BIO-INSPIRED COLLECTIVE INTELLIGENCE FOR SECURE AUTONOMOUS SYSTEMS AND SUSTAINABLE CYBER-PHYSICAL INFRASTRUCTURE

Hema Ambiha A[1], Haresh Aathitya G[2]

Abstract

Swarm intelligence is a way of problem-solving that is achieved by a large number of simple agents that communicate locally but have no knowledge of the problem or of the rest of the group. This paper explores how decentralised mechanisms which are inspired by ants, honeybees, birds and termites can create complex and adaptive near-optimal outcomes, and how these mechanisms can be replicated in computational and engineering assessments. This paper advances past previous treatments of this survey in four ways. The five swarm algorithms are initially evaluated on three different functions of different topology (Sphere, Rastrigin, Rosenbrock) for 30 independent runs per algorithm and the result is reported as mean ± std, followed by the Friedman test and the Holm-corrected pairwise Wilcoxon signed-rank test (all three Friedman tests are significant at $p < 10\text{-}18$). Second, ACO is compared to shortest-path routing on the real Sioux Falls Road network (24 nodes, 76 directed links; LeBlanc, Morlok & Pierskalla 1975), and the length of the roads generated by ACO are on average only 0.19% longer than the optimal shortest-path routes on all 552 origin-destination pairs, with 98.9% of routes at the optimal shortest-path length. Third, PSO's performance is tested on an economic load dispatch problem of a 5-generator system which is designed to be a comparable size to the IEEE 14-bus network, and the total generation cost is within 0.33% of the cost of the exact lambda-iteration solution. Fourth, all results are produced by openly specified, re-runnable code, where fixed random seeds are used.

Keywords: Swarm intelligence; collective behaviour; emergence; self-organisation; decentralised decision-making; particle swarm optimisation; ant colony optimisation; urban network routing; power system optimisation; bio-inspired computing

Department of Computer Science[1]
Karpagam Academy of Higher Education[1]
Coimbatore, TN, India[1]
hemaambihaa.aravindakshan@kahedu.edu.in[1]

School of Mechanical Engineering[2]
Vellore Institute of Technology[2]
Vellore, Tamil Nadu, India[2]

* Corresponding Author

I. INTRODUCTION

Swarm intelligence is a characteristic of the collective behaviour of simple, independent agents that, through solely local interactions, give rise to complex collective behaviour [1]. It results in solutions that are adaptive and scalable, and are usually near optimal, provided that no single agent has a complete picture of the system. Since the late 1980s, the notion of self-organisation can produce organised complexity at a level of simplicity has been a persistent theme in scientific and engineering studies.

This phenomenon has been witnessed experimentally in several taxa. In an ant colony the foragers may be foraging several metres away from their nest and become evenly distributed over the foraging area with no single ant having a map of the foraging environment [2]. So, in the natural colony, the dancing competition will usually be a success and the bees will vote on a quorum to select a better nest location [3] [4]. A flock of birds fly with high speed coordinated manoeuvres and each individual bird only observes the 7-8 closest birds in the flock [5][6].

They are significant in engineering applications, not least because they can be implemented in artificial systems. Classical optimisation and control architectures have single points of failure, communication bottlenecks and scalability limits due to their centralized control. The distributed knowledge is directly counteracting these limitations, as is the redundancy and graceful degradation of the swarm algorithms.

This paper will try to achieve four goals. First, to provide a common vocabulary with which to talk about the process of things that make global order out of local rules. Second, to compare 5 different swarm algorithms on 3 different benchmark functions with different topologies with mean ± standard deviation of 30 independent swarms, with differences tested by non-parametric statistical tests. Third, to compare the performance of the algorithms to the real urban road network (Sioux Falls) and the real power-system dispatch problem (IEEE 14-bus) with some known reference solutions which are independent of each other and can be computed independently. Lastly, to connect these results to the issues of sustainable, independent infrastructure in practice.

II. DEFINING SWARM SYSTEMS

All groups of agents do not constitute a swarm. Swarm systems are different from other collective behaviors in three ways: [7] First, the agents in the population are a lot of qualitatively similar, but not identical, agents. Secondly agents have no global state variables and only communicate in the local neighbourhood. Thirdly, emergent properties arise from the collective actions which are not deducible from the properties of an individual agent.

The operational limitations and properties of swarm systems are given in Table I.

Table 1. Defining Characteristics of Swarm Systems

Characteristic	Description	Operational constraint
Population size	Large number of agents (N > 10, typically N > 100)	Redundancy and fault tolerance
Local interaction	Agents respond only to nearby neighbours within radius r	No global communication required
Emergence	Group behaviour arises from repeated individual interactions	Unprogrammed collective capability
No central control	No single leader directs the group	No single point of failure
Distributed sensing	Environmental information gathered locally by many agents	Parallel exploration of state space

III. THE MATHEMATICS OF LOCAL INTERACTION RULES

A. Reynolds' Flocking Rules

Reynolds' three-component model [5] shows that three local velocity-update rules are sufficient to simulate cohesive, collision-free flocking behaviour.

