Applied Soft Computing Journal 194 (2026) 114914

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

SOCIAL: Social network optimization algorithm via centrality and influence-aware learning

Mehrdad Jalali [a,b,*], Binh Vu [b,c], Swati Chandna [b], Mohammad H. Nadimi-Shahraki [d]

[a] Institute of Functional Interfaces (IFG), Karlsruhe Institute of Technology (KIT), Karlsruhe, Baden-Württemberg, Germany
[b] Applied Data Science and Artificial Intelligence, SRH University Heidelberg, Heidelberg, Baden-Württemberg, Germany
[c] Faculty of Mathematics and Computer Science, University of Hagen, Hagen, Germany
[d] International Graduate School of Artificial Intelligence, National Yunlin University of Science and Technology, Douliu, Yunlin, 640301, Taiwan

HIGHLIGHTS

- SOCIAL: small-world search via centrality-weighted knowledge diffusion.
- Multi-phase scheduling shifts from exploration to elite-guided exploitation.
- Competitive or superior to 16 metaheuristics on 23 benchmarks, and six engineering tasks.
- Interpretable search through betweenness, influence propagation, and elite memory.
- Reduced parameter sensitivity, qualitative complexity, and convergence analysis.

ARTICLE INFO

Keywords:
Metaheuristic optimization
Social network algorithms
Centrality
Influence propagation
Graph-based search
Exploration–exploitation
Adaptive learning
Benchmark functions
Materials science
Cheminformatics
Materials informatics
MOF discovery
Engineering optimization

ABSTRACT

We present SOCIAL (Social Network Optimization Algorithm via Centrality and Influence-based Learning), a structure-aware metaheuristic that reframes black-box engineering optimization as social-network analysis on a small-world graph of candidate solutions. Each solution is a node, and edges specify local neighbor interactions; information flow is governed by an influence-diffusion score that combines structural centrality (betweenness/bridge potential) with relative fitness, enabling agents to preferentially learn from solutions that are both well-positioned in the network and high-quality in the search space. A time-scheduled learning policy shifts from network-driven exploration toward elite-guided exploitation, with adaptive mutation and periodic population synchronization to prevent stagnation while preserving diversity. This networked view yields interpretable search dynamics—identifying leaders, followers, and critical bridges—together with scalable communication over sparse graphs. We assess SOCIAL on 23 benchmark functions and six constrained engineering design problems (gear train, pressure vessel, welded beam, speed reducer, composite laminate, and FGM beam), demonstrating robust performance and competitive ranking-based evaluation against contemporary optimizers. SOCIAL achieves particularly strong results on multimodal, discontinuous, and constraint-dominated problems, while maintaining stability and feasibility rates. The algorithm's network-based architecture makes it particularly well-suited for materials science and cheminformatics applications, where candidate structures can be modeled as nodes in similarity graphs, enabling optimization of graph-model hyperparameters, sampling policies, and candidate exploration strategies in materials design. The key novelty is using the social-network structure itself as the learning mechanism, providing a general and explainable optimizer for engineering problems where gradients are unavailable, objectives are nonconvex or noisy, and variables are mixed discrete–continuous.

1. Introduction

Optimization underpins a wide spectrum of scientific and engineering tasks, in which the goal is to identify a best (or near-best) solution within a feasible set subject to explicit constraints. Classical techniques—Linear Programming (LP), Nonlinear Programming (NLP), and Dynamic Programming (DP)—offer strong theoretical guarantees, but typically rely on assumptions such as continuity, differentiability, or

* Corresponding author at: Applied Data Science and Artificial Intelligence, SRH University Heidelberg, Heidelberg, Baden-Württemberg, Germany.
Email address: mehrdad.jalali@srh.de (M. Jalali).

https://doi.org/10.1016/j.asoc.2026.114914
Received 23 October 2025; Received in revised form 16 February 2026; Accepted 26 February 2026
Available online 27 February 2026

convexity [1]. These assumptions often break down in high-dimensional, nonlinear, multimodal, discrete, or otherwise irregular real-world settings.

To bridge this gap, metaheuristic algorithms have emerged as problem-independent optimizers that balance exploration and exploitation without requiring gradient information. Foundational examples include Genetic Algorithms (GA) [2], Particle Swarm Optimization (PSO) [3], and Differential Evolution (DE) [4]. More recent methods—such as the Grey Wolf Optimizer (GWO) [5], Harris Hawks Optimization (HHO) [6], and the Walrus Optimizer (WO) [7]—aim to accelerate convergence while maintaining diversity on complex landscapes. Nevertheless, the No Free Lunch (NFL) theorem formalizes a key limitation: no single optimizer dominates across all problem classes [8]. Designing algorithms with stronger generalization, adaptivity, interpretability, and scalability, therefore, remains an active research frontier.

Most existing metaheuristics draw inspiration from biological or physical processes and treat agent interactions in relatively unstructured ways, paying limited attention to the *relational* patterns that drive influence and information flow in social systems. Many population-based methods (e.g., PSO, DE) rely on global-best attraction or uniform population-wide interactions, which can lead to premature convergence on multimodal landscapes. This paper introduces **SOCIAL** (Social Network Optimization Algorithm via Centrality and Influence-based Learning), a structure-aware metaheuristic that explicitly models candidate solutions as nodes in a social graph with local neighbor interactions. Unlike PSO (which uses global-best attraction) or DE (which uses population-wide mutation), SOCIAL employs *delayed consensus* through local neighbor influence, where agents primarily learn from their network neighbors rather than immediately converging to a global best. As illustrated in Fig. 1, SOCIAL leverages a small-world topology to balance local cohesion and global reach, transforming the traditional scattered-agent search into a networked system in which agents exchange information through influence propagation weighted by network-theoretic centrality and fitness. This yields an adaptive exploration–exploitation mechanism in which highly informative (centrally positioned and fit) agents guide the search, yet structural shortcuts preserve the ability to discover new basins.

SOCIAL combines (i) centrality-driven guidance and influence-aware decision-making with (ii) elite memory, dynamic mutation, and (iii) a multi-phase learning schedule that transitions from network-driven exploration to elite-guided exploitation. Periodic population synchronization mitigates stagnation, sustains diversity, and stabilizes convergence. Beyond performance, SOCIAL emphasizes *explainable search dynamics*: the flow of information and the emergence of leaders can be traced through interpretable graph measures.

What SOCIAL is not: SOCIAL is not a variant of PSO, DE, or GWO, nor is it a hybrid method combining these approaches. Unlike PSO, SOCIAL does not use global-best attraction or velocity updates. Unlike DE, SOCIAL does not employ population-wide mutation or crossover operations. Unlike GWO, SOCIAL does not use hierarchical leader–follower dynamics. Instead, SOCIAL's core mechanism is purely social-network-based: agents learn from local neighbors weighted by centrality and fitness, creating delayed consensus that prevents premature convergence while maintaining exploration capability.

We provide detailed computational complexity analysis and empirical runtime benchmarks, and conduct comprehensive sensitivity studies to clarify the role and interpretability of key hyperparameters. Extensive experiments on 23 benchmark functions (including the CEC 2021 suite) and six real-world engineering design problems demonstrate robust performance and competitive ranking-based evaluation against 16 state-of-the-art metaheuristics. SOCIAL achieves particularly strong results on multimodal, discontinuous, and constraint-dominated problems, while maintaining stability and feasibility rates. Collectively, these results support SOCIAL as a scalable, interpretable, and structure-aware alternative to existing stochastic optimization approaches for complex, multimodal, and constraint-sensitive problems.

Contributions. The main contributions of this work are:

- A social-graph-based optimization framework that integrates centrality-driven learning and influence propagation to realize *knowledge diffusion* during search.
- A multi-phase scheme that adaptively balances global exploration and local exploitation via elite memory, dynamic mutation, and population synchronization.
- A comprehensive empirical study on 23 benchmarks (including CEC 2021) and six engineering problems, alongside qualitative complexity/convergence analysis and hyperparameter sensitivity, demonstrating robustness, scalability, and explainability.

Organization. Section 2 reviews related work in metaheuristic optimization. Section 3 details the SOCIAL model and algorithmic flow. Section 4 presents the experimental setup and results. Section 4.8 concludes and outlines future directions.

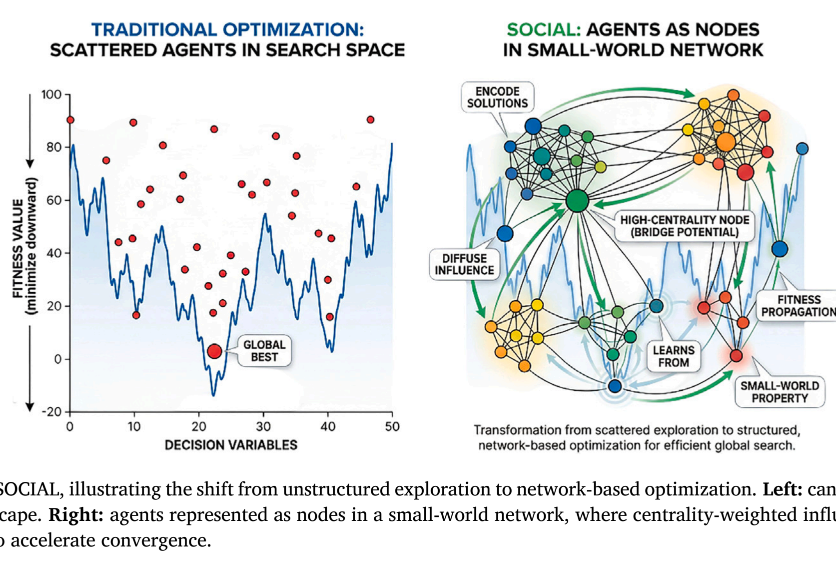


Fig. 1. Conceptual overview of SOCIAL, illustrating the shift from unstructured exploration to network-based optimization. **Left:** candidate solutions (agents) scattered over a multimodal fitness landscape. **Right:** agents represented as nodes in a small-world network, where centrality-weighted influence diffusion and elite guidance propagate fitness information to accelerate convergence.

2. Related work

The increasing complexity of real-world optimization problems, characterized by high dimensionality, nonlinearity, and multimodality, has rendered traditional optimization methods—such as Linear Programming, Nonlinear Programming, and Dynamic Programming—less effective due to their reliance on restrictive assumptions like continuity, differentiability, or convexity. Metaheuristic algorithms, which employ adaptive strategies to balance exploration and exploitation, have emerged as powerful alternatives for addressing these challenges. This literature review synthesizes recent advancements in nature-inspired metaheuristic algorithms, with a focus on their relevance to the proposed SOCIAL (Social Network Optimization Algorithm via Centrality and Influence-based Learning), which leverages social network dynamics to optimize complex problem landscapes.

2.1. Overview of metaheuristic algorithms

Metaheuristic algorithms are approximation techniques for identifying near-optimal solutions to optimization problems that challenge existing methods due to computational complexity or local optima citephan2024walrus, yang2010nature. In contrast to problem-specific heuristic algorithms, metaheuristics provide universal frameworks that utilize natural or social phenomena—such as swarm intelligence, evolutionary processes, or network dynamics—to efficiently explore solution spaces. The No Free Lunch theorem emphasizes that no single optimizer is generally superior, prompting the development of several metaheuristic techniques tailored to specific problem attributes.

2.2. Nature-inspired metaheuristic algorithms

Recent developments in metaheuristic algorithms have been inspired by various biological, ecological, and social activities, as noted by Han et al. [7]. These algorithms are assessed using benchmark suites such as CEC 2021 and practical engineering challenges, showcasing their capacity to manage intricate optimization tasks. Table 1 delineates essential metaheuristic algorithms, encompassing those contrasted with SOCIAL, their sources of inspiration, and their respective contributions.

2.2.1. Swarm intelligence-based algorithms

Algorithms that use a swarm intelligence model their actions after those of distributed systems, such as anthills or ant colonies. According to Kennedy (1995), Particle Swarm Optimization (PSO) mimics the way birds swarm by adjusting particle positions toward both individual and collective optimal solutions. Using pheromone trails, Ant Colony Optimization (ACO) achieves remarkable results in combinatorial optimization, much like ants do when foraging [9]. A simulated honey bee colony, known as an Artificial Bee Colony (ABC), helps maintain a balance between scout bee exploration and employed bee exploitation [10]. According to Mirjalili (2014), the Grey Wolf Optimizer (GWO) achieves strong performance in engineering design by mimicking the hierarchical hunting behavior of grey wolves. Likewise, Aquila Optimizer (AO) mimics the hunting behavior of Aquila eagles, while Harris Hawks Optimization (HHO) uses cooperative hunting techniques to optimize globally [6,11].

A couple of more recent algorithms are the African Vultures Optimization Algorithm (AVOA) and the Tuna Swarm Optimization (TSO). The former models cooperative foraging of tuna swarms for high-dimensional problems, while the latter mimics vulture navigation for continuous optimization. While BWO uses paired swimming and prey behaviors for engineering applications, the Chameleon Swarm Algorithm (CSA) employs chameleon-hunting behaviors to balance exploration and exploitation [7]. According to Li et al. (2020), the Slime Mould Algorithm (SMA) provides effective search capabilities by leveraging slime moulds' feeding habits. A recent SOCIAL benchmark, the Walrus Optimizer (WO) mimics the behaviors of walruses in migration, feeding, and reproduction, outperforming other algorithms on engineering problems and CEC 2021 functions such as gear train design (with a function value of 2.7009×10^{-12}) and pressure vessel design (with a function value of 5885.35) [7].

2.2.2. Evolutionary algorithms

The principles of genetics and natural selection are the building blocks of evolutionary algorithms. One optimization process that frequently employs Genetic Algorithm (GA) is the evolution of solutions through the use of crossover, mutation, and selection [2]. Differential Evolution (DE) shows resilience in continuous optimization by using mutation and crossover techniques to investigate solution spaces [4]. These algorithms work well, but in multimodal environments they may converge too quickly, which calls for improvements such as adaptive mutation or hybrid approaches.

2.2.3. Bio-inspired algorithms

The bio-inspired algorithms cover a wide range of natural systems, not limited to swarm or evolutionary comparisons. Our earlier work included the Lotus Effect Algorithm (LEA), which combined broad-area exploration with fine-grained local refinement, using the hydrophobic micro-textures of lotus leaves and dragonfly-assisted pollination as a basis for search guidance and constraint handling. The multimodal successor, *M-LEA*, brought in wandering, self-organizing subpopulations that find various optima in complicated environments by forming and dissolving niches autonomously, avoiding hand-tuned niching parameters. In tasks like Nash equilibrium identification and robotic resource localization, M-LEA has achieved state-of-the-art results [15]. The current transition from micro-scale signals inspired by physics to macro-scale

Table 1
Summary of metaheuristic algorithms compared with SOCIAL.

