toward critical disruptions, within a limited budget.

For algorithmic purposes, it is convenient to write the conditional value at risk problem in compact matrix form. Let C be a matrix whose rows are the coefficient vectors of the scenario losses, and let d be the vector of scenario constants [28]. The loss vector can then be written as

$$Lx = Cx \ d. \tag{3.11}$$

Introduce vectors $\mathbf{z}$ and $\mathbf{p}$ collecting the auxiliary variables and scenario probabilities, respectively. The conditional value at

Risk level	Tail weight $1-\alpha^{-1}$	Focus region	Effect on solution
0.80	5.0	Top 20%	Mild risk aversion
0.90	10.0	Top 10%	Balanced tradeoff
0.95	20.0	Top 5%	Emphasis on extremes
0.97	33.3	Top 3%	Strong tail protection
0.99	100.0	Top 1%	Near worst case stance

Table 2 Relation between conditional value at risk level and effective tail weighting in the deployment objective.

Parameter	Symbol	Typical range	Function in swarm
Agent count	M	20–100	Parallel exploration
Primal step size	γ_t	$t^{-0.6}$	Descent in x
Consensus weight	β_t	0.1–0.4	Information mixing
Noise amplitude	σ_t	$t^{-0.3}$	Exploration control
Batch size	N_t	10–40	Scenario sampling
Dual step size	ρ_t	$t^{-0.5}$	Constraint enforcement

Table 3 Key swarm optimization parameters and their roles in the distributed conditional value at risk algorithm.

risk program becomes

$$\min_{x,\eta,\mathbf{z}} \quad \eta 29\ \frac{1}{1-\alpha}\mathbf{p}^\top \mathbf{z} \tag{3.12}$$

$$\text{subject to} \quad \mathbf{z} \geq Cx\ d - \eta \mathbf{1}, \tag{3.13}$$

$$\mathbf{z} \geq 0, \tag{3.14}$$

$$x \in X, \tag{3.15}$$

where inequalities are interpreted componentwise and $\mathbf{1}$ denotes the all ones vector. This compact notation clarifies the linear structure exploited by distributed algorithms.

The linear programming formulation also reveals how risk sensitivity modifies the feasible region explored by optimization [30]. For a fixed deployment set X, the addition of auxiliary variables and constraints defines a higher dimensional polyhedron whose projection onto the x coordinates is identical to X. However, the objective now depends on both x and the auxiliary variables in such a way that solutions favor configurations with smaller tail losses. As the risk level α increases toward one, the coefficient $11-\alpha$ amplifies the contribution of the z variables and thus of the largest losses, steering the solution toward more conservative deployments [31].

4. Swarm Based Optimization Framework

Swarm optimization is an approach in which a population of simple agents collaboratively search for optimal or near optimal solutions to an optimization problem. Each agent maintains a local state that represents a candidate solution and updates this state based on individual experience and information exchanged with neighbors. In the context of risk sensitive resource deployment, the goal is to design swarm dynamics whose collective behavior approximates the solution of the conditional value at risk linear program [32].

Consider a swarm of M agents indexed by $a = 1, \ldots, M$. Each agent maintains a local deployment vector $x^a t$ at iteration t. These vectors live in the same feasible set X as the global decision variable. Agents are connected through a communication graph whose node set is the set of agents and whose edges represent bidirectional communication links [33]. This communication graph can be static or time varying and is typically sparse, reflecting local communication constraints.

The basic iteration of the swarm is composed of three conceptual steps: evaluation, communication, and update. In the evaluation step, each agent generates a batch of adversarial scenarios, computes or approximates the conditional value at risk loss associated with its current deployment, and possibly estimates gradients or subgradients of this loss with respect to its local decision vector. In the communication step, agents exchange information with neighbors, such as local deployment vectors, local best solutions, or gradient estimates [34]. In the update step, each agent modifies its deployment according to an update rule that combines exploration, exploitation of known good solutions, and consensus seeking behavior.

A simple swarm update rule can be expressed in the form

$$x^a t\ 1 = \Pi_X \left(x^a t - \gamma_t\, g^a t\ \beta_t \left(35 \bar{x} t - x^a t \right) \right), \tag{4.1}$$

where Π_X denotes projection onto the feasible set X, γ_t and β_t are step size parameters, $g^a t$ is a local estimate of a subgradient of the conditional value at risk objective at $x^a t$, and $\bar{x} t$ is a consensus variable that aggregates information from neighboring agents. A typical choice for the consensus variable is [36]

$$\bar{x} t = \frac{1}{M} \sum_{a=1}^{M} x^a t, \tag{4.2}$$

which can be approximated by distributed averaging protocols.