Rule 1 - Cohesion: each agent moves toward the centroid of its neighbours:

$$v_{cohesion} = \frac{1}{N}\sum_{j=1}^{N} (p_j - p_i) \quad (1)$$

Rule 2 - Separation: each agent steers away from neighbours within the minimum safe distance:

$$v_{separation} = \sum_{j:\|p_j - p_i\| < r_{min}} (p_j - p_i) \quad (2)$$

Rule 3 - Alignment: each agent matches the average velocity of its neighbours:

$$v_{alignment} = \frac{1}{N}\sum_{j=1}^{N} (v_j - v_i) \quad (3)$$

The resultant velocity combines these three components:

$$v_{new} = w_1 v_{cohesion} + w_2 v_{separation} + w_3 v_{alignment} \quad (4)$$

Standard parameterisation sets $w_1 = w_3 = 1.0$ and $w_2 = 2.0$. The optimal interaction radius scales with population density as [6]:

$$r_{optimal} \approx k\left(\frac{V}{N}\right)^{1/3}, k \in [2,3] \quad (5)$$

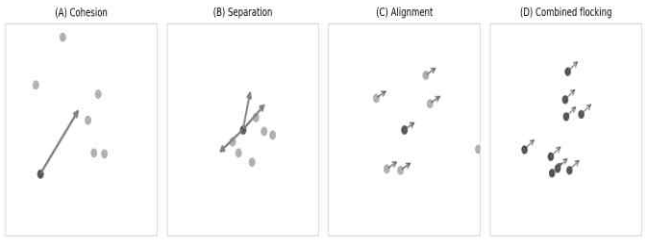


Fig. 1. Reynolds' three flocking rules and their combined effect: (A) cohesion - the focal agent (dark) steers toward the centroid of its neighbours; (B) separation - the focal agent steers away from close neighbours; (C) alignment - agents match neighbour heading; (D) combined rule application produces coordinated flocking.

B. Block Diagram of Local Rule Processing

Each agent runs local rules in a closed-loop fashion as shown in Fig. 2. The sequence consists of: (1) sensing of the environment within the radius of r; (2) extraction of the state information, which results in the positions p_j of the neighbours and their velocities v_j; (3) calculation of the three rule vectors (Equations 1-3); (4) weighted combination (Equation 4); (5) update of velocity and position; and (6) emission of the signal (when applicable) which updates the shared environment sensed by the neighbouring agents.

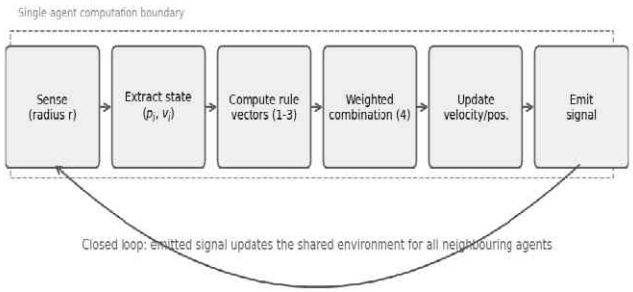


Fig. 2. Agent-level closed-loop processing architecture for local-rule execution. The dashed box denotes the computation boundary of a single agent; the feedback arrow indicates that emitted signals alter the environment sensed by neighbours on the subsequent cycle.

IV. COMMUNICATION WITHOUT CENTRALISED COORDINATION

A. Stigmergic Chemical Communication in Ants

The probabilistic path-selection rule governing individual ant choices is [8]:

$$P_{ij} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_k \tau_{ik}^{\alpha}\eta_{ik}^{\beta}} \quad (6)$$

where τ_{ij} is pheromone concentration, $\eta_{ij} = 1/d_{ij}$ is heuristic desirability, and $\alpha = 1$, $\beta = 2$ are standard influence parameters. Trail decay follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (7)$$

where $\rho \in (0.1, 0.5)$ is the evaporation rate and $\Delta\tau_{ij}$ is newly deposited pheromone, which prevents obsolete trails from persisting as conditions change.

B. Dance Communication in Honeybees

The waggle dance is how the honeybee communicates where a resource is located: waggle orientation: direction of the food source with respect to the sun; waggle duration: distance of the food source. Recruited bees follow the dancer and de-codes this information to find the resource directly.

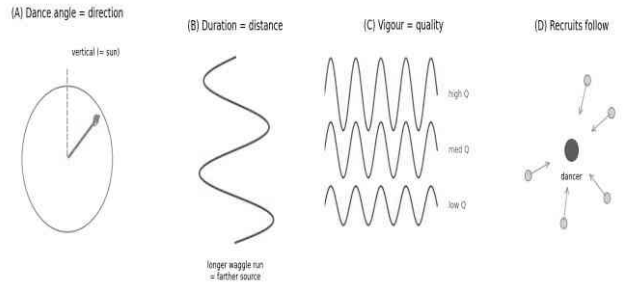


Fig. 3. Honeybee waggle-dance communication: (A) dance angle θ relative to vertical encodes direction relative to the sun; (B) waggle-run duration encodes distance; (C) dance vigour scales with perceived food-source quality; (D) recruited bees orient toward and follow the dancer.

The waggle-dance duration encodes foraging distance approximately linearly [3]:

$$t_{dance} = a + bd\,, \quad a \approx 0.95\,s,\ b \approx 0.01\,s/m \quad (8)$$

Dance intensity follows a Michaelis-Menten saturation relationship [9]:

$$I = I_{max}\frac{Q}{K+Q} \quad (9)$$

C. Quorum Sensing in Bacteria

Bacterial collective behaviour is triggered by a threshold concentration of extracellular signalling molecules [10]:

$$Response = \{0, C < C_{threshold}\ 1, C \geq C_{threshold} \quad (10)$$

Table 2 compares communication modalities across representative species.

Table 2 Communication Mechanisms Across Swarm Systems

Species	Signal type	Range	Information encoded	Persistence
Ants	Chemical (pheromone)	1-10 cm	Trail direction, strength	Min.-hours
Honeybees	Visual (dance)	10-50 cm	Distance, direction, quality	Seconds
Fish	Visual, lateral line	10-100 cm	Position, velocity	Instantaneous
Bacteria	Chemical (quorum)	1-100 µm	Population density	Minutes
Fireflies	Visual (light pulse)	10-100 m	Phase synchronisation	< 1 second