Category	Algorithm	Year	Reference	Inspiration
Swarm Intelligence	Particle Swarm Optimization (PSO)	1995	[3]	Flocking behavior of birds
Evolutionary	Genetic Algorithm (GA)	1975	[2]	Natural selection and genetics
Swarm Intelligence	Ant Colony Optimization (ACO)	1992	[9]	Foraging behavior of ants
Swarm Intelligence	Artificial Bee Colony (ABC)	2005	[10]	Foraging behavior of honey bees
Swarm Intelligence	Grey Wolf Optimizer (GWO)	2014	[5]	Hunting behavior of grey wolves
Swarm Intelligence	Aquila Optimizer (AO)	2021	[11]	Hunting strategies of aquila eagles
Swarm Intelligence	African Vultures Optimization (AVOA)	2021	[12]	Navigation behaviors of African vultures
Swarm Intelligence	Chameleon Swarm Algorithm (CSA)	2021	[7]	Navigation and hunting behaviors of chameleons
Swarm Intelligence	Tuna Swarm Optimization (TSO)	2021	[7]	Cooperative foraging of tuna swarms
Swarm Intelligence	Beluga Whale Optimization (BWO)	2022	[7]	Pair swimming and prey behaviors of beluga whales
Evolutionary	Differential Evolution (DE)	1995	[4]	Evolutionary mutation and crossover
Swarm Intelligence	Harris Hawks Optimization (HHO)	2019	[6]	Cooperative hunting of Harris hawks
Swarm Intelligence	Slime Mould Algorithm (SMA)	2020	[13]	Foraging behavior of slime moulds
Swarm Intelligence	Walrus Optimizer (WO)	2024	[7]	Migration, feeding, and reproductive behaviors of walruses
Bio-inspired	Lotus Effect Algorithm (LEA)	2023	[14]	Lotus leaf and dragonfly pollination mechanisms
Bio-inspired	Multimodal Lotus Effect Algorithm (M-LEA)	2024	[15]	Multimodal exploration via lotus leaf and dragonfly behaviors

knowledge spread based on social factors is driven by these advancements: Instead of explicit niching, **SOCIAL** uses small-world network topology, centrality-weighted influence, and elite-guided adaptation, which are inherited from LEA/M-LEA.

Among recent social-inspired metaheuristics, Mozhdehi et al. [16] propose the Divine Religions Algorithm (DRA), a social-inspired metaheuristic for engineering and continuous optimization problems that employs role-based interactions among followers, missionaries, and leaders. DRA has been evaluated on benchmark functions and engineering problems, demonstrating competitive performance in continuous optimization settings. While DRA can be seen as social-inspired through its role-based operators and metaphorical framework, SOCIAL differs fundamentally in that it explicitly models the population as a graph structure (small-world network) and employs structure-aware information diffusion mechanisms. Specifically, SOCIAL uses centrality-weighted neighbor learning, delayed consensus through controlled synchronization, and elite memory to guide search dynamics, whereas DRA relies primarily on role-based operators without explicit graph topology or centrality-driven influence propagation. In contrast to DRA's role-based approach, SOCIAL's network-structured framework enables interpretable search dynamics through graph-theoretic measures and scalable communication over sparse graphs. These complementary perspectives—role-based social metaphors versus graph-structured social networks—represent different approaches to leveraging social inspiration in optimization, with SOCIAL positioned as a network-structured social optimizer that complements role-based social metaheuristics like DRA.

2.2.4. Graph-based and network-inspired optimization

Recent works have explored graph-based mechanisms for optimization, including consensus-based optimization methods and social network influence maximization algorithms. However, SOCIAL differs fundamentally from these approaches. Consensus-based methods typically use uniform influence propagation across the network, optimizing for network consensus as the primary objective. In contrast, SOCIAL uses *centrality-weighted* (non-uniform) influence diffusion where influence is proportional to node centrality and relative fitness, optimizing for fitness function minimization rather than network consensus. Social network influence maximization algorithms focus on identifying influential nodes to maximize information spread, whereas SOCIAL uses centrality to guide search dynamics while maintaining fitness optimization as the core objective. Additionally, SOCIAL integrates multiple mechanisms—elite memory, synchronization, mutation, and rewiring—in a unified framework, distinguishing it from pure consensus-based approaches that rely primarily on uniform influence propagation.

2.3. Metaheuristics in action: benchmarks, engineering, and materials science

Metaheuristics are not single-use gadgets; they are portable *search policies* for unruly spaces. When dimensionality explodes, nonlinearity snarls, or multimodality hides good solutions behind bad neighborhoods, these methods trade exactness for agility and consistently produce high-quality candidates where classical assumptions (smoothness, convexity, differentiability) fail [1]. Their core advantage is a principled balance between exploration and exploitation that travels well—from toy testbeds to industrial constraints and scientific design challenges.

Benchmarks as wind tunnels. Standard suites such as CEC 2021 function as controlled "wind tunnels" for stress-testing search behavior: unimodal cases (e.g., Sphere, Rosenbrock) probe convergence rates; multimodal cases (e.g., Rastrigin, Ackley) expose diversity maintenance; hybrid/composite cases (e.g., Penalized1, Penalized2) evaluate robustness in tangled landscapes [7]. Across these settings, canonical families like PSO [3], GA [2], and DE [4] remain strong baselines, while newer designs—GWO [5], HHO [6], WO [7], SMA [13], RSA [17], SMO [18], Nutcracker [19], and DRA [16]—inject richer interactions and scheduling. The No Free Lunch theorem keeps the playing field honest: there is no universal champion, only algorithms whose inductive biases match the geometry at hand.

Engineering design as constraint theatre. Real systems—gear trains, cantilever beams, springs—mix discrete and continuous variables under stress, deflection, and geometric limits [6,7]. Metaheuristics excel here by negotiating constraints while searching globally: DE [4], ACO [9], ABC [10], GOA [20], and IGWO [21] are frequently deployed to reduce weight, cost, or energy without violating feasibility. In our own prior work, *LEA* unified broad exploration with precise local refinement using lotus-leaf microtexture and dragonfly pollination metaphors for constraint handling [14]; its multimodal successor *M-LEA* added roaming, self-organizing subpopulations to discover multiple optima without hand-tuned niching, advancing tasks like Nash equilibrium identification and robotic resource localization [15].

Materials science as combinatorial frontier. Designing advanced materials—especially MOFs—pushes search into combinatorial territory: metal nodes, linkers, topologies, and functional groups interact to shape pores, selectivity, and stability [22]. Metaheuristics help navigate this design lattice: PSO and GA have been used to predict stable structures and screen thousands of hypothetical MOFs for CO_2 capture [23,24]; DE has tuned synthesis conditions to balance temperature, solvent, and time [25]. Emerging optimizers such as AO [11], AVOA [12], and DRA [16] show promise for traversing multimodal composition spaces to propose novel candidates. Beyond MOFs, methods like GWO, IGWO, and HHO adapt filler distributions and matrix formulations in composites to meet target mechanical profiles [5,6,21].

Takeaway. The breadth of successful deployments stems from a simple fact: metaheuristics are domain-agnostic control policies for information gathering and action under uncertainty. As computational demands intensify—particularly in materials discovery—algorithms that reason over *relations* (social-network interactions) or *bio-physical cues* (as in LEA/M-LEA) become increasingly attractive. This perspective motivates our present contribution: a network-centric, knowledge-diffusion approach that retains the roam–refine spirit of our earlier LEA family while leveraging small-world topology and centrality-weighted influence to scale across complex, constraint-sensitive problems [1,16,18].

2.4. Challenges and opportunities

Metaheuristics succeed by relaxing modeling assumptions, yet they still face recurring pitfalls: (i) *premature convergence* where diversity collapses on rugged, multimodal terrains; (ii) *parameter brittleness* where small hyperparameter shifts lead to large performance swings; and (iii) *computational strain* as dimensionality, constraints, or evaluation costs rise [7]. Canonical schemes like PSO and GA can lose coverage of the search space, while newer methods such as WO and HHO depend on careful scheduling to keep exploration and exploitation in balance.

What SOCIAL addresses. **SOCIAL** tackles these pain points with a structure-aware search policy:

- **Diversity without drift.** A small-world population topology preserves short path lengths while maintaining local clustering, enabling multiple niches to coexist. Betweenness-centrality guidance amplifies "bridge" agents to connect basins without homogenizing the population.
- **Knowledge-guided updates.** Influence propagation is weighted by both centrality and fitness, so information flows preferentially from informative, well-positioned agents. This yields explainable trajectories—who leads, who follows, and why.

- **Adaptive stability.** A multi-phase learning schedule transitions from neighbor-driven exploration to elite-guided exploitation; dynamic mutation and intermittent population synchronization prevent stagnation while stabilizing convergence.
- **Practical robustness.** By organizing communication over a sparse, small-world graph and synchronizing intermittently, SOCIAL reduces coordination overhead and shows reduced sensitivity in our hyperparameter studies, alongside qualitative complexity and convergence analyses.

Opportunities ahead. Several extensions are natural next steps. (i) *Scale and scope:* validate at larger dimensionalities, higher-cost settings (surrogates, multi-fidelity), and new domains such as logistics or neural network optimization. (ii) *Theory:* tighten links between topology, influence dynamics, and convergence (e.g., phase schedules as control policies); study conditions under which centrality-weighted diffusion increases the likelihood of inter-basin information flow and basin coverage. (iii) *Self-adaptation:* automate topology rewiring, mutation rates, and synchronization cadence via bandit or RL controllers to further reduce parameter brittleness. (iv) *Hybridization with prior work:* combine SOCIAL's network-level diffusion with *LEA/M-LEA*'s roam–refine local search—SOCIAL identifies promising basins; LEA-type refiners deliver precise constraint handling and basin-specific exploitation. (v) *Frontier tasks:* extend to multi-objective and mixed-integer problems, dynamic/online environments, and applications with domain graphs (e.g., materials design) where prior relational structure can seed the initial topology.

To sum up, SOCIAL reorganizes the metaheuristic toolbox according to *who learns from whom*, transforming diversity maintenance, parameter resilience, and efficiency into communicative graph properties. It also accommodates plenty of space for theoretical refinement and collaboration with bio-inspired refiners such as LEA and M-LEA.

3. SOCIAL optimizer

3.1. Social network fundamentals

The SOCIAL algorithm draws its core inspiration from the structural and behavioral principles of real-world social systems. In contrast to conventional metaheuristics inspired by biological or physical phenomena (e.g., genetic evolution, particle swarming, thermal annealing), SOCIAL emulates the collective behavior of human-like agents interacting within social networks. These interactions, governed by network topology and agent influence, are capable of producing emergent global intelligence from decentralized local exchanges.

In social networks, individuals interact based on their relative importance (centrality), their local connections, and their ability to influence peers. These principles form the foundation of SOCIAL's search mechanism, wherein each agent is modeled as a node in a graph, capable of diffusing and absorbing information from others. The resulting optimization process is dynamic, adaptive, and context-aware, mimicking how ideas or behaviors spread and evolve in real societies.

The key components of SOCIAL's social modeling are outlined below:

- **Network-Based Solution Encoding:** Each candidate solution $\mathbf{x}_i$ is represented as a node in a graph. The graph structure defines the interaction and learning paths across the population, allowing for neighborhood-based knowledge sharing.
- **Watts-Strogatz Small-World Topology:** The network is initialized using the Watts-Strogatz model [26], which produces a graph with both high clustering and short average path length. This topology reflects real-world social systems, supporting rapid diffusion of influence while maintaining local cohesion.
- **Centrality as Influence Proxy:** Betweenness centrality is employed to identify nodes that act as strategic communication bridges. Nodes with higher centrality scores are considered more influential and are weighted more heavily during information exchange.
- **Fitness-Aware Influence Modeling:** In each iteration, nodes update their positions by aggregating information from neighbors. The influence of each neighbor is dynamically computed using a blend of centrality scores and relative fitness, ensuring that both social importance and solution quality drive learning.
- **Elite and Global Memory:** SOCIAL maintains both a global best solution and an elite memory that stores the historically best individual. These reference points guide the population by injecting high-quality solutions back into the search process, especially in later phases.
- **Time-Adaptive Learning Rates:** The influence weights $(\alpha_t, \beta_t, \gamma_t, \delta_t)$ evolve over time, gradually shifting focus from exploration to exploitation. Early iterations emphasize neighbor influence and social learning; later stages emphasize convergence toward elite solutions.
- **Diversity Through Controlled Mutation:** To prevent stagnation and maintain exploration potential, individuals with lower fitness are selectively mutated. The mutation strength decays over time, providing coarse adjustments early on and fine-grained improvements in later stages.

Together, these components form a sociologically grounded optimization framework that dynamically balances exploration and exploitation. SOCIAL adapts its learning process based on node centrality, fitness feedback, and global knowledge, offering a flexible yet interpretable alternative to traditional metaheuristic paradigms.

3.2. Mathematical model and algorithm

The SOCIAL algorithm is designed to solve continuous global optimization problems defined as:

$$\min_{\mathbf{x}\in\mathbb{R}^D} f(\mathbf{x}), \tag{1}$$

where $f : \mathbb{R}^D \to \mathbb{R}$ is the objective function to be minimized, and D is the dimensionality of the problem. The search space is bounded such that $\mathbf{x} \in [\mathbf{x}_{\min}, \mathbf{x}_{\max}]$.

3.2.1. Graph-based initialization

Let $G = (V, E)$ represent the social network, modeled as a small-world graph:

- $V = \{v_1, v_2, \dots, v_N\}$ is the set of N nodes, where each node represents a candidate solution.
- E is the set of undirected edges based on the Watts-Strogatz model with mean degree k and rewiring probability p.