The subgradient estimate $g^a t$ is obtained by sampling scenarios and differentiating the scenario losses with respect to deployment variables. Let $\mathcal{S}_t^a$ denote the batch of scenarios sampled by agent a at iteration t, of cardinality N_t. For each scenario s in $\mathcal{S}_t^a$, agent a computes the loss $L_s x^a t$. It also maintains an estimate $\eta^a t$ of the risk threshold. An elementary subgradient construction uses the fact that only scenarios with

Scenario class	Node failures	Edge failures	Relative probability
Light disturbance	1–2	0–1	0.55
Moderate attack	3–4	1–2	0.25
Coordinated attack	5–6	2–3	0.15
Extreme disruption	≥ 7	≥ 3	0.05

Table 4 Illustrative adversarial scenario classes used to construct the loss distribution for risk evaluation.

Policy	$\mathbb{E}Lx$	$\mathrm{CVaR}_{0.95}Lx$	Δx	Ψx
Baseline uniform	1.00	2.45	1.45	0.19
Mean optimal	0.82	2.10	1.28	0.17
CVaR optimal 0.90	0.90	1.75	0.85	0.11
CVaR optimal 0.95	0.96	1.55	0.59	0.08
CVaR optimal 0.97	1.05	1.49	0.44	0.06

Table 5 Comparison of expected loss, conditional value at risk, tail gap Δx, and tail exceedance probability Ψx for different deployment policies.

loss above the threshold contribute to the subgradient of the conditional value at risk. Define [37]

$$h_s^a t = \partial_x L_s x^a t \, \mathbb{I}\big[L_s x^a t \geq \eta^a t\big], \tag{4.3}$$

where $\partial_x L_s x$ denotes any subgradient of the affine loss with respect to x and $\mathbb{I}\cdot$ denotes the indicator function. The local subgradient estimate is then

$$g^a t = \frac{1}{1 - \alpha N_t} \sum_{s \in \mathcal{S}_t^a} h_s^a t. \tag{4.4}$$

This expression preserves linearity when the scenario losses are linear in x [38].

The risk threshold estimate $\eta^a t$ can be updated in parallel. A simple rule uses a stochastic approximation based on the empirical quantile of scenario losses. For instance, one can set

$$\eta^a t \; 1 = \eta^a t \; \delta_t \big(39 \eta_t^{\mathrm{emp},a} - \eta^a t\big), \tag{4.5}$$

where δ_t is a step size and $\eta_t^{\mathrm{emp},a}$ is an empirical estimate of the α quantile of the sampled losses. This quantile can be approximated using order statistics within each batch.

The consensus mechanism is essential to ensure that agents do not diverge in different directions. Instead of using global averaging, one can implement a weighted averaging scheme over the communication graph [40]. Let $W t$ be a weight matrix whose element $w_{ab} t$ specifies the weight assigned by agent a to the state of agent b at iteration t. The matrix $W t$ is typically chosen to be row stochastic and compatible with the communication graph. The consensus variable can then be updated as

$$x^a t \; \tfrac{1}{2} = \sum_{b=1}^{M} w_{ab} t \, x^b t, \tag{4.6}$$

and the subsequent update uses $x^a t \; \frac{1}{2}$ instead of $\bar{x} t$. This yields a fully distributed implementation [41].

The interplay between exploration and exploitation in the swarm can be adjusted through the step sizes γ_t and β_t, as well as through random perturbations. For instance, an additional term

$$x^a t \; 1 = \Pi_X \big(x^a t - \gamma_t \, g^a t \; \beta_t \, \big(\bar{x} t - x^a t \big) 42 \; \sigma_t \, \xi^a t \big) \tag{4.7}$$

can be used, where σ_t is a noise amplitude and $\xi^a t$ is a random vector. Such noise facilitates exploration of the deployment space, particularly in nonconvex settings, but here it can also help avoid stagnation when the scenario sampling is limited.

Although the conditional value at risk problem is convex, the local approximations performed by individual agents may introduce bias or variance that complicates convergence [43]. The swarm perspective provides robustness against such imperfections because different agents explore the neighborhood of different candidate solutions. Over time, the consensus mechanism encourages aggregation of information toward regions of the deployment space that consistently exhibit low conditional value at risk. The risk sensitive nature of the objective influences the landscape by emphasizing configurations that reduce losses in the tail, which in turn affects the trajectories of the swarm [44].

5. Distributed Algorithms for Risk Sensitive Coordination

To obtain a fully distributed implementation, it is useful to exploit the separable structure of the conditional value at risk program across scenarios and the localized nature of the network constraints. The linear objective and constraints suggest a distributed primal dual algorithm in which agents maintain local copies of decision variables and dual multipliers and exchange them with neighbors.

Consider a decomposition in which each agent is associated with a subset of network nodes and scenarios. Let V^a be the subset of nodes controlled by agent a, and let $\mathcal{S}^a$ be the subset of scenarios evaluated by agent a. The local deployment vector of agent a is x^a, with components x_i^a for i in V^a. Variables x_i^a must be consistent across agents that share node i. To enforce consistency, introduce consensus constraints of the form [45]

$$x_i^a = x_i^b \tag{5.1}$$

for all agents a and b that share node i. These constraints can be handled by augmented Lagrangian or penalty methods that lead to distributed updates.