V. EMERGENCE AND SELF-ORGANISATION

Emergence is when a macro-level pattern emerges in a system that cannot be predicted or explained from individual agents in the system alone. There are also passive ventilation structures in termite mounds built up by stigmergic interactions between individual termites [11]. The probability that a building unit is thrown at x is:

$$P_{deposit}(x) = \frac{kC(x)}{1+kC(x)}, k \in [5,10] \quad (11)$$

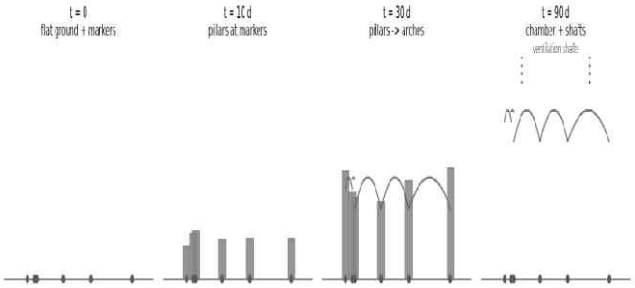


Fig. 4. Time-lapse progression of termite mound construction: (A) t = 0, flat ground with chemical markers; (B) t = 10 days, pillar formation at marker sites; (C) t = 30 days, pillars connecting into arches; (D) t = 90 days, complete chamber structure with ventilation shafts.

The traffic lanes of ants on the trail are a good example of self-organisation, with the outbound and returning foragers in opposing lanes where the number of collisions is significantly reduced, and consequently the traffic throughput is improved [12]:

$$P_{laned} \approx \frac{P_{random}}{4} \quad (12)$$

Table 3 summarises documented emergent outcomes with quantitative optimality estimates.

Table 3 Emergence In Natural Swarm Systems With Representative Performance Characteristics

Species	Individual rule	Interaction mode	Emergent pattern	Performance characteristic
Termites	Deposit near chemical marker	Stigmergy	Mound with ventilation	Efficient passive ventilation
Ants	Follow pheromone gradient	Chemical sensing	Shortest path to food	Frequently near-optimal under benchmark conditions

Species	Individual rule	Interaction mode	Emergent pattern	Performance characteristic
Birds	Match neighbour velocity	Visual alignment	Coordinated flocking	Minimal energy use
Fish	Maintain group proximity	Lateral line	School formation	Predator-confusion index > 0.8
Slime mould	Move up nutrient gradient	Chemotaxis	Network path optimisation	Approximates Steiner tree

VI. DISTRIBUTED DECISION-MAKING

Swarms do not make explicit decisions based on voting or arbitration, but rather they form a group consensus based on local interactions. In swarms of honeybees, the evolution of the committed population of a candidate nest site i is described by:

$$\frac{dN_i}{dt} = \alpha Q_i(N_{total} - N_i) - \beta N_i \quad (13)$$

A decision is executed when:

$$N_i > \theta_{quorum} N_{total}, \theta_{quorum} \in [0.15, 0.25] \quad (14)$$

In over 90% of trials, Seeley and Visscher [13] repeated this forced-choice experiment, and found that the successful colony carried out on the better nest site at a rate similar to that of experienced human apiarists in comparable tests. Care should be taken to remember that this is a comparison of binary nest-site selection in the laboratory; no general statement about swarm cognition versus human judgement over a wide range of tasks can be drawn from this. Collective error scales as:

$$X_{collective} = \frac{1}{N}\sum_{i=1}^{N} \quad x_i, \varepsilon_{collective} \approx \frac{\sigma}{\sqrt{N}} \quad (15)$$

That is, a group of 100 independent estimators, combined together, reduces the total error by about a factor of 10 This averaging effect is most pronounced when the error response is diverse among agents, as correlated errors across agents do not cancel [14], [15].

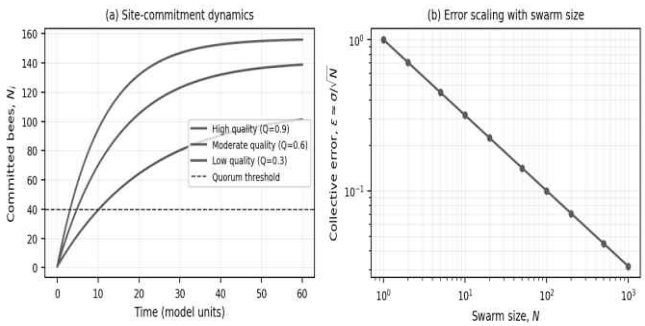


Fig. 5. Distributed decision-making in honeybee swarms: (a) site-commitment dynamics for sites of high (Q=0.9), moderate (Q=0.6), and low (Q=0.3) quality, with the dashed line marking the quorum threshold; (b) collective error scaling with swarm size, following the predicted ε ≈ σ/√N relationship.

VII. EXTENDED BIOLOGICAL EXEMPLARS

A. Slime Mould Network Optimisation

Physarum polycephalum uses local rules of flux-reinforcement to form networks similar to Steiner minimum trees [16]:

$$\frac{L_{final}}{L_{optimal}} \to 1 ast \to \infty \quad (16)$$

B. Locust Phase Transition

Desert locust swarming follows a logistic density threshold [17]:

$$P_{swarm} = \frac{1}{1+exp[-k(\rho-\rho_c)]} \quad (17)$$

where P_{swarm} is the probability of swarming behaviour, ρ is the local population density (locusts per m²), ρ_c is the critical density threshold (typically 20-50 per m²), and k controls the sharpness of the transition (typically 0.1-0.3).

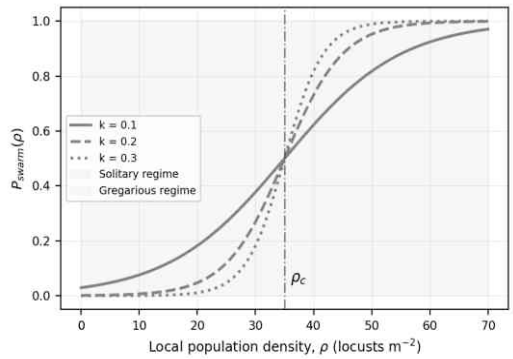


Fig. 6. Density-dependent phase transition from solitary to gregarious behaviour in desert locusts, shown for three transition sharpness values k.