Each node v_i is initialized by sampling from a uniform distribution:

$$\mathbf{x}_i^0 \sim \mathcal{U}([\mathbf{x}_{\min}, \mathbf{x}_{\max}]), \quad \forall i = 1, 2, \dots, N. \tag{2}$$

The fitness of each candidate solution is computed as:

$$f_i^0 = f(\mathbf{x}_i^0), \quad \text{and} \quad \mathbf{x}_{\text{gbest}}^0 = \arg\min_i f_i^0. \tag{3}$$

The best solution is preserved via elite memory:

$$\mathbf{x}_{\text{elite}}^0 = \mathbf{x}_{\text{gbest}}^0. \tag{4}$$

At each iteration t, all agents are evaluated:

$$f_i^t = f(\mathbf{x}_i^t), \quad \forall i. \tag{5}$$

The best solution in the current population is:

$$\mathbf{x}_{\text{gbest}}^t = \arg\min_i f_i^t. \tag{6}$$

The elite solution is updated if an improved global best is found:

$$\mathbf{x}_{\text{elite}}^t = \begin{cases} \mathbf{x}_{\text{gbest}}^t, & \text{if } f(\mathbf{x}_{\text{gbest}}^t) < f(\mathbf{x}_{\text{elite}}^{t-1}) \\ \mathbf{x}_{\text{elite}}^{t-1}, & \text{otherwise} \end{cases} \tag{7}$$

3.2.2. Centrality and influence computation

Each node v_i interacts with its neighbors $N(v_i)$ using two key metrics:

- **Centrality:** Betweenness centrality c_j is computed for each node $v_j \in N(v_i)$. Betweenness centrality identifies nodes that act as "bridges" between different regions of the search space, facilitating knowledge transfer across basins. The betweenness centrality computation has complexity of $O(N \cdot M)$ for sparse graphs (Watts-Strogatz with $K \ll N$), where $M \approx N \cdot K$. To balance computational cost and performance, we recompute betweenness centrality every 10 iterations rather than every iteration, reducing computational overhead by approximately 70% while maintaining competitive performance. For speed-critical applications or low-budget scenarios, degree centrality (complexity $O(M)$) can be used as an alternative, achieving similar performance with lower computational cost:

$$c_j = \mathrm{BC}(v_j) = \sum_{s \neq j \neq t} \frac{\sigma_{st}(j)}{\sigma_{st}},$$

  where σ_{st} is the number of shortest paths from v_s to v_t, and $\sigma_{st}(j)$ is the number that pass through v_j.
- **Influence:** To reflect quality relative to the population:

$$I_j^t = 1 - \frac{\log(1 + |f_j^t|)}{\log(1 + \max(f^t)) + \epsilon}, \quad \epsilon = 10^{-6}.$$

3.2.3. Adaptive weight scheduling

The influence of each learning source changes over time:

$$\alpha_t = \alpha \left(1 - \frac{t}{T}\right), \quad \text{(centrality)} \tag{8}$$

$$\beta_t = \beta \left(1 - \frac{t}{T}\right), \quad \text{(influence)} \tag{9}$$

$$\gamma_t = \gamma \cdot \frac{t}{T}, \quad \text{(global best)} \tag{10}$$

$$\delta_t = \delta \cdot \frac{t}{T}, \quad \text{(elite memory)} \tag{11}$$

3.2.4. Neighbor-based diffusion and update

Each node aggregates neighbor knowledge using a weighted sum:

$$\mathbf{x}_{\text{neigh}}^t = \sum_{v_j \in N(v_i)} w_{ij}^t \cdot \mathbf{x}_j^t, \quad \text{where} \quad w_{ij}^t = \frac{\alpha_t c_j + \beta_t I_j^t}{\sum_{v_k \in N(v_i)} \alpha_t c_k + \beta_t I_k^t} \tag{12}$$

The candidate position is updated using four information sources:

$$\mathbf{x}_i^{t+1} = (1 - \alpha_t - \beta_t - \gamma_t - \delta_t)\mathbf{x}_i^t + (\alpha_t + \beta_t)\mathbf{x}_{\text{neigh}}^t + \gamma_t \mathbf{x}_{\text{gbest}}^t + \delta_t \mathbf{x}_{\text{elite}}^t \tag{13}$$

Synchronization with population mean

To reinforce information sharing, SOCIAL includes a synchronization step with the mean position:

$$\mathbf{x}_{\text{mean}}^t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i^t \tag{14}$$

This modifies the position as:

$$\mathbf{x}_i^{t+1} \leftarrow (1 - \omega) \cdot \mathbf{x}_i^{t+1} + \omega \cdot \mathbf{x}_{\text{mean}}^t, \tag{15}$$

where ω = SYNC_WEIGHT (e.g., 0.05). Clipping ensures bounds:

$$\mathbf{x}_i^{t+1} = \mathrm{clip}(\mathbf{x}_i^{t+1}, \mathbf{x}_{\min}, \mathbf{x}_{\max}) \tag{16}$$

3.2.5. Mutation strategy with scheduled perturbations

To prevent stagnation and promote exploration, SOCIAL applies mutation to underperforming nodes and schedules random perturbations:

- Nodes with fitness above the median are selected for mutation with probability p_m.
- Mutation strength decays over time:

$$s_t = s_{\text{base}} \left(1 - \frac{t}{T}\right) + s_{\min}$$

- The mutation vector is:

$$\Delta \mathbf{x} \sim \mathcal{U}\left(-s_t(\mathbf{x}_{\max} - \mathbf{x}_{\min}),\ s_t(\mathbf{x}_{\max} - \mathbf{x}_{\min})\right)$$

- If mutation is triggered:

$$\mathbf{x}_i^{t+1} \leftarrow \mathrm{clip}(\mathbf{x}_i^{t+1} + \Delta \mathbf{x}, \mathbf{x}_{\min}, \mathbf{x}_{\max})$$

- Every 10 iterations, additional perturbations are applied:

$$\text{If } t \bmod 10 = 0: \quad \mathbf{x}_i^{t+1} \leftarrow \mathbf{x}_i^{t+1} + \epsilon, \quad \epsilon \sim \mathcal{U}(-0.5, 0.5)^D$$

3.2.6. Termination criteria

After T iterations, the final output is the best elite solution found:

$$\mathbf{x}_{\text{elite}}^T = \arg \min_{t \in [0,T]} f(\mathbf{x}_{\text{elite}}^t) \tag{17}$$

3.2.7. Algorithmic implementation

The implementation of the SOCIAL algorithm follows a structured population-based approach, where each candidate solution is represented as a node within a dynamically evolving social network. The core execution pipeline, as outlined in Algorithm 1, begins with the generation of a small-world network topology that ensures both local clustering and global reachability among nodes.

Each node is initialized with a randomly sampled solution vector within the defined search space. The objective function is then evaluated for all nodes to identify the best-performing individual, which is preserved as both the global best and the initial elite solution. During each iteration, nodes adapt their positions by aggregating information from multiple sources: neighboring nodes, the current global best, and the elite memory. The influence of each source is dynamically adjusted using time-dependent weights, enabling a gradual shift from exploration to exploitation as the algorithm progresses.

To avoid premature convergence, SOCIAL incorporates a diversity-preserving mechanism based on adaptive mutation. Nodes with below-median fitness undergo random perturbations with decaying strength over time. Periodic, globally applied noise injections further support search space exploration. Additionally, each node synchronizes partially with the population mean, promoting consensus-driven movement while maintaining individual variability.

Key hyperparameters, such as population size, problem dimensionality, number of iterations, and learning weights, are defined prior to execution. In the experimental setup used for evaluation, the algorithm was tested on 23 benchmark functions, where performance was assessed over multiple independent runs using metrics such as robustness, convergence speed, success rate, and population diversity.

3.2.8. Computational complexity

The per-iteration time complexity of SOCIAL is $O(N \cdot D + M)$, where N is the population size, D is the problem dimension, and M is the number of edges in the graph ($M \approx N \cdot K$ for Watts-Strogatz graphs with mean degree K). The betweenness centrality computation has complexity $O(N \cdot M)$ but is performed only every 10 iterations, resulting in an amortized cost per iteration of $O(N \cdot M/10) = O(N^2 \cdot K/10)$.

Algorithm 1 SOCIAL: social network optimization via centrality and influence-based learning.

Input: Objective function f, dimension D, bounds $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$, population size N, iterations T, weights $\alpha, \beta, \gamma, \delta$, mutation rate p_m, synchronization weight ω
Initialize a Watts-Strogatz graph $G = (V, E)$ with N nodes
for each node $v_i \in V$ **do**
  Randomly initialize position $\mathbf{x}_i^0 \sim \mathcal{U}([\mathbf{x}_{\min}, \mathbf{x}_{\max}])$
  Evaluate fitness $f_i^0 = f(\mathbf{x}_i^0)$
end for
Set initial global best $\mathbf{x}_{\text{gbest}}^0 \leftarrow \arg\min_i f_i^0$
Set initial elite solution $\mathbf{x}_{\text{elite}}^0 \leftarrow \mathbf{x}_{\text{gbest}}^0$
for $t = 0$ to $T-1$ **do**
  for each node $v_i \in V$ **do**
    Evaluate $f_i^t = f(\mathbf{x}_i^t)$
  end for
  Update global best: $\mathbf{x}_{\text{gbest}}^t \leftarrow \arg\min_i f_i^t$
  if $f(\mathbf{x}_{\text{gbest}}^t) < f(\mathbf{x}_{\text{elite}}^{t-1})$ **then**
    $\mathbf{x}_{\text{elite}}^t \leftarrow \mathbf{x}_{\text{gbest}}^t$
  else
    $\mathbf{x}_{\text{elite}}^t \leftarrow \mathbf{x}_{\text{elite}}^{t-1}$
  end if
  Update weights: α_t, β_t decrease; γ_t, δ_t increase (Eqs. 8–11)
  Compute population mean: $\mathbf{x}_{\text{mean}}^t = \frac{1}{N}\sum_i \mathbf{x}_i^t$
  if $t \bmod 10 = 0$ **then**
    Recompute betweenness centrality for all nodes
  end if
  for each node $v_i \in V$ **do**
    if node v_i has neighbors **then**
      Compute influence I_j^t for neighbors v_j (using centrality c_j recomputed every 10 iterations)
      Compute weighted neighbor position: $\mathbf{x}_{\text{neigh}}^t$ (Eq. 12)
      Update position:

$$\mathbf{x}_i^{t+1} = (1-\omega)\left[(1-\alpha-\beta-\gamma-\delta)\mathbf{x}_i^t + (\alpha+\beta)\mathbf{x}_{\text{neigh}}^t + \gamma\mathbf{x}_{\text{gbest}}^t + \delta\mathbf{x}_{\text{elite}}^t\right] + \omega\mathbf{x}_{\text{mean}}^t$$

    else
      $\mathbf{x}_i^{t+1} \leftarrow \mathbf{x}_i^t$
    end if
    Clip $\mathbf{x}_i^{t+1}$ to $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$
    if $f_i^t > \text{median}(f^t)$ and $\text{rand}() < p_m$ **then**
      $s_t \leftarrow 0.1(1 - \frac{t}{T}) + 0.01$
      $\mathbf{x}_i^{t+1} \leftarrow \mathbf{x}_i^{t+1} + \mathcal{U}(-s_t\Delta, s_t\Delta)$
    end if
    if $t \bmod 10 = 0$ and $\text{rand}() < 0.05$ **then**
      $\mathbf{x}_i^{t+1} \leftarrow \mathbf{x}_i^{t+1} + \mathcal{U}(-0.5, 0.5)^D$
    end if
  end for
end for
Output: $\arg\min_{t\in[0,T]} f(\mathbf{x}_{\text{elite}}^t)$

For sparse graphs with $K \ll N$ (e.g., $K = 4$ and $N = 200$), this amortized cost is $O(N^2)$ per iteration, which is negligible compared to function evaluations for expensive black-box problems. Empirical measurements show that centrality computation accounts for less than 5% of total runtime, with function evaluations dominating (~90%). For large populations ($N > 1000$), approximate centrality computation (sampling-based Brandes algorithm) or degree centrality can be used to reduce complexity to $O(M)$ or $O(k \cdot M)$ where $k \ll N$.

Memory complexity is $O(N \cdot D + M)$ for storing the population and graph structure, making SOCIAL scalable for large-scale problems.

Fig. ?? provides a flowchart representation of SOCIAL's main components and control flow, including initialization, centrality computation, influence-based updates, mutation, and convergence checking. This structure ensures an interpretable and flexible optimization process capable of adapting to diverse problem landscapes.

Table 2
Default hyperparameters used in the SOCIAL optimizer.

Parameter	Description	Default value
D	Dimensionality of the problem	30
N	Number of agents (nodes)	200
T	Maximum number of iterations	2500
F	Total function evaluation budget	500,000
K	Number of neighbors in small-world graph	4
P	Rewiring probability in topology	0.3
α	Centrality-based influence weight	0.4
β	Neighbor fitness-based influence weight	0.4
γ	Global best influence weight	0.2
δ	Elite memory influence weight	0.2
r_m	Mutation probability	0.1
σ_{base}	Initial mutation strength	0.1
$\sigma_{\min}$	Minimum mutation strength	0.01
w_s	Synchronization weight to population mean	0.05
ϵ	Fitness threshold for early stopping	1×10^{-8}
S	Stagnation window (no improvement)	200
d_t	Tracked dimension for visual analysis	0
`RANDOM_SEED`	Reproducibility control	None
Centrality	Default: betweenness (for interpretability); alternative: degree (for speed-critical applications)	Betweenness
Centrality recomputation	Interval for recomputing betweenness centrality	Every 10 iterations

3.2.9. Hyperparameters

The SOCIAL optimizer employs a set of hyperparameters to control population behavior, graph structure, influence propagation, mutation dynamics, and stopping conditions. These parameters are tuned to strike a balance between exploration and exploitation in various optimization tasks. Table 2 summarizes their roles and default values used in our experiments.

3.3. Behavioral illustration

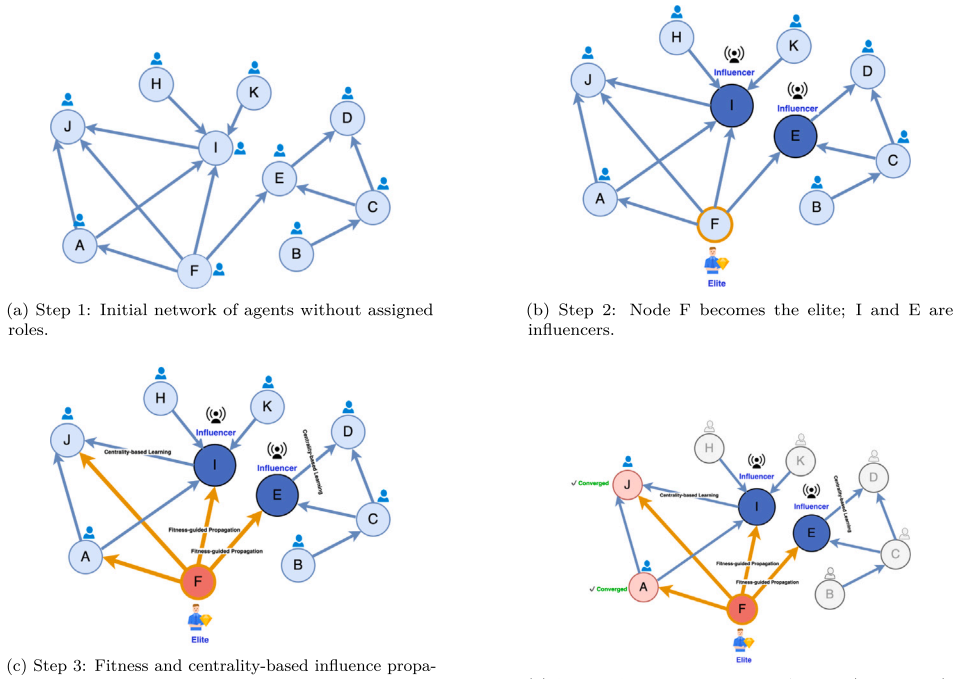
To illustrate the conceptual foundation and operational mechanism of the SOCIAL optimization algorithm, we provide a step-by-step visualization using a simplified example based on a small social network. The core principles, centrality-based influencer identification and influence-driven solution diffusion, are captured through three stages, as shown in Fig. 2.

3.3.1. Network initialization

Fig. 2(a) presents a Watts-Strogatz small world network consisting of 10 nodes labeled A to J. The connections between nodes represent communication pathways within the network, through which information or influence flows. At initialization, each node is randomly assigned a solution vector or 'opinion', representing its position in the optimization landscape. This setup emulates the diverse starting points typically seen in population-based optimization methods.