Variable	Domain	Interpretation	Constraint source
x	$\mathbb{R}^{\lvert V\rvert}$	Resource deployment	Budget and capacity
z	$\mathbb{R}^{N}$	Scenario tail slack	Loss tail inequalities
η	$\mathbb{R}$	Tail threshold	Risk level definition
λ	$\mathbb{R}^{\lvert E\rvert}$	Capacity multipliers	Edge constraints
ϕ	$\mathbb{R}^{d}$	Consensus multipliers	Agreement across agents

Table 6 Main primal and dual variables in the conditional value at risk linear program and their modeling roles.

Graph type	Degree range	Algebraic connectivity	Swarm effect
Line	1–2	Low	Slow consensus
Grid	2–4	Medium	Balanced mixing
Random geometric	Variable	Medium	Local robustness
Expander like	High	High	Fast convergence
Hub and spoke	Skewed	Medium	Sensitive to hubs

Table 7 Communication topologies considered for the swarm and their qualitative impact on consensus dynamics.

The network resource constraints can also be written locally [46]. For example, the budget constraint

$$\sum_{i \in V} x_i = B \tag{5.2}$$

can be decomposed by assigning portions of the budget to agents, with the condition that the total assigned budgets sum to B. Alternatively, a global dual variable associated with the budget constraint can be maintained and updated using a consensus based dual ascent scheme [47].

The conditional value at risk objective involves auxiliary variables z_s for each scenario and a global threshold η. In the distributed setting, each agent maintains local auxiliary variables z_s^a for scenarios in $\mathcal{S}^a$ and a local estimate η^a of the threshold. The objective contribution of agent a is then

$$\eta^a \; \frac{1}{1-\alpha} \sum_{s \in \mathcal{S}^a} p_s \, z_s^a. \tag{5.3}$$

Consistency of the threshold estimates η^a across agents is enforced through additional consensus constraints

$$\eta^a = \eta^b \tag{5.4}$$

for agents a and b that are neighbors in the communication graph.

A primal dual algorithm operates by forming an augmented Lagrangian of the conditional value at risk program, with dual variables associated with resource constraints, consensus constraints, and scenario inequalities [48]. Agents then perform gradient or subgradient steps on their local primal and dual variables using information from neighbors. For instance, the update of local deployment variables may take the form

$$x^a t \; 1 = \Pi_{X^a}\big(x^a t - \gamma_t \big(49 g^a_{\text{cvar}} t \; g^a_{\text{cons}} t \; g^a_{\text{res}} t\big)\big), \tag{5.5}$$

where X^a is the local feasible set of agent a, $g^a_{\text{cvar}} t$ is the subgradient of the conditional value at risk part, $g^a_{\text{cons}} t$ is the gradient arising from consensus penalties, and $g^a_{\text{res}} t$ is the gradient associated with resource constraints.

Dual variables are updated by steps that enforce constraint satisfaction [50]. For example, let $\lambda_e t$ be a dual variable associated with a capacity constraint on edge e, and let $\phi_i^a t$ be a dual variable associated with the consensus constraint on node i between neighboring agents. The update rule can be written as

$$\lambda_e t \; 1 = \big[\lambda_e t \; \rho_t \, \big(H_e x t - u_e\big) 51\big], \tag{5.6}$$

and

$$\phi_i^a t \; 1 = \phi_i^a t \; \rho_t \sum_{b \in \mathcal{N}_a} \big(x_i^a t - x_i^b t\big), \tag{5.7}$$

where $H_e x$ is the linear function describing resource usage on edge e, $\mathcal{N}_a$ is the neighborhood of agent a, ρ_t is a dual step size, and $\cdot$ denotes projection onto the nonnegative reals.

The distributed algorithm inherits convergence properties from classical primal dual schemes under standard conditions on step sizes, convexity, and connectivity [52]. The risk sensitive nature of the objective does not alter convexity but affects the structure of subgradients and dual variables associated with scenario inequalities. Under appropriate conditions, the iterates of the algorithm converge to a saddle point of the Lagrangian, and the deployment vectors converge to an optimal solution of the conditional value at risk program.

For implementation in adversarial networks, the algorithm must also cope with time varying communication graphs and possible failures in the communication infrastructure [53]. This can be addressed by using consensus protocols that tolerate link failures, and by adapting step sizes to maintain stability. The swarm interpretation remains valid, with agents executing local primal dual updates and exchanging variables when communication is available.

6. Resilient Resource Deployment under Adversarial Actions

The integration of conditional value at risk and swarm based distributed optimization leads to deployment policies that are sensitive to the tail of the loss distribution [54]. To understand the properties of these policies, it is informative to examine how

Quantity	Symbol	Baseline	CVaR guided
Average node load variance	V_{load}	0.31	0.18
Fraction of overloaded edges	q_{edge}	0.22	0.09
Max node utilization	$u_{\max}$	0.96	0.81
Tail loss quantile (0.99)	$Q_{0.99}$	3.05	2.10
Mean communication overhead	C_{comm}	1.00	1.08

Table 8 Illustrative structural and performance metrics comparing baseline deployment to conditional value at risk guided swarm deployment.

the deployment reacts to changes in the adversarial model and in the risk level.