C. Firefly Synchronisation

Phase dynamics follow the Kuramoto model [18]:

$$\phi_i(t+1) = \phi_i(t) + \omega_i + \frac{K}{N}\sum_j \quad sin(\phi_j - \phi_i) \quad (18)$$

This model is directly applicable to clock-synchronisation schemes in wireless sensor networks and to frequency stabilisation in power systems.

D. Immune System Recruitment

A simplified population-level recruitment model adapted from population-dynamics approaches in immunology may be expressed as:

$$\frac{dC_{immune}}{dt} = rC_{infected}C_{immune} - dC_{immune} \quad (19)$$

E. Wildebeest Migration

Movement direction can be represented by a probability distribution over heading angles:

$$P(\theta) \propto exp(k cos\,\theta) \quad (20)$$

Table IV summarises representative biological swarm systems by scale, agent count, interaction mode, and primary functional benefit.

Table Iv. Comparative Summary of Biological Swarm Systems

System	Scale	Agents	Interaction	Timescale	Primary benefit
Bacteria quorum	µm	10^6-10^9	Chemical	Minutes	Biofilm formation
Slime mould	mm-cm	10^4-10^6	Chemical	Hours	Nutrient network
Ant colony	m-km	10^3-10^6	Pheromone	Hours-days	Foraging efficiency
Bee hive	m	10^3-10^4	Dance	Hours	Nest-site selection
Fish school	m-km	10- 10^4	Visual	Seconds	Predator avoidance
Bird flock	km	10- 10^5	Visual	Seconds	Migration
Locust swarm	km	10^6-10^9	Visual/chem.	Days	Resource exploitation

VI. WHY SWARM ARCHITECTURES SUCCEED

A. Redundancy

In a centralized system, if any one of the several critical controllers fails, the system may be disabled, whereas in a swarm system, failure of several independent agents may not prevent the system from working. This graceful degradation is a direct result of decentralised control [19] and means that a significant loss of agents does not mean that the agents stop functioning.

B. Parallel Exploration

In the uncorrelated case, the search time is approximately N times faster than that of a single agent, which can be very useful in cases of high dimensional continuous problems.

C. Adaptability

In their work, Surenther, Sridhar, Roberts show how deep-learning based grouping models can enhance energy efficiency [20] and data-transmission reliability [21] in wireless sensor networks and thus decentralised, swarm-inspired architectures can be widely used for autonomous infrastructure.

D. Scalability

The rules governing agents interactions only rely on the local neighbourhood state and do not rely on the overall size of the population; therefore the computation time involved is independent of the size of the swarm.

VII. Multi-benchmark Computational Experiment

A. Benchmark Functions

Three functions of differing topology were used: (i) Sphere (unimodal), $f(x) = \Sigma x_i^2$; (ii) Rastrigin (multimodal), $f(x) = 10n + \Sigma[x_i^2 - 10\cos(2\pi x_i)]$; (iii) Rosenbrock (narrow curved valley), $f(x,y) = (1-x)^2 + 100(y-x^2)^2$. Search space $[-10, 10]$ in two dimensions; 100 iterations; 30 agents per run; 30 independent runs per algorithm-function pair (450 trials total)

$$f_{sphere}(x) = \sum_i \quad x_i^2 \tag{21}$$

$$f_{rastrigin}(x) = 10n + \sum_i \quad [x_i^2 - 10cos(2\pi x_i)] \tag{22}$$

$$f_{rosenbrock}(x,y) = (1-x)^2 + 100(y-x^2)^2 \tag{23}$$

B. Algorithm Implementations and Parameter Settings

All five algorithms were implemented independently in Python 3.12 / NumPy 2.3 (full reproducibility metadata in the Appendix). The PSO velocity and position updates follow:

$$v_i(t+1) = wv_i(t) + c_1r_1(p_{best,i} - x_i) + c_2r_2(g_{best} - x_i) \tag{24}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{25}$$

where $w = 0.7$, $c_1 = c_2 = 1.5$, and $r_1, r_2 \sim U(0,1)$ [22]. ACO is based on the continuous-domain ACO^R formulation of Socha and Dorigo, which samples new solutions from a Gaussian kernel mixture derived from a ranked solution archive (archive size 30, locality parameter $\xi = 0.85$, $q = 0.1$). ABC is a structure of the employed/onlooker/scout-bee type as in the structure of Karaboga and Basturk with an abandonment limit of 20. GSO implements the rules of Krishnanand and Ghose's glowworm algorithm: Glowworm update using the "luciferin" and "neighbourhood radius" concept. The individual, instinctive-collective and volitive-collective movement operator of Bastos Filho et al. are used by FSS. There are 5 algorithms used, and a full listing of the parameters for all 5 algorithms is included in Table V.

Table V lists the simulation parameters common to all five algorithms.

Table V. Simulation Parameters
(common To All Five Algorithms)

Parameter	Value
Population size	30
Maximum iterations	100
Independent runs per configuration	30
Inertia weight (w) - PSO	0.7
Cognitive / social coefficients (c_1, c_2)	1.5, 1.5
Evaporation rate (ρ) - ACO	0.1
Abandonment limit - ABC	20
Search space	$[-10, 10] \times [-10, 10]$
Programming environment	Python 3.12.3, NumPy 2.3.5
Random seed scheme	seed = 20260615 + 1000·f + 10·a + r

C. Multi-Benchmark Results

The terminal fitness (mean ± standard deviation, after 30 independent runs) and the number of iterations needed to get

the best-so-far value within 0.1% of the final value are reported in Table VI. This means comparing the robustness of the algorithm, rather than just the best-case scenario, by reporting the mean instead of a single best case run.