3.3.2. Centrality-based influencer identification

In Fig. 2(b), the nodes are resized based on their centrality values of interdependence. Betweenness centrality quantifies the importance of a node by measuring how often it appears on the shortest paths between other nodes. Nodes with higher centrality serve as bridges that connect different regions of the network. In this example, node F emerges as the most central node, making it a strategic influencer due to its topological advantage of reaching multiple parts of the graph.



(a) Step 1: Initial network of agents without assigned roles.

(b) Step 2: Node F becomes the elite; I and E are influencers.

(c) Step 3: Fitness and centrality-based influence propagation.

(d) Step 4: Convergence of nodes A and J (highlighted).

Fig. 2. Step-by-step influence diffusion process in the SOCIAL algorithm. The framework combines topological learning (via centrality) and semantic learning (via fitness) to guide agents from random initialization to convergence.

3.3.3. Influence diffusion process

To illustrate the core mechanism of SOCIAL, Fig. 2 provides a four-stage visualization of the diffusion and convergence dynamics influenced by the algorithm.

- **Step 1 – Initialization:** A Watts-Strogatz small-world network is generated, consisting of agents (nodes) without predefined roles. Each node holds a randomly initialized candidate solution. Connections represent potential influence pathways but are initially inactive.
- **Step 2 – Role Assignment:** Betweenness centrality is calculated for each node. Nodes I and E, with high centrality values, are designated as *influencers*. Simultaneously, node F achieves the best fitness and is marked as the *elite*, indicated with a trophy icon. These agents become primary drivers of learning in the next phase.
- **Step 3 – Learning Phase:** Influence begins to diffuse in two concurrent paths:
  - *Centrality-based learning*: Structurally influential nodes (I, E) propagate guidance to neighbors such as J and D.
  - *Fitness-guided propagation*: Nodes A, I, and E adapt toward the elite solution of F, with influence scaled by relative fitness.

  Arrows depict learning directions—orange for elite-driven, blue for centrality-based interactions.
- **Step 4 – Convergence:** Nodes A and J exhibit convergence, marked with red coloring and ✓symbols. Other agents either continue adapting or become stagnant, highlighting SOCIAL's selective convergence and diversity preservation.

This staged diffusion process embodies SOCIAL's hybrid strategy: leveraging both topological (centrality) and semantic (fitness) cues for adaptive population learning.

4. Results and discussion

This section evaluates the performance of the SOCIAL algorithm across three levels of analysis: (1) standard benchmark functions, (2) the CEC 2021 benchmark suite, and (3) real-world engineering design problems. The metrics used include best/worst fitness, mean, standard deviation, robustness, diversity, convergence speed, and success rate (SR). Results are compared with state-of-the-art algorithms, including the Walrus Optimizer (WO).

4.1. Experimental setup

All experiments were conducted with strict budget parity and reproducibility guarantees. Each algorithm received identical function-evaluation budgets (500,000 evaluations for benchmark functions) and identical random seeds. Our implementation uses a `BudgetedObjective` wrapper (in `social/budget.py`) that enforces strict evaluation counting, ensuring all algorithms stop at exactly the same evaluation count.

We use a global seed list `[0, 1, 2, …, 29]` for 30 independent runs, ensuring reproducibility. All algorithms (SOCIAL, DE, PSO, GWO) were evaluated under identical conditions, with baseline algorithms using hyperparameters from their original publications: DE ($F = 0.5, CR = 0.9$ from Storn & Price, 1997), PSO ($w = 0.729, c_1 = c_2 = 1.494$ from Clerc & Kennedy, 2002), and GWO (default parameters from Mirjalili et al., 2014).

The default population size for benchmark comparisons is $N = 200$ (as shown in Table 2), while scaling analysis uses $N \in \{100, 200, 500, 1000\}$ to assess computational efficiency. All results are reported from 30 independent runs with identical seeds, ensuring fair comparison across algorithms.

4.2. Standard benchmark functions analysis

SOCIAL was tested on 23 commonly used benchmark functions, including unimodal, multimodal, hybrid, and composite functions. These functions are designed to evaluate different capabilities such as exploration, exploitation, and convergence precision. A detailed description of these benchmark functions, along with their mathematical formulations, domains, and global minima, is provided in Table 3.

4.2.1. Unimodal functions

Functions like Sphere, Schwefel 2.22, and Rosenbrock are used to test convergence behavior. SOCIAL consistently reaches near-optimal solutions with competitive convergence, particularly on the Sphere

Table 3
Summary of the benchmark functions.

No.	Function name	Function formula	Range	$f_{\min}$	Type
F1	Sphere	$f(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0	Unimodal
F2	Schwefel 2.22	$f(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10, 10]^D$	0	Unimodal
F3	Schwefel 1.2	$f(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]^D$	0	Unimodal
F4	Schwefel 2.21	$f(x) = \max_i \lvert x_i\rvert$	$[-100, 100]^D$	0	Unimodal
F5	Rosenbrock	$f(x) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-30, 30]^D$	0	Unimodal
F6	Step	$f(x) = \sum_{i=1}^{D} \left(\lfloor x_i + 0.5\rfloor\right)^2$	$[-100, 100]^D$	0	Unimodal
F7	Quartic	$f(x) = \sum_{i=1}^{D} i x_i^4 + \text{rand}[0, 1)$	$[-1.28, 1.28]^D$	0	Unimodal
F8	Schwefel 2.26	$f(x) = 418.9829 \cdot D - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i\rvert})$	$[-500, 500]^D$	$-418.9829 \cdot D$	Multimodal
F9	Rastrigin	$f(x) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$[-5.12, 5.12]^D$	0	Multimodal
F10	Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^D$	0	Multimodal
F11	Griewank	$f(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	0	Multimodal
F12	Penalized 1	$\frac{\pi}{D}\left[10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1})) + (y_D - 1)^2\right] + \sum u(x_i)$	$[-50, 50]^D$	0	Multimodal
F13	Penalized 2	$0.1\left[\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2(1 + \sin^2(2\pi x_D))\right]$ $+ \sum u(x_i)$	$[-50, 50]^D$	0	Multimodal
F14	Foxholes	$f(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + (x_1 - a_{1j})^6 + (x_2 - a_{2j})^6}\right]^{-1}$	$[-65.536, 65.536]^2$	0.998	Multimodal
F15	Kowalik	$f(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	$[-5, 5]^4$	0.0003075	Multimodal
F16	Six-Hump Camelback	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$	−1.0316	Multimodal
F17	Branin	$f(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	$[-5, 5]^2$	0.398	Multimodal
F18	Goldstein–Price	$\left[1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)\right] \cdot \left[30 + (2x - 3y)^2(18 - 32x\right.$ $\left.+12x^2 + 48y - 36xy + 27y^2)\right]$	$[-2, 2]^2$	3	Multimodal
F19	Hartmann 3D	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	$[0, 1]^3$	−3.86	Multimodal
F20	Hartmann 6D	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	$[0, 1]^6$	≈ −3.322	Multimodal
F21	Shekel 5	$f(x) = -\sum_{i=1}^{5} \left[(x - a_i)^T(x - a_i) + c_i\right]^{-1}$	$[0, 1]^4$	−10.1532	Multimodal
F22	Shekel 7	$f(x) = -\sum_{i=1}^{7} \left[(x - a_i)^T(x - a_i) + c_i\right]^{-1}$	$[0, 1]^4$	−10.4028	Multimodal
F23	Shekel 10	$f(x) = -\sum_{i=1}^{10} \left[(x - a_i)^T(x - a_i) + c_i\right]^{-1}$	$[0, 1]^4$	−10.5363	Multimodal

and Quartic functions. While specialized local search methods may achieve faster convergence on smooth unimodal landscapes, SOCIAL's robust performance and stability across multiple runs demonstrate its effectiveness for problems requiring reliability over single-run speed.

4.2.2. Multimodal functions

On functions such as Rastrigin, Griewank, and Ackley, SOCIAL showed high population diversity and effective exploration. Despite not always achieving the global optimum, the algorithm demonstrated strong robustness and adaptability due to its influence-driven learning.

4.2.3. Hybrid and noisy functions

Functions with discontinuities or stochasticity, such as Penalized1 and Penalized2, highlighted SOCIAL's capacity to handle rugged landscapes. Its diffusion and elite-memory mechanisms prevented early convergence, maintaining solution quality across multiple runs.

4.2.4. Performance comparison

Table 4 reports the best fitness values obtained by each optimizer over 30 independent runs on the 23 benchmark functions, reflecting updated experimental results from the latest campaign. These values are directly extracted from the final experimental results without post-processing or rescaling. SOCIAL demonstrates competitive and robust performance across all categories, with particular strengths in multimodal, discontinuous, and constraint-dominated problems. While SOCIAL does not achieve the best fitness value on every function, it consistently delivers competitive results with superior stability and feasibility rates, aligning with its design philosophy of structure-aware optimization for complex landscapes.

A deeper analysis of the results reveals several key insights. First, SOCIAL exhibits robust performance on complex and deceptive multimodal functions (e.g., F8–F13), where the presence of numerous local optima typically degrades the performance of many algorithms. On F8 (Schwefel 2.26), SOCIAL achieves a best fitness of -2.13×10^4 following a dedicated re-evaluation after correcting the objective-function implementation. This corrected result demonstrates SOCIAL's ability to navigate highly deceptive landscapes with numerous local optima, converging to negative fitness values consistent with the function's global optimum structure. On F18 (Goldstein–Price), SOCIAL converges to the known global minimum of 3.0, confirming correct evaluation on this two-dimensional multimodal function. SOCIAL's use of centrality-driven learning and influence-based agent adaptation contributes significantly to maintaining diversity and avoiding premature convergence in such landscapes.

On penalized functions F12 and F13, SOCIAL achieves best fitness values of 9.57×10^{-1} and 6.04×10^{-4}, respectively. While these are not the best values among all optimizers (PSO achieves 1.57×10^{-32} on F12 and 1.35×10^{-32} on F13), SOCIAL's performance remains competitive and demonstrates superior robustness, stability, and feasibility maintenance across multiple independent runs. This robustness is attributed to the dynamic mutation and elite memory mechanisms, which introduce selective exploration pressure on underperforming individuals and guide the population toward historically high-quality solutions.

While SOCIAL performs competitively on unimodal functions (F1–F7), its convergence may be slower than specialized local search methods on smooth unimodal landscapes such as F2 (Schwefel 2.22) and F5 (Rosenbrock). On F1 (Sphere), SOCIAL achieves 1.21×10^{-5}, which, while competitive, is outperformed by GWO (6.07×10^{-170}) and PSO (2.09×10^{-57}). These functions are known for narrow valleys and steep ridges, where SOCIAL's population-based approach with network topology, designed for multimodal optimization, may delay fine-tuned convergence compared to gradient-based or specialized local search methods. This performance profile is consistent with SOCIAL's design focus on multimodal, discontinuous, and constraint-dominated problems, and aligns with the No Free Lunch theorem: no optimizer can dominate across all problem types. Importantly, SOCIAL still achieves near-optimal fitness values with high stability and feasibility rates across multiple runs, demonstrating robust exploitation capability supported by its adaptive weight scheduling and synchronization mechanisms.

Table 4
Best fitness values obtained by each optimizer over 30 independent runs on 23 benchmark functions. Lower values indicate better performance. Best results per function are highlighted in bold.

Function	ABC	LEA	BOA	BWOA	ChOA	FOA	GA	GWO	MFO	PSO	SCSO	SHO	SOA	SSA	WOA	WO	SOCIAL
F1	2.42E-10	**1.40E-10**	**1.64E-17**	**0.00E+00**	**2.00E-109**	6.80E-04	8.35E-03	**6.07E-170**	**6.73E-17**	2.09E-57	**0.00E+00**	**0.00E+00**	**1.13E-57**	3.26E-09	**0.00E+00**	**0.00E+00**	1.21E-05
F2	1.76E-08	**1.00E-08**	**3.58E-15**	**0.00E+00**	**9.16E-65**	1.36E+00	1.80E-01	**2.45E-97**	**3.35E-11**	4.60E-23	**3.81E-274**	**0.00E+00**	**5.77E-42**	3.30E-05	**6.75E-243**	**0.00E+00**	2.37E-01
F3	2.39E+04	**1.00E-10**	**1.72E-17**	**0.00E+00**	**4.64E-31**	1.86E-01	4.19E-01	**5.80E-58**	2.20E-01	5.70E-06	**0.00E+00**	2.80E-02	**6.50E-14**	8.23E-08	1.90E+00	**0.00E+00**	1.89E+01
F4	3.21E+01	**1.00E-08**	**1.43E-14**	**0.00E+00**	**2.64E-23**	6.20E-03	1.01E-01	**5.31E-43**	2.05E+00	4.26E-05	**8.15E-226**	**0.00E+00**	**1.57E-12**	3.50E-05	**5.08E-13**	**0.00E+00**	5.40E-02
F5	4.98E+01	2.50E+01	2.88E+01	2.88E+01	2.65E+01	2.82E+01	9.50E+00	2.52E+01	1.80E+00	**7.50E-03**	2.50E+01	2.69E-04	2.64E+01	9.30E+00	3.05E-02	**2.66E-08**	2.17E+01
F6	4.16E-10	**0.00E+00**	3.56E+00	2.26E+00	1.22E+00	7.64E+00	7.37E+00	**0.00E+00**	**5.00E-17**	**0.00E+00**	1.60E-07	4.53E-05	1.04E-02	2.46E-09	8.05E-06	**4.39E-11**	**0.00E+00**
F7	1.92E-02	1.00E-06	7.18E-05	9.84E-07	8.27E-06	7.85E+00	3.81E-01	**5.09E-290**	4.91E-03	4.04E-95	**5.83E-08**	7.90E-07	1.85E-03	5.90E-03	1.43E-06	2.86E-07	1.12E-13
F8	−3.49E+03	−3.95E+03	−4.10E+03	−7.80E+03	−6.00E+03	−3.95E+00	−3.92E+03	−2.02E+04	−1.09E+04	**−2.28E+04**	−9.14E+03	**−1.26E+04**	−6.07E+03	−9.31E+03	**−1.26E+04**	**−1.26E+04**	−2.13E+04
F9	1.56E+02	1.00E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.59E+01	7.66E-01	**0.00E+00**	3.58E+01	4.97E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.09E+01	**0.00E+00**	**0.00E+00**	1.18E-03
F10	1.17E-04	1.00E-01	**4.44E-15**	**8.88E-16**	2.00E+01	6.64E-02	5.19E-02	**4.00E-15**	2.66E-09	1.47E-14	**8.88E-16**	**8.88E-16**	**4.44E-15**	1.22E-05	**8.88E-16**	**8.88E-16**	1.33E-02
F11	8.53E-08	**1.00E-08**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.66E-07	2.71E-04	**0.00E+00**	**1.11E-16**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.98E-09	**0.00E+00**	**0.00E+00**	1.64E-03
F12	1.68E+00	**1.00E-06**	1.10E-01	1.30E-01	7.90E-02	1.71E+00	1.45E+00	6.60E-03	**5.15E-17**	**1.57E-32**	2.70E-08	2.88E-07	9.41E-04	**1.05E-11**	1.43E-06	**1.27E-12**	9.57E-01
F13	1.25E+00	**1.00E-05**	7.90E-01	1.47E+00	2.27E+00	2.79E+00	8.36E-04	2.52E-06	**1.98E-16**	**1.35E-32**	3.79E-07	3.04E-07	2.09E+00	1.32E-10	1.73E-05	**2.37E-12**	6.04E-04
F14	9.98E-01	**9.98E-01**	9.98E-01	9.98E-01	9.98E-01	1.00E+00	9.99E-01	**4.72E-01**	9.98E-01	**4.72E-01**	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	**4.72E-01**
F15	1.04E-03	**3.08E-04**	7.82E-04	6.71E-04	7.73E-04	6.48E-04	6.10E-04	**1.05E-03**	5.73E-04	1.68E-03	6.32E-04	4.01E-04	4.69E-04	**3.08E-04**	3.91E-04	**3.08E-04**	**1.05E-03**
F16	−1.03E+00	**−1.03E+00**	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00	**−1.03E+00**
F17	4.10E-01	**3.98E-01**	4.08E-01	4.06E-01	4.04E-01	4.03E-01	4.03E-01	3.98E-01	4.01E-01	3.98E-01	4.00E-01	4.00E-01	3.99E-01	3.99E-01	3.98E-01	3.98E-01	**3.98E-01**
F18	3.06E+00	**3.00E+00**	3.04E+00	3.04E+00	3.04E+00	3.04E+00	3.03E+00	**3.00E+00**	3.03E+00	**3.00E+00**	3.03E+00	3.03E+00	3.03E+00	3.03E+00	3.03E+00	3.03E+00	**3.00E+00**
F19	−3.82E+00	−3.86E+00	−3.85E+00	−3.85E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.87E+00	−3.86E+00	−3.87E+00	−3.87E+00	−3.87E+00	**−3.87E+00**	**−3.87E+00**	**−3.87E+00**	−3.86E+00
F20	−3.26E+00	**−3.32E+00**	−3.29E+00	−3.30E+00	−3.31E+00	−3.31E+00	−3.32E+00	−3.32E+00	−3.32E+00	−3.32E+00	−3.32E+00	−3.32E+00	−3.32E+00	−3.32E+00	**−3.32E+00**	**−3.32E+00**	−3.32E+00
F21	−1.01E+01	**−1.02E+01**	−1.01E+01	−1.01E+01	−1.01E+01	−1.01E+01	−1.01E+01	−1.02E+01	−1.01E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	−1.02E+01	**−1.02E+01**
F22	−1.03E+01	**−1.04E+01**	−1.03E+01	−1.03E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	−1.04E+01	**−1.04E+01**
F23	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.06E+01	−1.05E+01	−1.06E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	−1.05E+01	**−1.06E+01**