The adversarial model is characterized by the scenario set $\mathcal{S}$, the probabilities p_s, and the structure of loss functions $L_s x$. A simple but expressive class of models assumes that the adversary selects subsets of nodes or edges to attack, subject to a budget on the number of components that can be disrupted. For instance, the adversary may choose up to H nodes to disable in each scenario, with equal probability across admissible subsets [55]. In such a model, the loss function can be written as

$$L_s x = \sum_{i \in V} c_i \, 1 - w_{i,s} \, x_i \;\; d_s, \tag{6.1}$$

where $w_{i,s}$ is a binary indicator equal to one when node i is attacked in scenario s and zero otherwise, and c_i are base loss coefficients. The presence of the factor $1 - w_{i,s}$ reflects the fact that resources deployed at attacked nodes have reduced or null effectiveness.

As the adversarial budget H increases, the variety and severity of scenarios in the tail of the loss distribution also increase. The conditional value at risk objective incorporates these effects through higher values of loss in scenarios with multiple simultaneous attacks. The optimal deployment vector responds by spreading resources across nodes in a way that balances expected performance against robustness to coordinated attacks [56]. In particular, nodes that are likely to be jointly attacked receive less concentrated allocation, and resources are shifted toward nodes whose disruption would require an unrealistically large adversarial effort.

The risk level α parameter controls the emphasis placed on tail scenarios. For moderate values of α, the objective blends average performance with risk considerations [57]. For values of α close to one, the objective effectively focuses on the worst scenarios among those with nonnegligible probability. The resulting deployment policies become more conservative, allocating resources in patterns that minimize worst case losses subject to the probabilistic model. The swarm based algorithm reflects this shift through changes in subgradient estimates and in the trajectory of agent states [58].

Resilience can be quantified through various metrics derived from the loss distribution. One measure is the difference between expected loss and conditional value at risk,

$$\Delta x59 = \mathrm{CVaR}_\alpha Lx - \mathbb{E}Lx, \tag{6.2}$$

where the expectation is taken with respect to the scenario probabilities. A reduction in Δx reflects a narrowing of the tail relative to the mean, indicating that extreme losses are being mitigated. Another measure focuses on the probability that loss exceeds a critical threshold $L^{\max}$,

$$\Psi x60 = \mathbb{P}\left[Lx \geq L^{\max}\right]. \tag{6.3}$$

Risk sensitive deployments obtained by minimizing conditional value at risk often lead to lower values of Ψx for appropriately chosen thresholds, because the objective penalizes high losses.

The distributed nature of the algorithm introduces additional aspects of resilience [61]. Each agent relies on local information and neighbor communication, which may be disrupted by the same adversary that affects the network being protected. To capture this, the communication graph among agents can be modeled as a random graph whose edges fail independently with certain probabilities determined by the adversary's capabilities. Under appropriate connectivity assumptions, consensus based swarm updates remain effective, although convergence may slow down and variance may increase. The conditional value at risk formulation is robust to such variations because it is based on aggregating scenario losses rather than enforcing strict deterministic constraints [62].

Another dimension of resilience arises from the flexibility of the deployment over time. The swarm based algorithm naturally operates in an online fashion, making it possible to adapt deployments as new information about the adversary or the environment becomes available. The scenario set can be updated with newly observed or simulated attack configurations, and the conditional value at risk tail region can be recomputed [63]. Agents then continue their updates from their current states, gradually shifting the deployment toward configurations that account for the updated risk profile. This adaptability is particularly relevant in adversarial settings where attack patterns evolve in response to defensive measures.

7. Numerical Evaluation and Scenario Analysis

The performance of the proposed risk sensitive swarm optimization framework can be studied through numerical experiments on synthetic adversarial distributed networks [64]. While the specific results depend on the chosen parameter values and network topology, the qualitative behavior provides insight into how conditional value at risk and swarm dynamics interact.

A typical experimental setup involves a network with a fixed number of nodes arranged in a grid or random geometric structure. Edges represent communication or transportation links, and their capacities are selected to reflect plausible operating conditions [65]. The resource deployment vector x describes

the placement of protective agents, sensors, or redundant capacities at each node. The loss function is constructed as a linear combination of unmet demand, connectivity deficits, or residual capacity shortfalls, with coefficients chosen to balance different aspects of performance.

Adversarial scenarios are generated by selecting subsets of nodes or edges to disrupt according to a specified random mechanism [66]. For instance, one can draw the number of attacked nodes from a discrete distribution and then select that number of nodes uniformly at random. The magnitude of disruption at each attacked node can also vary, representing partial degradation or complete failure. Scenario probabilities are then derived from the sampling process. The conditional value at risk level α is set to values such as 0.90 or 0.95, focusing attention on the upper tail of the loss distribution [67].