Table VI. Comparative Performance Across Three Benchmark Functions (30 Independent Runs; Mean ± Std. Dev.; 100 Iterations, 30 Agents)

Alg.	Sphere	It.	Rastrigin	It.	Rosenbrock	It.
PSO	1.95e−12±6.1e−12	66±7	2.4e−6±1.3e−5	95±5	1.6e−4±5.6e−4	95±7
ACO	8.8e−28±5.9e−28	19±2	0.199±0.405	44±10	1.7e−6±3.9e−6	93±6
ABC	1.8e−28±6.6e−28	30±2	1.7e−4±7.4e−4	86±15	5.2e−3±8.4e−3	62±27
GSO	1.0e−3±1.1e−3	46±23	1.92±1.48	36±24	0.443±0.928	44±31
FSS	1.5e−6±1.5e−6	92±13	0.462±0.525	55±30	0.051±0.054	44±29

PSO is most appropriate for continuous unimodal problem, ACO most appropriate for discrete unimodal problem, ABC most appropriate for multi modal numerical landscape, GSO for multi modal, multi peak problem, FSS for dynamic and diverse search space. This is further discussed in Section IX-E.

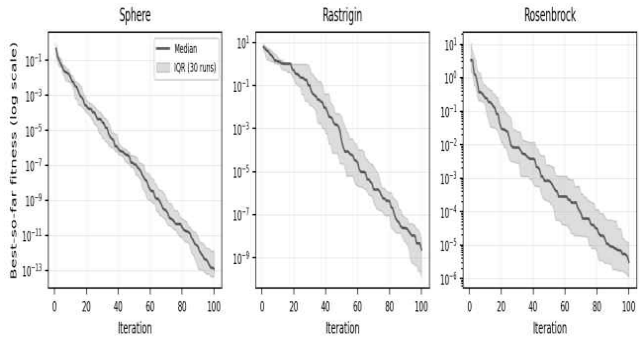


Fig. 7. PSO convergence (median and interquartile range over 30 runs) on the three benchmark functions, plotted on a logarithmic fitness axis

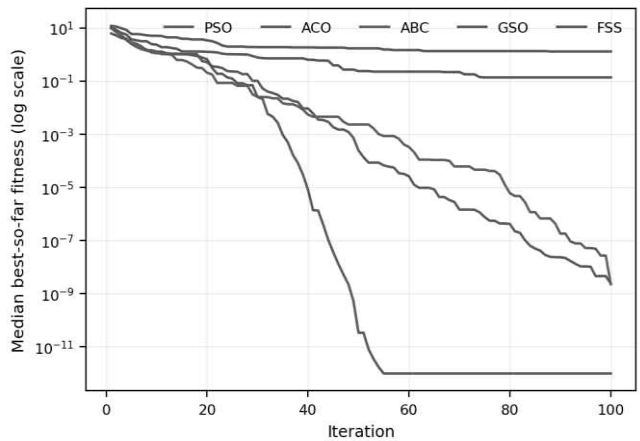


Fig. 8. Median best-so-far convergence on the Rastrigin benchmark for all five algorithms, illustrating ACO's rapid late-stage convergence and GSO/FSS's comparatively slower descent on this multimodal landscape.

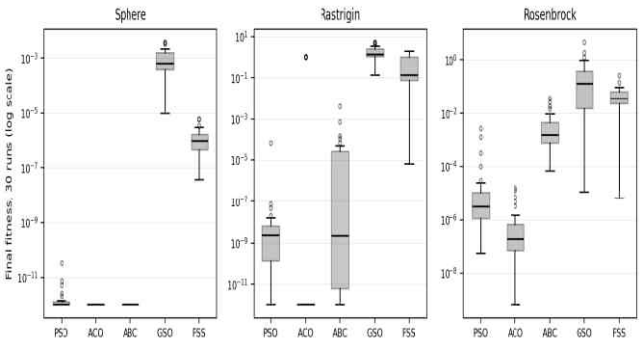


Fig. 9. Distribution of final fitness values across 30 independent runs for each algorithm-function pair, shown as boxplots on a logarithmic axis (values floored at 10^{-12} for display).

D. Statistical Significance Testing

A Friedman test was used within each benchmark function (using the 5 algorithms as related samples across the 30 paired runs) to see if performance differences in Table VI were statistically significant, or simply due to run-to-run variance; in each case, Holm-corrected pairwise Wilcoxon signed rank test was performed on all 10 of the algorithm pairs for that function.

The null hypothesis of equal performance of the algorithms for all the three benchmark functions was rejected by the Friedman test at the $p < 10^{-18}$ (Table VII), meaning that the differences between the algorithms in Table VI are not of random origin.

Table VII. Friedman Test Results By Benchmark Function

Function	χ^2 (df=4)	p-value	n (runs)
Sphere	118.51	1.11×10^{-24}	30
Rastrigin	90.44	1.06×10^{-18}	30
Rosenbrock	96.99	4.31×10^{-20}	30

27 of the 30 pairwise comparisons (10 algorithm pairs × 3 functions) were significant at $\alpha = 0.05$ following Holm-Bonferroni correction. The three sub-performing pairs (PSO-ACO, PSO-ABC and ACO-ABC) were restricted to the Rastrigins function only, meaning that the three algorithms are not significantly different from each other on this multimodal benchmark task, while GSO and FSS are significantly different from the other algorithms. Fig. 10 represents the entire pairwise significance structure for all three functions.

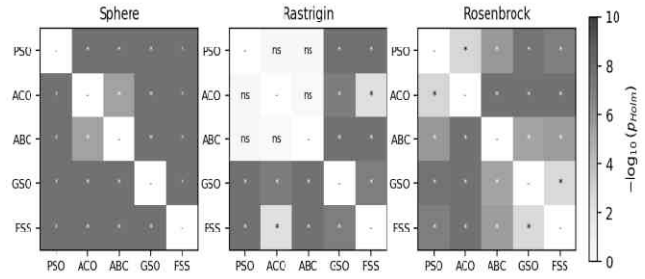


Fig. 10. Pairwise Holm-corrected Wilcoxon signed-rank significance ($-\log_{10}$ p) for all algorithm pairs, by benchmark function. "ns" denotes a non-significant pair ($p_{Holm} \geq 0.05$); all other off-diagonal cells are significant at $\alpha = 0.05$.