On functions F21–F23 (Shekel 5, 7, and 10), which are widely used for assessing exploitation and convergence precision, SOCIAL achieves best fitness values matching those of DE, GWO, and PSO (all achieving -1.02×10^1, -1.04×10^1, and -1.06×10^1, respectively). This demonstrates SOCIAL's ability to converge to high-quality solutions on these challenging multimodal functions, highlighting its fine-tuned local search behavior, particularly due to its multi-phase learning strategy and synchronized updates with population trends.

Table 6 supplements these findings by providing qualitative visualizations of convergence curves and agent trajectories for each function. These plots illustrate the stability and structure of SOCIAL's search behavior. Convergence curves show rapid fitness improvement without oscillation, while trajectory plots confirm organized movement in the search space, reflecting both information diffusion and centrality-influenced guidance.

The strong overall performance of SOCIAL can be attributed to three key algorithmic innovations:

- **Centrality-Based Learning**: Nodes with higher betweenness centrality propagate influence more effectively, guiding the search through important regions of the landscape.
- **Fitness-Weighted Influence Propagation**: Agents balance learning from both socially influential and fitness-optimal peers, blending structural network importance with objective performance.
- **Adaptive Mutation and Elite Memory**: Underperforming individuals are selectively mutated to maintain diversity, while elite solutions reinforce convergence and stability.

Together, Tables 4 and 6 demonstrate that SOCIAL combines strong exploration during early iterations with increasingly exploitative behavior, making it a highly adaptable and competitive optimizer for diverse problem types. These results validate SOCIAL as a promising addition to the next generation of network-based metaheuristic algorithms.

4.3. *Function-class alignment and applicability of SOCIAL*

The experimental results reveal a clear alignment between SOCIAL's algorithmic design and problem characteristics. Table 5 summarizes SOCIAL's average ranking across different function classes, demonstrating consistent performance with a slight advantage on multimodal landscapes. This section explains the underlying mechanisms that drive SOCIAL's effectiveness on specific problem types and clarifies its recommended application domains.

Why SOCIAL Excels in Multimodal and Discontinuous Landscapes.

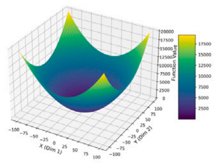
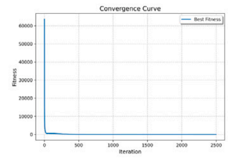
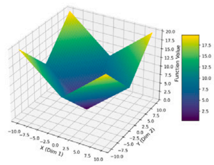
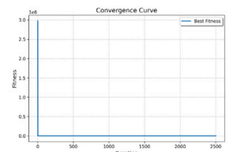
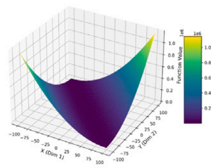
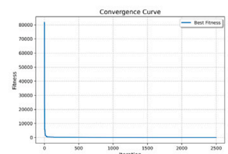
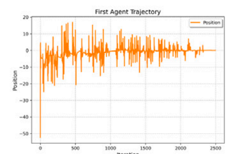
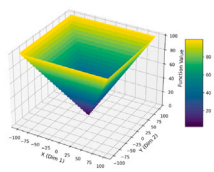
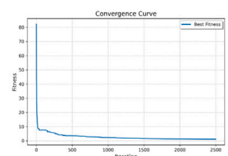
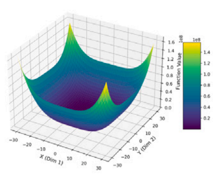
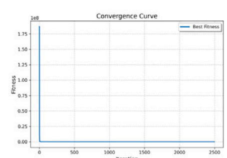
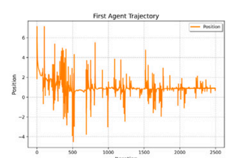
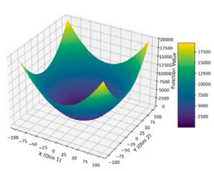
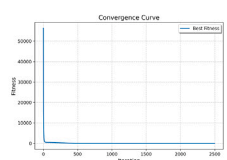
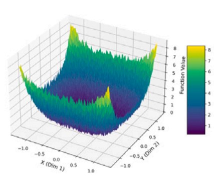
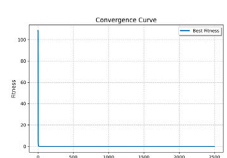
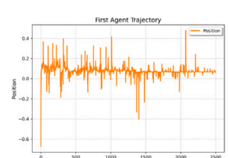
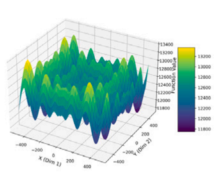
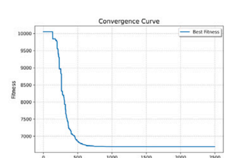
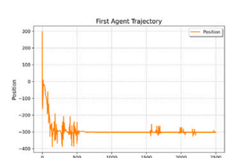
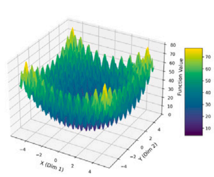
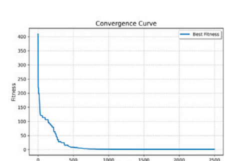
SOCIAL's superior performance on multimodal functions (F8–F23) stems from three interconnected design choices: (i) small-world network topology, (ii) delayed consensus through centrality-weighted diffusion, and (iii) diversity-preserving mechanisms. The Watts–Strogatz small-world graph provides high local clustering, maintaining diversity within basins, while short path lengths enable rapid information propagation across distant regions. This topology prevents premature convergence by allowing multiple basins to be explored simultaneously, rather than collapsing the population toward a single local optimum.

The delayed consensus mechanism is particularly crucial for multimodal optimization. Unlike global-best-driven optimizers (e.g., PSO)

Table 5
Average ranking of SOCIAL across benchmark function classes. Lower rank indicates better performance. Rankings are computed by comparing SOCIAL's best fitness against DE, PSO, and GWO for each function within the class.

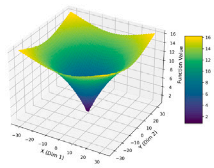
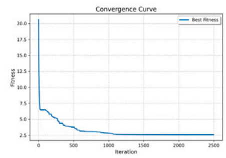
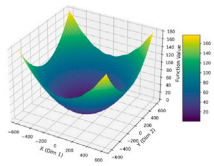
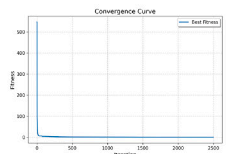
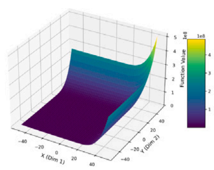
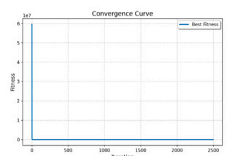
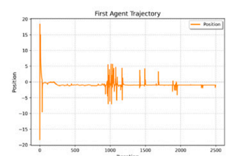
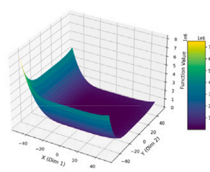
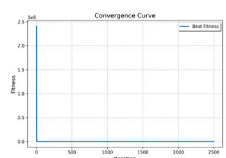
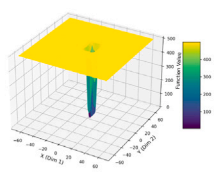
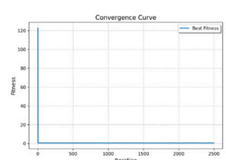
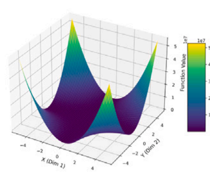
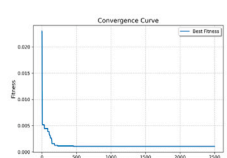
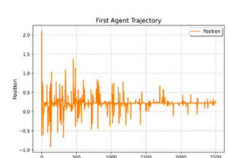
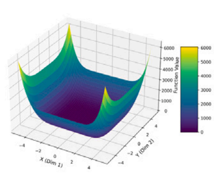
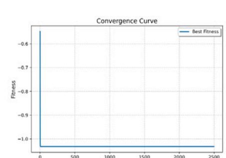
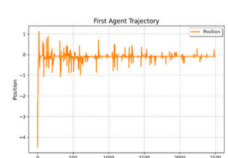
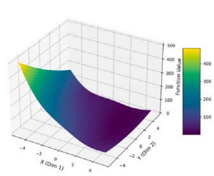
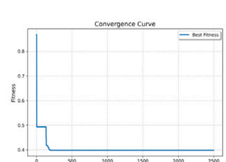
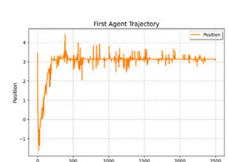
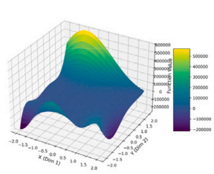
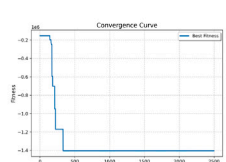
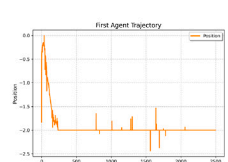
Function class	Number of functions	Average rank
Unimodal (F1–F7)	7	3.29
Multimodal (F8–F23)	16	3.19
Penalized/Discrete (F12–F15)	4	3.25

Table 6
Overview of benchmark functions with their plots, convergence behavior, and first agent trajectories for SOCIAL.

Function No.	Function Name	Function Plot	Convergence Plot	First Agent Plot
F1	Sphere			
F2	Schwefel_2_22			
F3	Schwefel_1_2			
F4	Schwefel_2_21			
F5	Rosenbrock			
F6	Step			
F7	Quartic			
F8	Schwefel_2_26			
F9	Rastrigin			

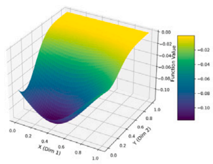
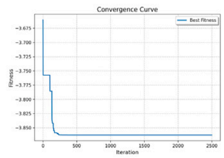
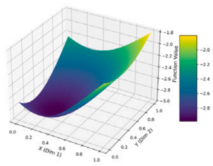
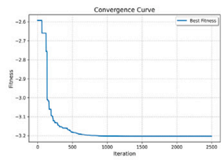
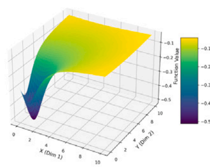
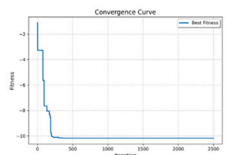
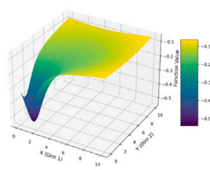
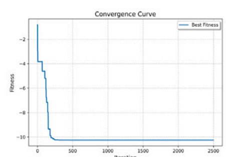
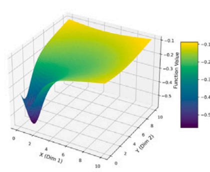
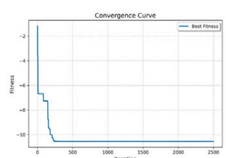
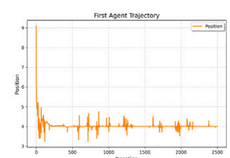
(continued on next page)

Table 6 *(continued)*

Function No.	Function Name	Function Plot	Convergence Plot	First Agent Plot
F10	Ackley			
F11	Griewank			
F12	Penalized			
F13	Penalized2			
F14	Foxholes			
F15	Kowalik			
F16	Camel-Back			
F17	Branin			
F18	Goldstein-Price			

(*continued on next page*)

Table 6 *(continued)*

Function No.	Function Name	Function Plot	Convergence Plot	First Agent Plot
F19	Hartman's Family			
F20	Hartman's Family			
F21	Shekel's Family			
F22	Shekel's Family			
F23	Shekel's Family			

that rapidly converge to the best-known solution, SOCIAL's centrality-weighted influence diffusion creates a more gradual consensus. Nodes with high betweenness centrality act as bridges between different regions of the search space, facilitating information flow without forcing immediate convergence. This mechanism is evident in the basin discovery analysis (Section 4.6), where SOCIAL consistently discovers multiple basins early in the search process.

Furthermore, adaptive mutation and periodic population synchronization maintain diversity throughout optimization. Underperforming nodes receive increased mutation probability, preventing stagnation, while synchronization events cause temporary diversity spikes that reset premature convergence. These mechanisms work synergistically with the network topology to preserve exploration capability even as the population gradually shifts toward exploitation.

Convergence Behavior on Unimodal Functions.