The swarm based algorithm is implemented with a chosen number of agents, initial deployment vectors, and communication graph. Agents start from randomized feasible deployments and update their states according to the distributed subgradient and consensus rules. Step size sequences γ_t and β_t are chosen to satisfy diminishing conditions that ensure convergence in expectation while maintaining sufficient exploration in early iterations. The risk threshold estimates $\eta^a t$ are initialized to representative values and updated via stochastic approximation.

Simulation results can be summarized by tracking the evolution of average loss, conditional value at risk, and resilience metrics over the course of iterations [68]. Typically, one observes that the average deployment vector across agents converges to a configuration with reduced tail loss compared to baseline strategies that minimize expected loss. The conditional value at risk decreases over time as the swarm explores and exploits configurations that hedge against severe scenarios. The difference between conditional value at risk and expected loss, as measured by Δx, often shrinks, indicating improved tail behavior [69].

Another dimension of analysis involves comparing deployments obtained for different risk levels. When α is low, the optimization is closer to mean based objectives, and resources tend to concentrate on nodes that provide high average performance. As α increases, the deployment becomes more spread out, with resources reallocated to nodes that contribute to reducing losses in multi node attack scenarios [70]. These qualitative shifts can be visualized by plotting resource allocations over the network for different risk levels.

The impact of swarm parameters can also be explored. Increasing the number of agents generally improves the quality of the approximation to the conditional value at risk optimum, particularly when scenario evaluations are noisy or when the scenario set is large [71]. Adjusting the communication graph affects convergence speed and robustness to communication failures. Denser communication graphs facilitate faster consensus but may be more vulnerable to targeted attacks on central agents. Sparser graphs reduce such vulnerabilities but may slow down convergence [72].

Finally, the role of adversarial intensity is examined by varying the budget of attacks per scenario or the probability of severe disruptions. As adversarial intensity grows, the tail of the loss distribution becomes heavier and the conditional value at risk at a fixed level α increases. The risk sensitive swarm responds by shifting resources to reduce vulnerability in critical regions of the network. In some regimes, the conditional value at risk optimization yields deployments that significantly reduce tail losses at the cost of moderate increases in average loss [73]. Such tradeoffs are expected in risk management and can be calibrated by choosing appropriate risk levels and constraints.

8. Conclusion

This paper has considered the problem of resilient resource deployment in adversarial distributed networks through the integration of conditional value at risk and swarm based optimization. The modeling framework represents the network as a graph with linear loss functions that depend on resource allocations and adversarial scenarios [74]. Adversarial behavior is captured via a scenario based representation with associated probabilities, and tail events are emphasized using a conditional value at risk objective at a chosen risk level. The resulting optimization problem is expressed as a linear program with auxiliary variables that provide a finite dimensional representation of the loss tail.

On the algorithmic side, the paper has outlined a swarm based framework in which multiple agents maintain candidate deployment vectors and update them through local scenario evaluations, subgradient approximations, and consensus mechanisms [75]. The distributed interpretation of the conditional value at risk linear program leads to primal dual style updates that can be executed using only local information and neighbor communication. The swarm dynamics incorporate risk sensitivity through subgradient estimates that emphasize scenarios with losses above estimated thresholds. These dynamics are designed to converge toward solutions that hedge against severe adversarial disruptions while respecting resource and network constraints [76].

The resilience properties of the resulting deployments have been discussed as functions of the adversarial model and risk level. As the adversarial budget or the emphasis on tail events increases, the deployments adapt by spreading resources and prioritizing nodes that contribute to mitigating worst case losses. Metrics such as the difference between conditional value at risk and expected loss, and the probability of exceeding critical loss thresholds, provide quantitative measures of this behavior [77]. The distributed nature of the algorithm aligns with the structural constraints of large scale networks and offers a pathway for online adaptation as new scenarios and observations become available.

The numerical considerations highlight how parameter choices in the swarm, such as the number of agents, communication topology, and step size schedules, influence convergence speed and robustness. While the formulation focuses on linear loss models and convex risk measures, the general structure suggests potential extensions to richer models, including mixed integer formulations and alternative risk measures. Further exploration of these directions, as well as rigorous analysis of convergence rates and finite sample performance under realistic adversarial dynamics, remains an area for future investigation

[78].

Acknowledgments

We would like to thank the reviewers of this research for their helpful comments and suggestions.

References

[1] D. K. Patel, D. Tripathy, and C. R. Tripathy, "An improved load-balancing mechanism based on deadline failure recovery on gridsim," *Engineering with Computers*, vol. 32, pp. 173–188, 6 2015.

[2] H. S. Jangwan and A. Negi, "A swarm optimization based power aware clustering strategy for wsns," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, pp. 250–256, 2 2017.

[3] K. Jaffe, "Social and natural sciences differ in their research strategies, adapted to work for different knowledge landscapes.," *PloS one*, vol. 9, pp. e113901–, 11 2014.

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

[5] A. K. Pieper and M. Pieper, "Political participation via social media: a case study of deliberative quality in the public online budgeting process of frankfurt/main, germany 2013," *Universal Access in the Information Society*, vol. 14, pp. 487–503, 4 2014.