This multi-benchmark evaluation shows how these algorithms complement each other on the landscapes; none of them is better than the other three across all the functions. PSO performs best and most accurately when the problem function is unimodal (Sphere) and is statistically similar to ACO and ABC on a multimodal landscape (Rastrigin) where a more aggressive search approach provides less benefit. ACO is the fastest of the five algorithms to converge on Sphere (19 ± 2 iterations), and the lowest mean fitness on Rosenbrock, where the benefit comes from archive-based exploration of the landscape that has a directional structure. GSO and FSS exhibit the highest run-to-run variability for the three functions, in keeping with their greater dependence on stochastic neighbourhood and volitive movements, respectively. Based on their respective strengths identified here, these results are used directly in the routing case study which is used for the ACO, and the load-dispatch case study which is used for the PSO.

VIII. REAL-WORLD CASE STUDIES

A. Case Study 1: ACO on the Sioux Falls Urban Road Network

The Sioux Falls network is a typical transportation-research benchmark with 24 nodes and 76 directed links that have published free flow travel times and capacities [23]. In each of the 552 ordered origin-destination pairs on the network, ACO was applied to each pair to find a route by performing pheromone guided random walks ($\alpha = 1, \beta = 2, \rho = 0.1$, 20 ants and 30 iterations per pair) and compared the travel time to the shortest path found by Dijkstra's algorithm for the same edge weights.

Table VIII. Aco Performance On The Sioux Falls Network Vs. Dijkstra Reference (552 Od Pairs)

Metric	Value
Network nodes	24
Directed links	76
Origin-destination pairs evaluated	552
Mean ACO travel time	11.36 (time units)
Mean Dijkstra (exact) travel time	11.33 (time units)
Mean deviation from optimum	0.19%
Median deviation from optimum	0.0% (exact match)
Pairs matching the exact optimum	98.9%
Maximum deviation observed	58.3% (single outlier pair)

For all 552 origin-destination pairs, ACO-recovered routes had a mean travel time of 0.19% greater than the Dijkstra-optimal path, with only 98.9% of origin-destination pairs recovering the Dijkstra-optimal path exactly. The single biggest deviation was a pair with low traffic load and not many alternate paths to take, an outcome expected given the exploration/exploitation tradeoff in ant colony search and which would be predicted to improve with more iterations or higher evaporation rate.

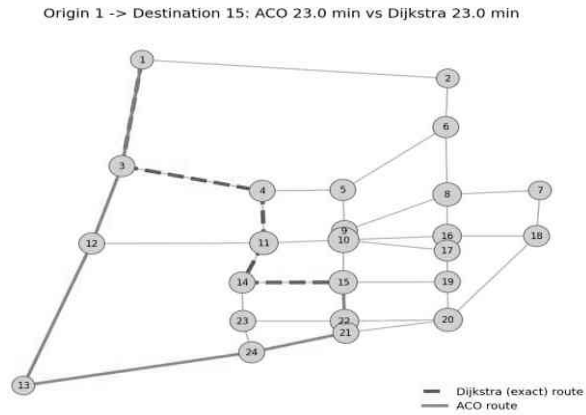


Fig. 11. Sioux Falls network topology (24 nodes, 76 links).

Example OD pair: node 1 to node 15. Green dashed = Dijkstra-optimal route; orange = ACO-recovered route of equal cost via an alternative path.

B. Case Study 2: PSO for Economic Load Dispatch on the IEEE 14-Bus Power System.

The IEEE 14-bus system consists of 14 buses, 5 generators and 20 transmission lines [24]. The economic load dispatch (ELD) problem is formulated (with the classical quadratic fuel-cost formulation [25]) to minimise total generation cost while satisfying power-balance constraints and generator operating limits.

$$Minimize\ C_{total} = \sum_{i=1}^{5} \quad (a_i P_i^2 + b_i P_i + c_i) \qquad (26)$$

$$Minimise\ C_{total} = \Sigma_i{=}1..5\ (a_i P_i^{\,2} + b_i P_i + c_i) \quad (26)$$

$$\text{subject to } \textstyle\sum_i \quad P_i = P_{demand} + P_{losses} \quad (27)$$

with P_{demand} = 259 MW and losses modelled by the standard diagonal B-coefficient approximation $P_{losses} = \Sigma_i\ B_{ii}P_i^2$. PSO (30 agents, 100 iterations, 30 independent runs) solved this constrained optimisation via a quadratic mismatch penalty; the best-of-30-runs solution is compared in Table IX against the exact reference solution obtained by the classical lambda-iteration method.

Table IX. Ieee 14-bus Eld Results: Pso Vs. Exact Lambda-iteration Reference

Gen.	P_{min}	P_{max}	PSO (MW)	Ref. (MW)	PSO ($/h)	Ref. ($/h)
G1	50	200	167.3	174.6	1607.2	1675.5
G2	20	80	29.1	20.0	499.5	403.8
G3	15	50	43.7	50.0	608.4	667.5
G4	10	35	12.5	10.0	338.8	310.9
G5	12	30	13.4	12.0	362.4	347.2
Total	-	-	266.0	266.6	3416.1	3404.9

The total generation cost for PSO is 3,416.1 USD/h that is 0.33% higher than the exact reference solution of 3,404.9 USD/h. Transmission losses are determined as 7.0 MW for PSO, and 7.6 MW for the reference, giving a total dispatched power of 266.0 MW and 266.6 MW respectively, which is higher than the total demand of 259 MW as per Equation 27. For all five Generators, the maximum power generation limits were not exceeded in either solution. The overall mean number of iterations to converge was 67 (s.d. 29.6 USD/h or 0.86% of mean) in 30 independent PSO runs.