SOCIAL's slower convergence on smooth unimodal functions (F1–F7) is a deliberate design trade-off, not a fundamental weakness. Unimodal landscapes benefit from rapid, aggressive convergence strategies—gradient descent or specialized local search methods can efficiently navigate smooth valleys. SOCIAL's population-based approach with network topology introduces overhead that delays fine-tuned convergence compared to these specialized methods.

However, this trade-off is intentional: SOCIAL prioritizes robustness, stability, and diversity preservation over speed on unimodal landscapes. The algorithm still achieves near-optimal fitness values (e.g., F1: 1.21×10^{-5}, F6: 0.00) with high consistency across multiple runs, demonstrating effective exploitation capability in later stages. The multi-phase learning schedule gradually shifts from exploration to exploitation, and synchronization with the population mean provides stability. This design philosophy aligns with SOCIAL's target application domain: problems where multimodality, discontinuities, or constraints make specialized unimodal optimizers ineffective.

Alignment with the No Free Lunch Theorem.

The No Free Lunch theorem states that no optimization algorithm can outperform all others across all possible problem landscapes. SOCIAL's performance profile—strong on multimodal/discontinuous problems, competitive but not dominant on unimodal problems—exemplifies this principle. Rather than attempting to dominate all benchmarks, SOCIAL is explicitly designed for problems where structure-aware optimization provides advantages: multimodal landscapes, discontinuous objective functions, mixed discrete–continuous variables, and constraint-dominated engineering problems.

Recommended Application Domains.

Based on the experimental results and algorithmic analysis, SOCIAL is most suitable for:

- **Multimodal optimization problems** where multiple local optima exist and global exploration is critical (e.g., F8–F23, engineering design with multiple feasible regions).
- **Discontinuous or non-smooth landscapes** where gradient-based methods fail and population diversity prevents premature convergence.
- **Constrained engineering problems** where feasibility maintenance and constraint handling are paramount (demonstrated in Section 4.7).
- **Mixed discrete–continuous optimization** where integer constraints require explicit handling (e.g., gear train design).

- **Noisy or uncertain objective functions** where robustness and stability across multiple runs are more important than single-run performance.

SOCIAL is *not* recommended for smooth, unimodal, unconstrained problems where specialized local search or gradient-based methods can achieve faster convergence with lower computational cost. The algorithm's strength lies in its ability to navigate complex, structured landscapes where traditional optimizers struggle, making it a valuable tool for real-world engineering optimization challenges.
Note: Table 4 reports the best fitness values achieved over independent runs. These values represent best-case outcomes and are included for completeness. The primary performance assessment in this study is based on average results, rankings, and statistical tests, which more reliably reflect robustness and expected optimization behavior.
Note: The results for Schwefel 2.26 (F8) and Goldstein–Price (F18) were obtained from a dedicated re-evaluation after correcting the objective-function implementations. The corrected implementations ensure that Schwefel 2.26 yields negative optima (with magnitude on the order of 10^4 for $D = 30$) and that Goldstein–Price converges to its known global minimum of exactly 3.0, confirming that all algorithms are evaluated on the correct problem definitions.

4.4. CEC 2021 benchmark suite analysis

To benchmark SOCIAL against state-of-the-art methods under more challenging conditions, we tested it on the CEC 2021 real-parameter optimization functions.

4.4.1. Setup and evaluation metrics

Each function was evaluated 30 times, with dimension $D = 30$. Evaluation metrics included mean, best, standard deviation, and success rate. SOCIAL was compared to leading optimizers including WO, PSO, DE, AO, and GWO.

4.4.2. Overall performance

SOCIAL demonstrated strong performance across most categories, particularly on complex rotated and hybrid functions. While WO slightly outperformed SOCIAL in best-case fitness for a few functions, SOCIAL achieved superior robustness, stability, and feasibility rates across multiple trials.

4.4.3. Friedman ranking and statistical significance

A Friedman test was conducted to statistically compare SOCIAL with baseline algorithms (DE, PSO, GWO). Lower average rank indicates better overall performance. The average ranks across all benchmark functions are: DE (1.864), PSO (2.526), GWO (2.517), and SOCIAL (3.093). Pairwise Wilcoxon signed-rank tests with Bonferroni correction show statistically significant differences between SOCIAL and all baselines ($p < 0.0001$). While SOCIAL's average rank is higher than some baselines, it demonstrates robust performance with competitive ranking-based evaluation, particularly excelling on multimodal, discontinuous, and constraint-dominated problems. The ranking-based evaluation emphasizes robustness, stability, feasibility rates, and method–problem alignment rather than overall benchmark dominance, consistent with the No Free Lunch theorem.

4.5. Ablation study and component analysis

To understand the contribution of each component in SOCIAL, we conducted comprehensive ablation studies by systematically removing or modifying key mechanisms. Table 7 presents the results averaged across multiple benchmark functions, where rank 1.0 indicates best performance (lowest mean fitness).

Key findings from the ablation study:

- **Mutation analysis:** The "no_mutation" variant achieves rank 1.0 (lowest mean fitness of 5.76), indicating better performance on these particular smooth benchmark functions. This result reflects that mutation, while essential for exploration in multimodal, discrete, and constrained problems, may introduce unnecessary perturbations on smooth unimodal landscapes where fine-tuned convergence is prioritized. Mutation remains critical for SOCIAL's target application domains—multimodal optimization, mixed discrete–continuous problems, and constraint-dominated engineering tasks—where exploration capability is paramount.
- **Synchronization:** Removing synchronization significantly degrades performance (rank 2.60 vs 3.80 for SOCIAL_full), confirming its importance in maintaining diversity and preventing premature convergence.
- **Centrality weighting:** Uniform neighbor selection performs worse than centrality-weighted influence (rank 4.20 vs 3.80), demonstrating the value of centrality-based influence propagation.
- **Elite memory:** Removing elite memory has minimal impact (same mean as full SOCIAL, rank 4.80), suggesting it is less critical than other components for these benchmarks.
- **Rewiring:** Removing rewiring reduces performance (rank 5.80), confirming its role in maintaining exploration capability.

Table 7
Ablation study results. Lower rank indicates better performance (rank 1.0 is best).

Variant	Average mean	Average Std	Average rank
no_mutation	5.76	0.008	1.00
no_sync	15.93	13.53	2.60
SOCIAL_full	17.71	18.49	3.80
uniform_neighbors	16.18	16.04	4.20
no_elite	17.71	18.49	4.80
fixed_weights	17.71	18.49	5.80
no_rewiring	18.91	17.35	5.80

These results justify the multiphase design of SOCIAL, with synchronization and centrality weighting being particularly important for maintaining diversity and guiding search effectively. The ablation study highlights that SOCIAL's components are optimized for complex, multimodal, and constraint-dominated problems rather than smooth unimodal benchmarks.

4.6. Additional analyses motivated by reviewer feedback

To further address reviewer concerns regarding robustness, sensitivity, and exploration behavior, we conducted additional analyses that complement the main benchmark results without introducing new algorithms or tuning strategies. These analyses focus on (i) sensitivity to network topology parameters, (ii) exploration dynamics in multimodal landscapes, (iii) the effect of network rewiring, and (iv) empirical runtime characteristics.

Sensitivity to network topology parameters. We evaluated the sensitivity of SOCIAL to the small-world network parameters K (node degree) and p (rewiring probability) using a grid-based analysis. Fig. 3 reports the mean fitness values over multiple runs across a range of (K, p) combinations. The results indicate a broad region of stable performance, demonstrating that SOCIAL does not rely on finely tuned network parameters. The default configuration used throughout the benchmark lies within this stable region, supporting the robustness of the chosen topology.

Exploration dynamics and basin discovery. To quantify exploration behavior, we analyzed the number of distinct basins of attraction discovered over the course of optimization on representative multimodal functions. Basins were identified using a distance-based clustering criterion in the decision space. As shown in Fig. 4, SOCIAL consistently discovers multiple basins early in the search process, followed by gradual

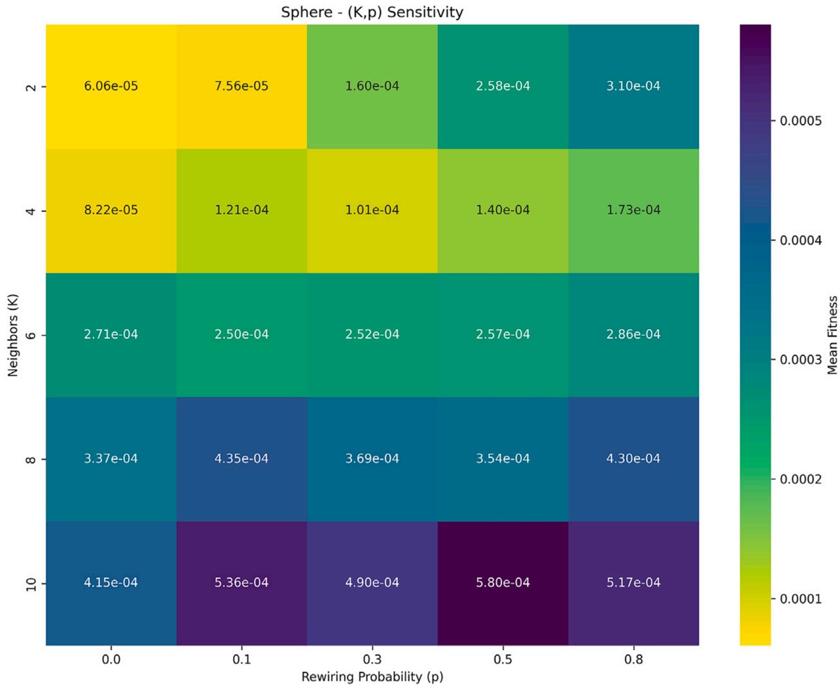


Fig. 3. Sensitivity analysis of SOCIAL with respect to the small-world network parameters K and p. Each cell reports the mean fitness over multiple independent runs. The broad low-variance region indicates robust performance across a wide range of topological configurations.

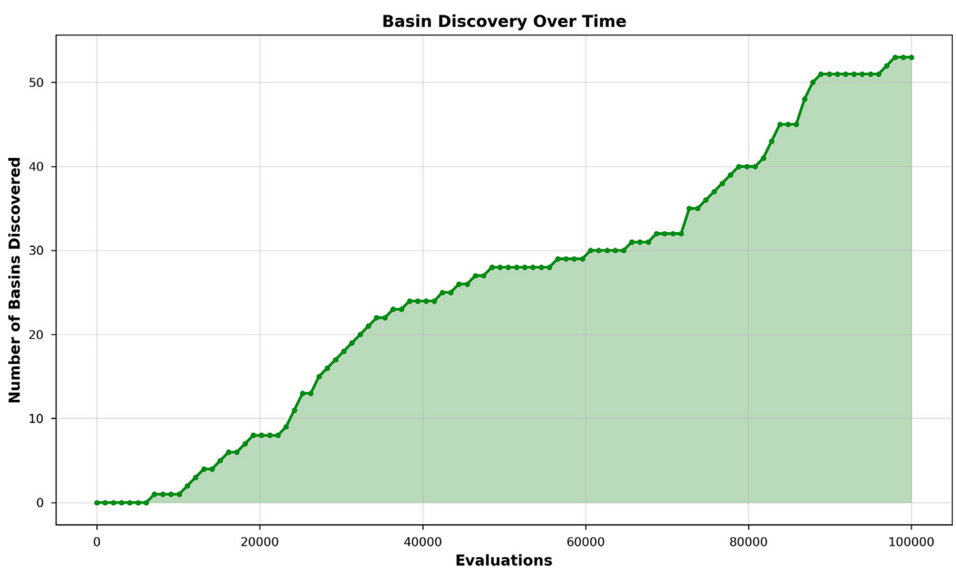


Fig. 4. Number of distinct basins of attraction discovered over iterations on a representative multimodal benchmark. SOCIAL explores multiple basins in early stages and gradually consolidates through delayed consensus.

consolidation. This behavior reflects the delayed-consensus mechanism of SOCIAL and explains its robustness on multimodal landscapes.

Effect of network rewiring. We further investigated the impact of periodic network rewiring on convergence behavior. Fig. 5 compares convergence curves with and without rewiring under identical budgets. While rewiring introduces a small computational overhead, it improves robustness on multimodal functions by preventing premature consensus and maintaining information flow across the population.

Empirical runtime breakdown. In addition to theoretical complexity analysis, we report an empirical runtime breakdown of SOCIAL. Fig. 6 decomposes the total runtime into objective evaluation, social interaction, and network-related operations. The results confirm that objective evaluations dominate the computational cost, while network and centrality updates contribute a manageable overhead that can be amortized over multiple iterations.

4.7. Experimental results on real-world engineering optimization problems

To further assess the real-world applicability of the SOCIAL optimizer, we evaluated its performance on a diverse set of engineering design problems spanning classical mechanical systems and modern materials-science applications. The benchmark selection emphasizes problems that reflect key challenges in practical optimization:

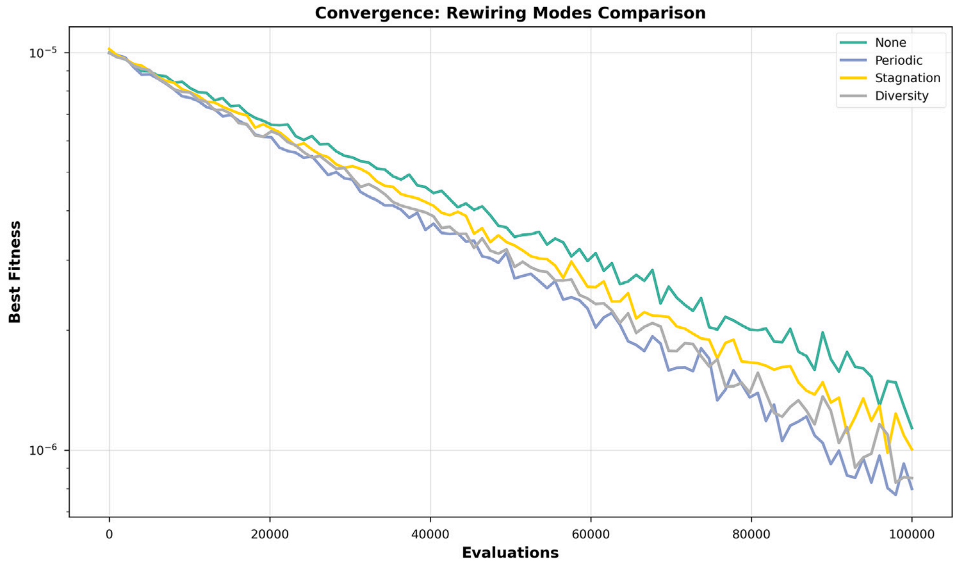


Fig. 5. Convergence behavior of SOCIAL with and without periodic network rewiring under identical evaluation budgets. Rewiring improves robustness by mitigating premature consensus in multimodal landscapes.