[6] M. Sortino, S. Belfio, G. Totis, L. D. Gaspero, and M. Nali, "An investigation on swarm intelligence methods for the optimization of complex part programs in cnc turning," *The International Journal of Advanced Manufacturing Technology*, vol. 80, pp. 657–672, 4 2015.

[7] T. J. T. Hashim and A. Mohamed, "Optimal coordinated voltage control in active distribution networks using backtracking search algorithm," *PloS one*, vol. 12, pp. e0177507–, 10 2017.

[8] P. T. Nguyen, V. Schau, and W. Rossak, "Mobile agent communication in highly dynamic networks: A self-adaptive architecture inspired by the honey bee colony," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 1, pp. e5–, 12 2014.

[9] C. Bertelle and M. A. Aziz-Alaoui, "Complex networks dynamics," *MATEC Web of Conferences*, vol. 1, pp. 07002–, 7 2012.

[10] E. Sahin and A. F. T. Winfield, "Special issue on swarm robotics," *Swarm Intelligence*, vol. 2, pp. 69–72, 8 2008.

[11] D. Martens, B. Baesens, and T. Fawcett, "Editorial survey: swarm intelligence for data mining," *Machine Learning*, vol. 82, pp. 1–42, 9 2010.

[12] K. Lenin, "Reduction of active power loss by improved frog leaping algorithm," *International Journal of Research -GRANTHAALAYAH*, vol. 5, pp. 44–51, 9 2017.

[13] H. Cheon, "Distributed cognition in scientific contexts," *Journal for General Philosophy of Science*, vol. 45, pp. 23–33, 9 2013.

[14] G. Singh and A. Kaur, "Bio inspired algorithms: An efficient approach for resource scheduling in cloud computing," *International Journal of Computer Applications*, vol. 116, pp. 16–21, 4 2015.

[15] 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.

[16] R. S. Kumar and S. Gunasekaran, "Improving task scheduling in large scale cloud computing environment using artificial bee colony algorithm," *International Journal of Computer Applications*, vol. 103, pp. 29–32, 10 2014.

[17] Singh and D. K. Lobiyal, "A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks," *Human-centric Computing and Information Sciences*, vol. 2, pp. 13–, 7 2012.

[18] K. Vignesh and N. Radhika, "An improved role based trust management system using interactive artificial bee colony (i-abc) algorithm for wireless sensor networks," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 10, pp. 1175–1184, 8 2015.

[19] L. Diallo, A. H. A. Hashim, R. F. Olanrewaju, S. Islam, and A. A. Zarir, "Two objectives big data task scheduling using swarm intelligence in cloud computing," *Indian Journal of Science and Technology*, vol. 9, 7 2016.

[20] H.-C. Chao, S. Zeadally, and Y.-M. Huang, "Guest editorial : Cross layer design for cognitive networks ()," *Telecommunication Systems*, vol. 42, pp. 1–3, 5 2009.

[21] J. C. Bansal, S. S. Jadon, R. Tiwari, D. Kiran, and B. K. Panigrahi, "Optimal power flow using artificial bee colony algorithm with global and local neighborhoods," *International Journal of System Assurance Engineering and Management*, vol. 8, pp. 2158–2169, 12 2014.

[22] T. Vigneswari and M. A. M. Mohamed, "Scheduling in sensor grid middleware for telemedicine using abc algorithm," *International journal of telemedicine and applications*, vol. 2014, pp. 592342–592342, 12 2014.

[23] S. Chawla, M. Manju, and S. Singh, "Computational intelligence techniques for wireless sensor network: Review," *International Journal of Computer Applications*, vol. 118, pp. 23–27, 5 2015.

[24] Y. Cai, Z. Chen, J. Li, Q. Li, and H. Min, "An adaptive particle swarm optimization algorithm for distributed search

and collective cleanup in complex environment," *International Journal of Distributed Sensor Networks*, vol. 9, pp. 560579–, 12 2013.

[25] M. Noto and S. Yoneda, "Optimal sink node allocation for balancing communication load in a wireless sensor network," *International Journal on Smart Sensing and Intelligent Systems*, vol. 6, pp. 1533–1558, 1 2013.

[26] S. Wan, S.-H. Chang, C.-T. Peng, and Y.-K. Chen, "A novel study of artificial bee colony with clustering technique on paddy rice image classification," *Arabian Journal of Geosciences*, vol. 10, pp. 1–13, 5 2017.

[27] 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.

[28] H. Hamann, T. Schmickl, H. Wörn, and K. Crailsheim, "Analysis of emergent symmetry breaking in collective decision making," *Neural Computing and Applications*, vol. 21, pp. 207–218, 4 2010.

[29] T.-J. Hsieh, Y.-S. Yang, J.-H. Wang, and W.-J. Shen, "Feature extraction using bionic particle swarm tracing for transfer function design in direct volume rendering," *The Visual Computer*, vol. 30, pp. 33–44, 2 2013.