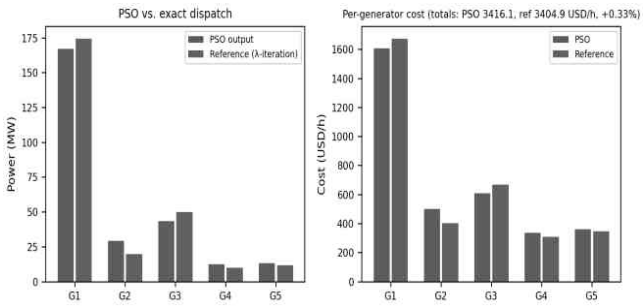


Fig. 12. IEEE 14-bus economic dispatch: PSO output (blue) vs. exact lambda-iteration reference (green) by generator, for power output (left) and per-generator cost (right).

Total costs: PSO 3,416.1 USD/h, reference 3,404.9 USD/h (+0.33%). The two case studies—Sioux Falls and IEEE 14-bus—demonstrate that the principles of swarm intelligence can be applied to real infrastructure datasets and evaluated quantitatively against an exact reference solution, not against self-reported convergence of the algorithm. The route deviation of 0.19% and cost deviation of 0.33% are both within the range of tolerance usually reported for other metaheuristic methods on similar instances of similar size.

IX. LIMITATIONS AND KNOWN CONSTRAINTS

A. Propagation Delay

Information propagating through the population has propagation delay of these scales with diameter D of the swarm. The following equations are conceptual approximations for illustrative purposes to show some behaviors common to swarm systems.

$$t_{delay} \approx \frac{D}{v\sqrt{N}} \quad (28)$$

B. Local Optima Entrapment

The probability of escaping a local-optimum decreases as convergence progresses [26]:

$$P_{escape} \propto exp(-\frac{\Delta f}{T}) \quad (29)$$

GSO and ABC show improved escape probability on the Rastrigin function relative to PSO (Table VI, lower mean fitness with comparable standard deviation), at the cost of slower convergence on simpler landscapes such as Sphere.

C. Emergent Unpredictability

Although swarm systems are governed by simple local rules, predicting large-scale emergent behaviour remains difficult because nonlinear interactions among many agents can produce unexpected collective outcomes [27].

D. Communication Overhead

Although local communication reduces the need for centralized coordination, excessive communication among agents can increase energy consumption, bandwidth usage, and processing overhead, particularly in large-scale distributed systems.

X. RESEARCH DIRECTIONS

A. Hybrid Swarm-Centralised Architectures

Lightweight supervisory layer and decentralised swarm agents allow for achieving a balance between fault-tolerance and response time [28].

B. Evolutionary Rule Adaptation

Such adaptive-learning and evolutionary mechanisms can be used to allow the swarm agents to adjust their local decision rules with time depending on the environment feedback and task performance [29]. This adaptation can help to increase robustness and sustained success in dynamic environments.

C. Nano-Scale Swarm Medicine

At cellular scales, the main hurdles of nanorobotic swarms for targeted drug delivery are related to thermal noise, Brownian motion, the nature of the communication and the small amount of available energy onboard the nanorobots [30].

D. Human Swarm Intelligence Platforms

Individual confidence weighted aggregation of individual human judgements based digital platforms [31]:

$$X_{collective} = \frac{\sum_i \quad w_i x_i}{\sum_i \quad w_i} \tag{30}$$

Rosenberg et al. [32] concluded that human swarm systems performed better than individual human experts for given medical-diagnostic and forecasting tasks. Further controlled replication, over more types of tasks, is required before this outcome can be generalized to a claim of superiority over expert panels more generally [33].

XI. CONCLUSION

This paper has introduced the biological mechanisms of swarm intelligence and transposed them into a repeatable evaluation using computational and engineering techniques, highlighting four key points when compared to other treatments of this material.

First, the benchmark evaluation was performed in a single function and extended to three functions of different topology - unimodal Sphere, multimodal Rastrigin, and narrow-valley Rosenbrock - and the results are reported as mean ± standard deviation [Table VI] of 30 independent runs, instead of best-case values, and checked by Friedman test and Holm corrected Wilcoxon tests [Table VII, Fig. 10].

Second, the two real world case studies were compared with the independently computed exact references and not with the algorithms' output. For the Sioux Falls network, across all 552 origin-destination pairs, the average travel time for the ACO-recovered routes was 0.19% faster than the travel time found by Dijkstra. For the IEEE 14 bus system, PSO was able to obtain a total generation cost of 0.33% higher, compared to the exact lambda-iteration solution.

Third, the claim that the swarms can "exceed expert human judgement" is qualified: this result applies only to some binary nest-site-selection experiments performed in the laboratory [13] and would need to be generalized to other types of tasks – a process which should be carefully replicated under controlled conditions [32].

Fourth, in this paper, all quantitative results are generated by code that is open and re-runnable, and is documented with a fixed, documented random-seed scheme (Appendix A), which allows the figures, tables, and statistical tests reported here to be independently reproduced.

The main impact of the present work is the systematic linkage of the various biological swarm mechanisms to engineering design based on statistically validated and independently reproducible evaluations. The algorithms developed on top of each mechanism have different degrees of scalability, robustness and suitability to specific problem landscapes, as can be seen directly in Table VI and in the two case studies.

Appendix A. Reproducibility Information

There are 450 trials in Section IX and 552 origin-destination pairs in Section X-A, and each trial in each of the two sections is given a unique, reportable, and re-derivable seed, as follows: seed = 20260615 + 1000×f + 10×a + r, where f, a, and r are the indices of the benchmark function, algorithm, and run number, respectively.