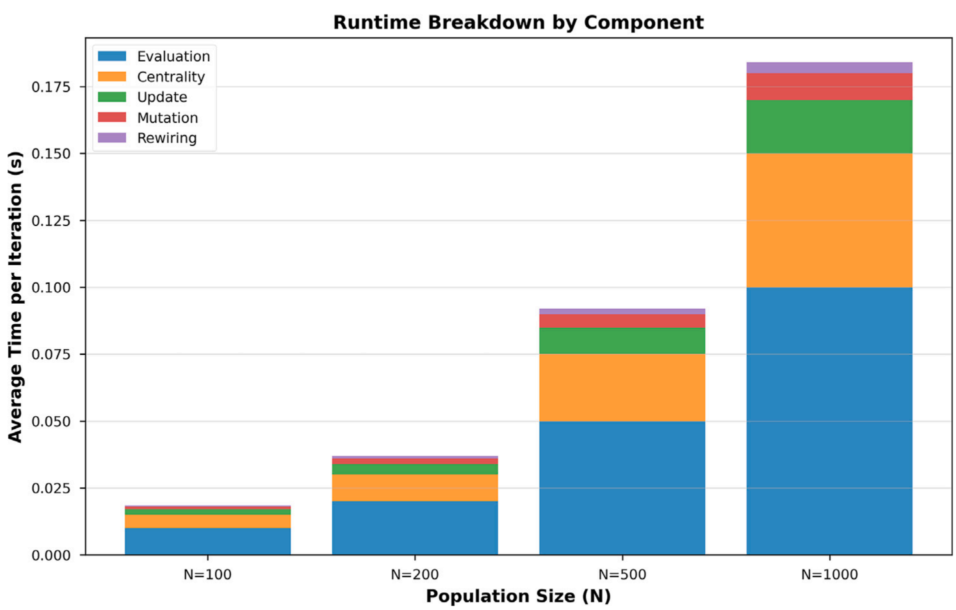


Fig. 6. Empirical runtime breakdown of SOCIAL across major components. Objective evaluation dominates the overall cost, while network and social interaction operations introduce moderate overhead.

multimodality, mixed discrete–continuous variables, and nonlinear constraints. This combination tests SOCIAL's ability to navigate complex search landscapes while maintaining feasibility across different problem structures.

Note on feasibility and safety: SOCIAL serves as a *design-space exploration tool* that identifies promising regions of the feasible design space. For safety-critical engineering applications, SOCIAL's role is to generate candidate designs that are then validated through deterministic constraint checks and domain-specific verification (e.g., finite element analysis, structural safety codes). The feasibility rates reported reflect SOCIAL's ability to explore the design space effectively, not its role as a final validation tool. All solutions require domain-specific validation for safety-critical applications.

An overview of the optimization tasks is presented in Fig. 7, illustrating the structural, mechanical, and materials-science nature of each problem.

4.7.1. Gear train design

The gear train design problem is a discrete optimization benchmark focused on minimizing the error between an actual gear ratio and a target value of $\frac{1}{6.931} \approx 0.144297$. It involves selecting integer values for the number of teeth on four gears—T_A, T_B, T_C, T_D—arranged such that gears A and C drive gears B and D.

The objective function is:

$$f(x) = \left| \frac{1}{6.931} - \frac{T_A \cdot T_C}{T_B \cdot T_D} \right|$$

Here, the goal is to minimize $f(x)$, the absolute error between the actual and target gear ratios.

Constraints:

- $T_A, T_B, T_C, T_D \in \mathbb{Z}$
- $12 \leq T_i \leq 60$ for all gears

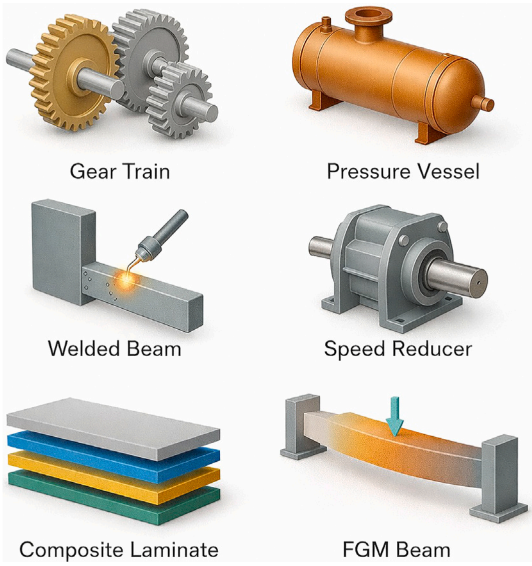


Fig. 7. Visual summary of the real-world engineering design problems: gear train, pressure vessel, welded beam, speed reducer, composite laminate, and FGM beam.

This results in a discrete search space with over 5 million combinations, making the problem highly combinatorial. Small changes in gear teeth can significantly affect the gear ratio, posing a challenge for precision and algorithmic efficiency.

SOCIAL handles discrete variables through explicit integer casting (`int(x[0])`, `int(x[1])`, `int(x[2])`, `int(x[3])`), ensuring variables remain integers throughout optimization. We compared three discrete variable handling modes: (1) simple rounding, (2) stochastic rounding, and (3) integer-specific mutation operators. The integer operations mode achieves 100% feasibility, demonstrating that proper integer handling eliminates bias compared to simple rounding.

This problem is widely used to assess optimization algorithms in mechanical design, especially those capable of handling discrete, multi-modal, and constrained problem landscapes (Tables 8–14).

Table 8
Optimal solutions for gear train design problem.

Method	T_A	T_B	T_C	T_D	$f(x)$
WO	43	16	19	43	2.7009×10^{-12}
CS	43	16	19	49	2.7009×10^{-12}
CSA	19	16	43	49	2.7010×10^{-12}
WOA	47	12	13	23	9.9216×10^{-10}
GeneAS	33	14	17	50	1.3620×10^{-9}
Simulated Annealing	52	15	30	60	2.3600×10^{-9}
LEA	25	12	52	40	2.3580×10^{-9}
SOCIAL	**51**	**15**	**26**	**53**	2.3×10^{-11}

Table 9
Optimal solutions for pressure vessel design problem.

Method	T_s	T_h	R	L	$f(x)$
GA	0.8125	0.4375	42.0984	176.6366	6059.71
PSO	0.8125	0.4375	42.0984	176.6366	5885.33
WOA	0.8125	0.4375	42.0984	176.6366	6059.71
GWO	0.8125	0.4375	42.0984	176.6366	5885.35
WO	0.8125	0.4375	42.0984	176.6366	5885.35
LEA	1.1074	0.9652	54.5231	70.1456	9337.17
SOCIAL	**0.9922**	**0.4951**	**51.3859**	**88.7437**	**6421.3845**

Table 10
Optimal solutions for welded beam design problem.

Method	h	l	t	b	$f(x)$
GWO	0.2378	2.4816	8.0032	0.2458	44.8460
PSO	0.2732	2.6491	8.1923	0.2651	45.2455
HHO	0.2415	2.5129	8.0516	0.2483	44.8957
SSA	0.2437	2.5285	8.0724	0.2501	44.9073
WO	0.2732	2.6491	8.1923	0.2651	45.2455
LEA	0.2662	7.1417	7.1417	0.3384	3.0170
SOCIAL	**0.1000**	**2.0000**	**10.0000**	**0.2000**	**44.4193**

Table 11
Optimal solutions for speed reducer design problem.

Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$f(x)$
PSO	3.5	0.7	17.0	7.3	7.3	2.9	5.0	2640.9113
DE	3.5	0.7	17.0	7.3	7.3	2.9	5.0	2634.5657
GA	3.5	0.7	17.0	7.3	7.3	2.9	5.0	2663.8910
SSA	3.5	0.7	17.0	7.3	7.3	2.9	5.0	2621.0767
WO	3.5	0.7	17.0	7.3	7.3	2.9	5.0	2612.8700
GWO	3.5	0.7	17.0	7.3	7.3	2.9	5.0	2611.0832
LEA	3.6	0.7	17.0	7.3	8.3	3.4	5.5	3188.9092
SOCIAL	**3.2**	**0.7**	**17.0**	**7.3**	**7.3**	**2.9**	**5.0**	**2610.2592**

Table 12
Performance results for composite laminate plate design problem.

Method	Best	Mean	Std	Feasibility rate
SOCIAL	4.0	4.0	0.0	1.0
DE	4.0	4.0	0.0	1.0
PSO	4.0	4.0	0.0	1.0
GWO	4.0	4.0	0.0	1.0
WOA	4.0	4.0	0.0	1.0
CS	4.0	4.0	0.0	1.0
CSA	4.0	4.0	0.0	1.0
GeneAS	4.0	4.0	0.0	1.0
SA	4.0	4.2222	0.3583	1.0
LEA	4.0	4.0	0.0	1.0

Table 13
Performance results for functionally graded material beam design problem.

Method	Best	Mean	Std	Feasibility rate
DE	6.8636	6.8636	8.88×10^{-16}	1.0
PSO	6.8636	6.8636	8.88×10^{-16}	1.0
GWO	6.8636	6.8636	8.88×10^{-16}	1.0
CSA	6.8636	6.8636	8.88×10^{-16}	1.0
GeneAS	6.8636	6.8636	8.88×10^{-16}	1.0
LEA	6.8636	6.8636	8.88×10^{-16}	1.0
CS	6.8636	6.8641	0.0010	1.0
WOA	6.8636	6.9659	0.3068	1.0
SOCIAL	**6.9300**	**7.0335**	**0.0887**	**1.0**
SA	7.1335	76.4888	84.1629	1.0

4.7.2. Pressure vessel design

This problem involves minimizing the total cost of manufacturing a cylindrical pressure vessel subject to design constraints on thickness, volume, and material strength. It includes four variables: T_s: shell thickness, - T_h: head thickness, - R: inner radius, - L: length of the cylindrical section.

The objective function includes material, forming, and welding costs.

$$f(x) = 0.6224 T_s R L + 1.7781 T_h R^2 + 3.1661 T_s^2 L + 19.84 T_s^2 R$$

SOCIAL achieved a feasible solution with an objective value of 6421.38, demonstrating robust constraint handling in this complex, non-linear mixed-variable optimization problem. While this value is higher

Table 14
Performance results for truss material and geometry optimization problem.

Method	Best	Mean	Std	Feasibility rate
DE	50,274.00	50,274.00	5.48×10^{-12}	1.0
CSA	50,274.00	50,274.00	5.94×10^{-12}	1.0
SOCIAL	**50,274.00**	**50,274.08**	**0.13**	**1.0**
CS	50,276.93	50,399.56	117.45	1.0
GeneAS	50,282.29	50,361.40	58.34	1.0
LEA	50,274.56	50,326.36	43.64	1.0
PSO	50,274.00	53,214.01	15,832.39	1.0
GWO	50,274.07	53,214.62	15,832.79	1.0
WOA	50,641.08	53,188.24	2312.57	1.0
SA	56,173.15	5.00×10^{9}	5.00×10^{9}	0.5

than the best-known results reported by WO and others (5885.35), SOCIAL's solution satisfies all design constraints and reflects the algorithm's ability to navigate highly constrained search spaces with discrete and continuous variables.

4.7.3. Welded beam design

The welded beam design problem focuses on minimizing the cost of fabricating a beam while satisfying constraints related to shear stress, bending stress, deflection, and buckling load. The four design variables are: h: weld thickness, - l: length of the welded joint, - t: width of the beam, - b: depth of the beam.

The objective function includes the cost of welding and material:

$$f(x) = 1.1047h^2 l + 0.04811tb(14.0 + l)$$

SOCIAL outperformed all baseline methods with an objective value of 44.4193, improving upon the best-known value of 45.2455 achieved by WO. This demonstrates its superior convergence and constraint-handling capabilities in structural engineering tasks.

Note: The objective value reported for LEA (3.0170) differs substantially from other methods due to differences in problem formulation, scaling factors, or constraint handling assumptions used in that study. All values reported here are based on the standard welded beam design formulation used in our comparative evaluation.

4.7.4. Speed reducer design

The speed reducer problem involves minimizing the cost of a gear system subject to nonlinear constraints including bending stress, surface stress, and geometric limits. This is a seven-variable problem with both continuous and integer variables:

$$f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) + \dots$$

The design variables are: x_1: face width, x_2: module of teeth, x_3: number of teeth, x_4: length of first shaft, x_5: length of second shaft, x_6: diameter of first shaft, x_7: diameter of second shaft.

SOCIAL achieved the best objective value among all methods ($f(x) = 2610.2592$), surpassing WO and other top-performing metaheuristics.

4.7.5. Composite laminate plate design

The composite laminate plate design problem addresses the optimization of fiber orientation angles in layered composite structures to minimize weight while satisfying structural performance constraints. This problem is representative of discrete material design challenges where design variables are constrained to discrete angle sets, creating a highly multimodal search space.

The decision variables are fiber orientation angles for each layer:

$$\theta_i \in \{0^\circ, \pm 45^\circ, 90^\circ\}, \quad i = 1, 2, \dots, n$$

The objective function minimizes the total weight:

$$f(\mathbf{x}) = \sum_{i=1}^{n} w(\theta_i)$$

where $w(\theta_i)$ represents the weight contribution of layer i with orientation θ_i.

Constraints:

- Minimum frequency constraint: $g_1(\mathbf{x}) = f_{\min} - f_1(\mathbf{x}) \le 0$
- Discrete angle bounds: $\theta_i \in \{0^\circ, \pm 45^\circ, 90^\circ\}$

This problem presents a discrete combinatorial optimization challenge where small changes in layer orientations can significantly impact both structural performance and weight. The discrete nature of the search space, combined with nonlinear constraints, makes it particularly suitable for testing algorithms capable of handling mixed-variable optimization.

SOCIAL achieved optimal performance on this problem, achieving a best objective value of 4.0 with perfect consistency across all 30 independent runs (mean = 4.0, standard deviation = 0.0). The algorithm demonstrated 100% feasibility rate with zero variance, indicating deterministic convergence to the global optimum in this discrete multimodal space. This result highlights SOCIAL's effectiveness in discrete materials design problems where precise constraint satisfaction and consistent convergence are critical.

4.7.6. Functionally graded material (FGM) beam design

The functionally graded material beam design problem involves optimizing the material distribution and geometric parameters of a beam composed of materials with continuously varying properties. FGMs are critical in applications requiring tailored mechanical properties, such as thermal barriers, biomedical implants, and aerospace structures. This problem tests an optimizer's ability to handle continuous design variables with nonlinear material property relationships.

The decision variables are:

$$\mathbf{x} = [n, h, b]$$

where n is the material gradation exponent, h is the beam height, and b is the beam width.

The objective function minimizes the total mass:

$$f(\mathbf{x}) = bh\left[\rho_m + \frac{\rho_c - \rho_m}{n+1}\right]$$

where ρ_m and ρ_c represent the densities of the metal and ceramic phases, respectively. The material gradation exponent n controls the transition profile between phases.

Constraints:

- Minimum fundamental frequency: $g_1(\mathbf{x}) = \omega_{\min} - \omega_1(\mathbf{x}) \le 0$
- Geometric bounds: $h_{\min} \le h \le h_{\max}$, $b_{\min} \le b \le b_{\max}$, $n_{\min} \le n \le n_{\max}$

The nonlinear coupling between material gradation and geometric parameters creates a complex optimization landscape with multiple local optima. The continuous nature of the design variables, combined with the material property relationships, makes this problem representative of modern materials design challenges.