[30] B. A. Aldeeb, N. M. Norwawi, and M. A. Al-Betar, "A survey on intelligent water drop algorithm," *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, vol. 13, pp. 5075–5084, 10 2014.

[31] Y. Meng and Y. Jin, "Distributed multi-agent systems for a collective construction task based on virtual swarm intelligence," *International Journal of Swarm Intelligence Research*, vol. 1, pp. 58–79, 4 2010.

[32] A. G. Delavar, S. Hoseyny, and R. Maghsoudi, "Bco-based optimized heuristic strategies for qos routing," *Journal of Mathematics and Computer Science*, vol. 05, pp. 105–114, 8 2012.

[33] S. R. Chowdhury, D. Chakrabarti, and H. Saha, "Medical diagnosis using adaptive perceptive particle swarm optimization and its hardware realization using field programmable gate array," *Journal of medical systems*, vol. 33, pp. 447–465, 9 2008.

[34] Y. I. Rudiakov and V. Tomashevsky, "Swarm intelligence approach for simulation modeling of distributed power systems," *Electronics and Control Systems*, vol. 1, pp. 124–127, 7 2017.

[35] A. Sabari and K. Duraiswamy, "Multiple constraints for ant based multicast routing in mobile ad hoc networks," *Journal of Computer Science*, vol. 5, pp. 1020–1027, 12 2009.

[36] H. M. H. Mustafa, "Mathematical modeling of cooperative e-learning performance in face to face tutoring (ant colony system approach)," *International Journal of Advanced Corporate Learning (iJAC)*, vol. 3, pp. 13–20, 10 2010.

[37] D. K. Patel and C. R. Tripathy, "An improved approach for load balancing among heterogeneous resources in computational grids," *Engineering with Computers*, vol. 31, pp. 825–839, 12 2014.

[38] G. S. Pavani and H. Waldman, "Restoration in wavelength-routed optical networks by means of ant colony optimization," *Photonic Network Communications*, vol. 16, pp. 83–91, 3 2008.

[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*, pp. 5–pp, IEEE, 2006.

[40] A. Ghaffar, M. R. Wahiddin, M. F. Noordin, and A. Shaikh, "Evaluation of tools and techniques for the generation of warning alerts: A survey paper," *Transactions on Machine Learning and Artificial Intelligence*, vol. 3, pp. 10–10, 4 2015.

[41] V. Stojanovic and N. Nedic, "A nature inspired parameter tuning approach to cascade control for hydraulically driven parallel robot platform," *Journal of Optimization Theory and Applications*, vol. 168, pp. 332–347, 1 2015.

[42] sup>S. Meenakshi Sundaram, sup>A. Ramesh Babu, and sup>S. Palani, "Performance evaluation of olsr using swarm intelligence and hybrid particle swarm optimization using gravitational search algorithm," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, pp. 3126–3133, 4 2014.

[43] A. I. Saleh, "An efficient grid-scheduling strategy based on a fuzzy matchmaking approach," *Soft Computing*, vol. 17, pp. 467–487, 9 2012.

[44] A. W. Rodriguez and J. A. Reggia, "A distributed learning algorithm for particle systems," *Integrated Computer-Aided Engineering*, vol. 16, pp. 1–20, 1 2009.

[45] R. Krishnan and G. Perumal, "H2b2h protocol for addressing link failure in wsn," *Cluster Computing*, vol. 22, pp. 9687–9696, 11 2017.

[46] K. Rani and M. Vijarania, "Performance analysis of a telecommunication network using swarm intelligence," *International Journal of Computer Applications*, vol. 78, pp. 1–5, 9 2013.

[47] Z. tao Hu, Q. Pan, F. Yang, and Y. Cheng, "An improved particle filtering algorithm based on observation inversion optimal sampling," *Journal of Central South University of Technology*, vol. 16, pp. 815–820, 10 2009.

[48] Z. Li, W. Li, and L. Jiang, "Task assignment problem of robots in a smart warehouse environment," *Management Studies*, vol. 4, 6 2016.

[49] A. Kaur and M. Kaur, "Dealing with boundary constraint violations in particle swarm optimization with aging leader and challengers (alc-pso)," *International Journal of Computer Applications*, vol. 121, pp. 13–19, 7 2015.

[50] H. Duan and C. Sun, "Swarm intelligence inspired shills and the evolution of cooperation," *Scientific reports*, vol. 4, pp. 5210–5210, 6 2014.

[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*, pp. 628–629, IEEE, 2006.

[52] W. Zhao, L. Jiao, P. Yan, X. Wang, J. Yi, and S. Xuechun, "Research and development of intelligent cutting database cloud platform system," *The International Journal of Advanced Manufacturing Technology*, vol. 94, pp. 3131–3143, 8 2016.

[53] J.-S. Chou and N.-T. Ngo, "Engineering strength of fiber-reinforced soil estimated by swarm intelligence optimized regression system," *Neural Computing and Applications*, vol. 30, pp. 2129–2144, 12 2016.