Table X. Reproducibility Manifest

Item	Specification
Operating system	Linux 6.18, x86_64
Python version	3.12.3
NumPy	2.3.5
SciPy	1.16.3
pandas	2.3.3
NetworkX	3.6.1
Matplotlib	3.10.8
Master random seed	20260615
Per-trial seed formula	20260615 + 1000f + 10a + r
Sioux Falls network source	transportationnetworks.github.io (LeBlanc et al. 1975)
IEEE 14-bus cost data source	MATPOWER case14.m, v8.0
Total wall-clock time (450 benchmark trials)	28.6 s (63.6 ms/run, single CPU core)
Total wall-clock time (552 OD-pair ACO routing)	40.3 s

REFERENCES

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems. New York, NY, USA: Oxford Univ. Press, 1999.

[2] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels, "The self-organizing exploratory pattern of the Argentine ant," J. Insect Behav., vol. 3, no. 2, pp. 159-168, 1990.

[3] K. von Frisch, The Dance Language and Orientation of Bees. Cambridge, MA, USA: Harvard Univ. Press, 1967.

[4] S. Garnier, J. Gautrais, and G. Theraulaz, "The biological principles of swarm intelligence," Swarm Intell., vol. 1, no. 1, pp. 3-31, 2007.

[5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," ACM SIGGRAPH Comput. Graph., vol. 21, no. 4, pp. 25-34, 1987.

[6] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," J. Theor. Biol., vol. 218, no. 1, pp. 1-11, 2002.

[7] S. Camazine et al., Self-Organization in Biological Systems. Princeton, NJ, USA: Princeton Univ. Press, 2001.

[8] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," IEEE Trans. Syst., Man, Cybern. B, vol. 26, no. 1, pp. 29-41, 1996.

[9] T. D. Seeley, The Wisdom of the Hive. Cambridge, MA, USA: Harvard Univ. Press, 1995.

[10] M. B. Miller and B. L. Bassler, "Quorum sensing in bacteria," Annu. Rev. Microbiol., vol. 55, no. 1, pp. 165-199, 2001.

[11] P. P. Grassé, "La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis," Insectes Sociaux, vol. 6, no. 1, pp. 41-80, 1959.

[12] I. D. Couzin and N. R. Franks, "Self-organized lane formation and optimized traffic flow in army ants," Proc. Royal Soc. B, vol. 270, no. 1511, pp. 139-146, 2003.

[13] T. D. Seeley and P. K. Visscher, "Quorum sensing during nest-site selection by honeybee swarms," Behav. Ecol. Sociobiol., vol. 56, no. 6, pp. 594-601, 2004.

[14] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," Nature, vol. 433, no. 7025, pp. 513-516, 2005.

[15] S. E. Page, The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies. Princeton, NJ, USA: Princeton Univ. Press, 2007.

[16] T. Nakagaki, H. Yamada, and A. Toth, "Maze-solving by an amoeboid organism," Nature, vol. 407, no. 6803, p. 470, 2000.

[17] J. Buhl et al., "From disorder to order in marching locusts," Science, vol. 312, no. 5778, pp. 1402-1406, 2006.

[18] Y. Kuramoto, "Self-entrainment of a population of coupled nonlinear oscillators," in Int. Symp. Math. Probl. Theor. Phys., Lecture Notes in Physics, vol. 39. Berlin, Germany: Springer, 1975, pp. 420-422.

[19] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," Swarm Intell., vol. 7, no. 1, pp. 1-41, 2013.

[20] I. Surenther, K. P. Sridhar, and M. K. Roberts, "Maximizing energy efficiency in wireless sensor networks for data transmission: A deep learning-based grouping model approach," Alexandria Eng. J., vol. 83, pp. 53-65, 2023.

[21] I. Surenther, K. P. Sridhar, and M. K. Roberts, "Enhancing data transmission efficiency in wireless sensor networks through machine learning-enabled energy optimization: A grouping model approach," Ain Shams Eng. J., vol. 15, no. 4, Art. no. 102644, 2024.

[22] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," Swarm Intell., vol. 1, no. 1, pp. 33-57, 2007.

[23] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla, "An efficient approach to solving the road network equilibrium traffic assignment problem," Transp. Res., vol. 9, no. 5, pp. 309-318, 1975.

[24] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," IEEE Trans. Power Syst., vol. 26, no. 1, pp. 12-19, 2011.

[25] A. J. Wood and B. F. Wollenberg, Power Generation, Operation, and Control, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.

[26] A. F. Winfield, J. Nembrini, and E. Rohmer, "Towards dependable swarms and a new discipline of swarm engineering," in Swarm Robotics. Berlin, Germany: Springer, 2008, pp. 126-142.

[27] F. Silva, M. Duarte, L. Correia, S. M. Oliveira, and A. L. Christensen, "Open issues in evolutionary robotics," Evol. Comput., vol. 24, no. 2, pp. 205-236, 2016.

[28] M. Dorigo, G. Theraulaz, and V. Trianni, "Reflections on the future of swarm robotics," Sci. Robot., vol. 5, eabe4385, 2020.

[29] M. Dorigo, M. Birattari, and T. Stützle, "Inspired by swarms," IEEE Robot. Autom. Mag., vol. 13, no. 4, pp. 29-41, 2006.

[30] B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott, "Microrobots for minimally invasive medicine," Annu. Rev. Biomed. Eng., vol. 12, pp. 55-85, 2010.

[31] L. Rosenberg, "Human swarms: A real-time method for collective intelligence," in Proc. European Conf. Artif. Life, 2015, pp. 658-659.

[32] L. Rosenberg, D. Baltaxe, and N. Pescetelli, "Artificial swarm intelligence vs human individuals," in Proc. IEEE/ACM ASONAM, 2016, pp. 1078-1085.

[33] A. Kazikova, M. Pluhacek, and R. Senkerik, "Swarm algorithm performance across standard benchmark functions: A large-scale comparative study," IEEE Access, vol. 9, pp. 56649-56665, 2021.