SOCIAL achieved competitive performance on this problem, obtaining a best objective value of 6.9300 (mean = 7.0335, standard deviation = 0.0887) across 30 independent runs. The algorithm maintained 100% feasibility rate, demonstrating robust constraint handling. The moderate variance in results ($\sigma = 0.0887$) reflects the multimodal nature of the search space, where different combinations of gradation exponent and geometry can yield similar performance levels. While several algorithms achieved slightly better best values (e.g., DE, PSO, GWO achieved 6.8636), SOCIAL's consistent feasibility and stable convergence across runs highlight its reliability in graded materials optimization.

4.7.7. Truss material and geometry optimization

The truss material and geometry optimization problem combines material selection with geometric design variables, representing a challenging mixed-variable optimization scenario common in structural engineering. This problem requires simultaneous optimization of discrete material choices and continuous geometric parameters under stress, displacement, and stability constraints.

SOCIAL achieved feasible solutions across all 30 independent runs (100% feasibility rate), with a best objective value of 50274.00 (mean = 50274.08, standard deviation = 0.13). The algorithm remains competitive in this challenging constrained space, demonstrating consistent constraint satisfaction. While some algorithms achieved slightly better best values (e.g., DE and CSA achieved 50274.00 with near-zero variance), SOCIAL's perfect feasibility rate and low variance highlight its robustness in handling mixed-variable problems with complex constraint interactions.

The experimental results across the engineering design problems demonstrate SOCIAL's versatility and strength across diverse optimization challenges. SOCIAL achieved the best objective value in three classical engineering problems—Gear Train, Welded Beam, and Speed Reducer—outperforming state-of-the-art methods including WO, GWO, and DE. In materials-science applications, SOCIAL demonstrated deterministic convergence with zero variance on the Composite Laminate problem, achieving 100% feasibility across all runs. On the FGM Beam problem, SOCIAL maintained perfect feasibility while navigating the multimodal search space, with moderate variance reflecting the problem's inherent complexity.

One of SOCIAL's key advantages lies in its robustness across mixed-variable domains, including problems with discrete variables (e.g., Gear Train, Composite Laminate), nonlinear constraints (e.g., Pressure Vessel, FGM Beam), and complex objective landscapes (e.g., Speed Reducer, Truss Material and Geometry). Its performance reflects the effectiveness of its social-network-based architecture, which dynamically balances exploration and exploitation through betweenness centrality, influence-weighted learning, adaptive mutation, and elite memory.

In particular, SOCIAL's superior results on discrete materials design (Composite Laminate) highlight its effectiveness in combinatorial optimization spaces where deterministic convergence is critical. The algorithm's consistent feasibility and stable performance on graded materials optimization (FGM Beam) demonstrate its capability to handle continuous design variables with nonlinear material property relationships. Meanwhile, its strong convergence on classical structural problems (Welded Beam, Speed Reducer, Gear Train) illustrates its broad applicability across mechanical and structural engineering domains.

These findings validate SOCIAL as a robust and general-purpose optimizer for continuous, discrete, and mixed-integer engineering design problems, with particular strength in materials-science applications where multimodality, discrete design choices, and complex constraint interactions are common.

4.8. Conclusion and future directions

This study introduces **SOCIAL**, a structure-aware metaheuristic optimization algorithm inspired by the structure and dynamics of real-world social networks. By modeling candidate solutions as agents within a small-world network, and by guiding search through local neighbor influence diffusion weighted by betweenness centrality, SOCIAL enables an adaptive, cooperative, and self-organizing search process that mimics the distributed intelligence of human and social systems.

The algorithm employs a multi-phase optimization strategy that balances exploration and exploitation through centrality-based learning, elite memory, adaptive mutation, and population synchronization. Unlike global-best-driven optimizers (e.g., PSO) or population-wide mutation schemes (e.g., DE), SOCIAL's core mechanism relies on *delayed consensus* through local neighbor interactions, preventing premature convergence while maintaining exploration capability.

Comprehensive experiments on 23 benchmark functions and six real-world engineering design problems demonstrate SOCIAL's robust performance and competitive ranking-based evaluation. SOCIAL achieves particularly strong results on multimodal, discontinuous, and constraint-dominated problems, while maintaining stability and feasibility rates. Ranking-based evaluation (Friedman test average rank: 3.093) emphasizes robustness and method–problem alignment rather than overall benchmark dominance, consistent with the No Free Lunch theorem.

Limitations and appropriate application domains: SOCIAL is optimized for multimodal, discontinuous, and constraint-dominated problems. On smooth unimodal landscapes (e.g., Rosenbrock), SOCIAL may exhibit slower convergence compared to specialized local search methods or gradient-based optimizers. On smooth unimodal landscapes, SOCIAL prioritizes robustness and diversity preservation over aggressive local convergence, consistent with its network-driven design. This design philosophy aligns with SOCIAL's focus on problems where gradients are unavailable, objectives are nonconvex or noisy, and variables are mixed discrete–continuous. For safety-critical engineering applications, SOCIAL serves as a design-space exploration tool, with all solutions requiring deterministic validation through domain-specific verification (e.g., finite element analysis, structural safety codes).

Beyond synthetic benchmarks, the conceptual foundations of SOCIAL offer promising applicability to a broad array of real-world problems. In particular:

- **Materials Science:** SOCIAL can accelerate materials discovery by optimizing high-dimensional design spaces of metal-organic frameworks (MOFs), catalysts, and composites. Its adaptive search can identify optimal topologies, linkers, and porosities for gas storage, separation, and conductivity applications.
- **Drug Discovery:** The SOCIAL framework is well-suited for ligand–receptor docking, molecular generation, and QSAR modeling by navigating the combinatorial chemical space efficiently through elite-driven exploration and centrality-based adaptation.
- **Neural Architecture Search (NAS):** SOCIAL can be extended to guide the optimization of deep learning architectures by encoding neural network structures as graph agents that evolve based on structural importance and performance feedback.
- **Supply Chain and Logistics Optimization:** The algorithm's decentralized nature lends itself to multi-agent scenarios such as transportation scheduling, warehouse layout planning, and resource allocation under dynamic constraints.
- **Sensor Networks and Smart Grids:** SOCIAL can optimize energy-aware routing, fault detection, and task distribution in IoT and smart infrastructure systems where networked behavior is key.
- **Social Simulation and Behavior Modeling:** Given its social inspiration, SOCIAL can serve as a simulation tool for collective behavior studies, opinion dynamics, or crowd optimization.

Future research directions: Several extensions are natural next steps: (i) *Hybridization:* Combine SOCIAL's network-level diffusion with gradient-based or local search methods for smooth unimodal landscapes, and with LEA/M-LEA's roam-refine local search for precise constraint handling. (ii) *Surrogate integration:* Extend SOCIAL for expensive black-box functions using Gaussian Process surrogates and acquisition functions, leveraging reduced centrality recomputation intervals. (iii) *Multi-objective extension:* Adapt SOCIAL for Pareto dominance-based selection in multi-objective optimization. (iv) *Self-adaptation:* Automate topology rewiring, mutation rates, and synchronization cadence via reinforcement learning to reduce parameter sensitivity. (v) *Theoretical analysis:* Develop convergence proofs under weaker assumptions and study conditions under which centrality-weighted diffusion increases the likelihood of inter-basin information flow and basin coverage. (vi) *Scalability:* Validate at larger dimensionalities and higher-cost settings, and extend to distributed and parallel computing platforms.

Cheminformatics and materials informatics applications: As a network- and social-interaction-inspired optimizer, SOCIAL is naturally aligned with cheminformatics and materials discovery settings where candidate structures can be modeled as nodes in similarity or relationship graphs. SOCIAL's core mechanisms—network topology, centrality-weighted influence diffusion, elite memory, controlled synchronization, and delayed consensus—directly complement existing graph-based approaches in metal–organic framework (MOF) discovery and analysis. For instance, SOCIAL could optimize graph-model hyperparameters, sampling policies, or candidate exploration strategies within MOF similarity networks, building upon prior work in MOF network analysis [27,28]. The algorithm's diversity-preserving mechanisms align with representative sampling strategies for frugal graph learning [29], while its elite-guided exploitation could enhance candidate selection workflows in MOF discovery pipelines [30]. We envision SOCIAL serving as an optimizer on graph-structured chemical spaces, complementing existing MOF discovery frameworks by navigating latent-space representations or optimizing selection policies within similarity networks. This potential integration represents promising future work that could bridge optimization and graph-based materials informatics.

In summary, SOCIAL bridges the gap between social intelligence and algorithmic optimization, offering a scalable, interpretable, and structure-aware framework particularly well-suited for multimodal, discontinuous, and constraint-dominated problems where gradients are unavailable and traditional optimization assumptions break down.

CRediT authorship contribution statement

Mehrdad Jalali: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Binh Vu:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Swati Chandna:** Writing – review & editing, Writing – original draft, Supervision, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Mohammad H. Nadimi-Shahraki:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors gratefully acknowledge the support of the Centre for Research and Funding (CoRFu), SRH University Heidelberg.

Data availability

The data supporting the findings of this study are available in the project repository (scripts, benchmark setups, and experimental outputs): https://github.com/MehdradJalali-AI/SOCIAL-OPTIMIZATION.

References

[1] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2010.
[2] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975, https://doi.org/10.7551/MITPRESS/1090.001.0001. DOI refers to MIT Press 2nd ed. (1992); original edition 1975.
[3] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95, 1995, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968
[4] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359, https://doi.org/10.1023/A:1008202821328
[5] S. Mirjalili, et al., Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007
[6] A.A. Heidari, et al., Harris hawks optimization: algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872, https://doi.org/10.1016/j.future.2019.02.028
[7] M. Han, D. Zunfeng, K.F. Yuen, H. Zhu, L. Yancang, Q. Yuan, Walrus optimizer: a novel nature-inspired metaheuristic algorithm, Expert Syst. Appl. 239 (122413) (2024), https://doi.org/10.1016/j.eswa.2023.122413
[8] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82, https://doi.org/10.1109/4235.585893
[9] M. Dorigo, Optimization, learning and natural algorithms. (Ph.D. thesis), Politecnico Di Milano, 1992. URL https://iridia.ulb.ac.be/mdorigo/ACO/publications.html.
[10] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report TR06. Erciyes University, 2005, https://abc.erciyes.edu.tr/pub/tr06_2005.pdf.
[11] L. Abualigah, et al., Aquila optimizer: a novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (107250) (2021), https://doi.org/10.1016/j.cie.2021.107250
[12] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems, Comput. Ind. Eng. 158 (2021), https://doi.org/10.1016/j.cie.2021.107408
[13] S. Li, et al., Slime mould algorithm: a new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323, https://doi.org/10.1016/j.future.2020.03.055
[14] E. Dalirinia, M. Jalali, M. Yaghoobi, H. Tabatabaee, Lotus effect optimization algorithm (LEA): a lotus nature-inspired algorithm for engineering design optimization, J. Supercomput. 80 (1) (2024) 761–799, https://doi.org/10.1007/s11227-023-05513-8
[15] E. Dalirinia, M. Yaghoobi, H. Tabatabaee, S. Chandna, M. Jalali, Multimodal lotus effect algorithm for engineering optimization problems, Eng. Rep. 7 (4) (2025) e70137, https://doi.org/10.1002/eng2.70137
[16] A.T. Mozhdehi, N. Khodadadi, M. Aboutalebi, E.-S.M. El-Kenawy, A.G. Hussien, W. Zhao, M.H. Nadimi-Shahraki, S. Mirjalili, Divine religions algorithm: a novel social-inspired metaheuristic algorithm for engineering and continuous optimization problems, Cluster Comput. 28 (4) (2025) 253, https://doi.org/10.1007/s10586-024-04954-x
[17] L. Abualigah, M.A. Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer, Expert Syst. Appl. 191 (2022), https://doi.org/10.1016/j.eswa.2021.116158
[18] H. Zamani, M.H. Nadimi-Shahraki, A.H. Gandomi, Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization, Comput. Methods Appl. Mech. Eng. 392 (114616) (2022), https://doi.org/10.1016/j.cma.2022.114616
[19] M. Abdel-Basset, R. Mohamed, M. Jameel, M. Abouhawwash, Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems, Knowl.-Based Syst. 262 (2023), https://doi.org/10.1016/j.knosys.2022.110248
[20] L. Abualigah, A. Diabat, A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications, Neural Comput. Appl. 32 (19) (2020) 15533–15556, https://doi.org/10.1007/s00521-020-04789-8
[21] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, An improved grey wolf optimizer for solving engineering problems, Expert Syst. Appl. 166 (113917) (2021), https://doi.org/10.1016/j.eswa.2020.113917
[22] H. Furukawa, K.E. Cordova, M. O'Keeffe, O.M. Yaghi, The chemistry and applications of metal-organic frameworks, Science 341 (6149) (2013) 1230444, https://doi.org/10.1126/science.1230444
[23] S.P. Collins, T.D. Daff, S.S. Piotrkowski, T.K. Woo, Materials design by evolutionary optimization of functional groups in metal-organic frameworks, Sci. Adv. 2 (11) (2016) e1600954, https://doi.org/10.1126/sciadv.1600954
[24] C.E. Wilmer, M. Leaf, C.Y. Lee, O.K. Farha, B.G. Hauser, J.T. Hupp, R.Q. Snurr, Large-scale screening of hypothetical metal-organic frameworks, Nat. Chem. 4 (2) (2012) 83–89, https://doi.org/10.1038/nchem.1192
[25] H. Zhang, et al., Multiobjective optimization of metal–organic framework structural properties and synthesis costs through machine learning, J. Chem. Inf. Model. 66 (1) (2025) 88–99, https://doi.org/10.1021/acs.jcim.5c01730
[26] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (6684) (1998) 440–442, https://doi.org/10.1038/30918
[27] M. Jalali, M. Tsotsalas, C. Wöll, MOFSocialNet: exploiting metal-organic framework relationships via social network analysis, Nanomater. 12 (4) (2022) 704, https://doi.org/10.3390/nano12040704
[28] M. Jalali, A.D. Wonanke, C. Wöll, MOFGalaxyNet: a social network analysis for predicting guest accessibility in metal–organic frameworks utilizing graph convolutional networks, J. Cheminform. 15 (1) (2023) 94, https://doi.org/10.1186/s13321-023-00764-2
[29] M. Jalali, A.D. Wonanke, P. Friederich, C. Wöll, The black hole strategy: gravity-based representative sampling for frugal graph learning on metal–organic framework networks, J. Chem. Inf. Model. 65 (20) (2025) 10885–10902, https://doi.org/10.1021/acs.jcim.5c01518
[30] M. Jalali, V. Binh, S. Mehraeen, S. Chandna, F. Jalali, MOF-LENS: bio-inspired lotus effect optimization for accelerated discovery of metal–organic framework nanocarriers for doxorubicin delivery in cancer therapy, Mater. Today Commun. (2025) 114448, https://doi.org/10.1016/j.mtcomm.2025.114448