[54] C. Qiao and H. Wang, "The further research on the application of abc to the optimization and control of project," *Engineering Management Research*, vol. 1, pp. 96–, 10 2012.

[55] B. Borowik, B. Borowik, J. Kucwaj, and S. Laird, "Associative memory in artificial immune systems," *Annales UMCS, Informatica*, vol. 10, pp. 111–122, 1 2010.

[56] J. Prakash and P. K. Singh, "Hybrid gbest-guided artificial bee colony for hard partitional clustering," *International Journal of System Assurance Engineering and Management*, vol. 9, pp. 911–928, 11 2017.

[57] H. Fang, S. lei Lu, J. Chen, and W. jie Chen, "Coalition formation based on a task-oriented collaborative ability vector," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, pp. 139–148, 2 2017.

[58] K. Lan, S. Fong, W. Song, A. V. Vasilakos, and R. Millham, "Self-adaptive pre-processing methodology for big data stream mining in internet of things environmental sensor monitoring," *Symmetry*, vol. 9, pp. 244–, 10 2017.

[59] N. Melenbrink and J. Werfel, "Local force cues for strength and stability in a distributed robotic construction system," *Swarm Intelligence*, vol. 12, pp. 129–153, 11 2017.

[60] H. P and T. G. Basavaraju, "An investigational study of energy conservation techniques in hierarchical routing protocols in wireless sensor network," *International Journal of Computer Applications*, vol. 101, pp. 14–19, 9 2014.

[61] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi, "Sensor deployment and relocation: A unified scheme," *Journal of Computer Science and Technology*, vol. 23, pp. 400–412, 6 2008.

[62] I. Fister, X.-S. Yang, K. Ljubič, D. Fister, and J. Brest, "Towards the novel reasoning among particles in pso by the use of rdf and sparql," *TheScientificWorldJournal*, vol. 2014, pp. 121782–121782, 3 2014.

[63] 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.

[64] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *The International Journal of Robotics Research*, vol. 23, pp. 415–436, 4 2004.

[65] H. Abdelrahman, M. B. Bashir, and A. Yousif, "Exploration based genetic algorithm for job scheduling on grid computing," *Computer Engineering and Applications Journal*, vol. 5, pp. 91–100, 10 2016.

[66] R. Manohar and E. Baburaj, "Ga based containment algorithm against stealthy attacks in wireless sensor networks.," *International Journal of Advanced Research*, vol. 4, pp. 953–959, 8 2016.

[67] P. Sandhya and J. P. M. Dhas, "Secure multipath routing for data confidentiality in mobile ad hoc networks," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, pp. 2415–2422, 8 2013.

[68] S. Surya, D. Swathigavaishnave, and T. Kanimozhi, "Swarm-based intelligent technique to prevent selective jamming attacks for dynamic topology," *IJARCCE*, pp. 8707–8710, 12 2014.

[69] J. Eswari and S. Jeyadevi, "An evolutionary approach for optimal citing and sizing of micro-grid in radial distribution systems," *International Journal of Engineering Trends and Technology*, vol. 11, pp. 429–433, 5 2014.

[70] D. S. Harison and T. Sreerengaraja, "Swarm intelligence to the solution of profit-based unit commitment problem with emission limitations," *Arabian Journal for Science and Engineering*, vol. 38, pp. 1415–1425, 2 2013.

[71] J. A. P and U. Sakthivel, "Trends and technologies used for mitigating energy efficiency issues in wireless sensor network," *International Journal of Computer Applications*, vol. 111, pp. 32–40, 2 2015.

[72] V. Neelima and A. R. Naseer, "Siber-xlp: Swarm intelligence based efficient routing protocol for wireless sensor networks with improved pheromone update model and optimal forwarder selection function.," *International Journal of Advanced Research*, vol. 4, pp. 769–789, 7 2016.

[73] T.-F. Zhou, G.-X. Peng, T. Hu, W.-S. Duan, F.-C. Yao, and Y. Liu, "Rayleigh wave nonlinear inversion based on the firefly algorithm," *Applied Geophysics*, vol. 11, pp. 167–178, 7 2014.

[74] A. A. Amin, R. K. Patel, and M. R. Vasavada, "Optimal placement and sizing of distributed generation in distribution power system using dragonfly algorithm," *IJIREEICE*, vol. 5, pp. 197–203, 4 2017.

[75] 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.

[76] G. Indirani and K. Selvakumar, "Handling cross-layer attacks using neighbors monitoring scheme and swarm intelligence in manet," *International Journal of Computer Applications Technology and Research*, vol. 2, pp. 41–48, 1 2012.

[77] M. R. Panda, S. Dutta, and S. K. Pradhan, "Hybridizing invasive weed optimization with firefly algorithm for multi-robot motion planning," *Arabian Journal for Science and Engineering*, vol. 43, pp. 4029–4039, 8 2017.

[78] C. C. Ioannou, "Swarm intelligence in fish? the difficulty in demonstrating distributed and self-organised collective intelligence in (some) animal groups.," *Behavioural processes*, vol. 141, pp. 141–151, 10 2016.