Swarm and Evolutionary Computation 62 (2021) 100848

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Robust and efficient swarm communication topologies for hostile environments

Vipul Mann, Abhishek Sivaram, Laya Das, Venkat Venkatasubramanian*

Department of Chemical Engineering, Columbia University, New York, NY 10027, USA

ARTICLE INFO

Keywords:
Particle swarm optimization
Communication networks
Global optimization
Non-convex optimization

ABSTRACT

Swarm Intelligence-based optimization techniques rely strongly on information communicated among agents during the search. These algorithms are generally used for solving challenging problems where the search function landscape, typically highly nonlinear, is not adequately known a priori and there are multiple local optima that could result in premature convergence for many algorithms. Furthermore, in certain applications, the search environment is hostile in the sense that the agents, and/or their communication channels, could get disrupted during the search, thereby adversely impacting the search performance. Such disruptions change the communication topology of the agents and hence the information available to them, ultimately influencing the performance of the algorithm. Here, we present a study of the impact of loss of agents on performance as a function of the initial network configuration. We use particle swarm optimization to optimize an objective function with multiple sub-optimal regions in a hostile environment and study its performance for a range of network topologies with loss of agents. The results reveal interesting trade-offs between efficiency, robustness, and overall performance for different topologies that are analyzed to discover general features of networks that maximize performance. We observe that small-world networks perform well under hostile conditions.

1. Introduction

Science and engineering have often drawn inspiration from nature in addressing challenging problems across fields such as complex systems engineering, active matter physics, and biomimetics towards understanding complex phenomena. Optimization has also witnessed a similar impact with a class of *nature-inspired* computing techniques [1]. These techniques model the behaviour of agents in nature such as ants and birds that work collectively towards achieving a *global* objective with *local* (agent-specific) rules and extend them towards solving optimization problems. Such approaches have given birth to techniques such as evolutionary programming, genetic algorithms, simulated annealing, and differential evolution. Swarm Intelligence [2] is one such class of nature-inspired optimization methods that makes use of interacting agents towards optimizing a collective goal. Algorithms based on swarm intelligence include, among others, particle swarm optimization [3], ant colony optimization [4], firefly algorithm [5,6], cuckoo algorithm [7], bat algorithm [8], and squirrel-search algorithm [9], which have been used in several applications [10–13]. These algorithms utilize recursive update rules drawn from natural agents and have proven application across a number of areas.

Particle Swarm Optimization (PSO) is inspired from the movement of a flock of birds and primarily works by combining an individual's cognizance and collective swarm intelligence [3]. PSO is a gradient-independent algorithm that has been identified as lying between genetic algorithm and evolutionary programming. The movement of its agents, based on individual and collective best strategy, is similar to crossover operations in genetic algorithms [14], and is dependent on stochastic processes similar to evolutionary programming [15]. PSO exhibits simple update rules and minimal set of tunable parameters along with a feedback control mechanism that make it a highly desirable choice for a diverse range of problems including– parameter estimation [16], dynamic optimization [17,18], forecasting properties of interest [19], clustering [20], and training feed-forward neural networks [21]. PSO has also been used in several applications including robotics [22], placement of distributed generators in smart grid [23], astronomy [24], manufacturing [25], game theory [26], and several more [27].

The majority of applications of PSO are characterized by a lack of complete knowledge of the function landscape, and at times an analytical formulation of the objective function. There have been several theoretical advances [28] and algorithmic modifications to the standard PSO algorithm since its introduction. These include the importance of choosing adaptive inertia weights [29] and the impact of maximum velocity on the algorithm's performance [30]. Furthermore, PSO has been modified and generalized to incorporate constraints that arise in several optimization problems in practice [31]. Extensions of the algorithm to multi-

* Corresponding author.
E-mail address: venkat@columbia.edu (V. Venkatasubramanian).

https://doi.org/10.1016/j.swevo.2021.100848
Received 19 March 2020; Received in revised form 5 October 2020; Accepted 28 January 2021
Available online 15 February 2021

objective optimization [32] and hybridization approaches [33] have also been proposed in the literature.

The impact of communication topologies between agents in the swarm has also been studied in the literature. However, in certain applications, the agents face a hostile environment that can result in the loss of some of the agents during the search, as in search-and-rescue missions behind enemy lines, in drone swarms in modern warfare, and in targeted drug-delivery using nano-technologies. For example, in targeted drug-delivery using smart nano-particles that could communicate with each other to locate the affected site, the carriers face a hostile environment wherein the body's natural defense mechanism (i.e., the immune system) perceives them as potential threats and attempts to kill them in the bloodstream. As a result, the number of agents as well as the information available to the remaining agents dynamically change during the search. This usually has a detrimental impact on the performance of the swarm. In applications such as supply chain optimization [34–36] and logistics [37], these problems become crucial. Graph-theoretic measures have been used to quantify efficiency and robustness of these network topologies [38,39] and there have been attempts to bridge the areas of network science and evolutionary algorithms [40].

In this paper, we build on the earlier contributions of Giridhar et al. [41] and Krishnamurthy et al. [42] to develop a generic framework that unifies the performance and graph-theoretic properties of optimal network topologies under hostile conditions. We consider an environment in which agents are killed at random during the search. We study the performance of different swarm topologies under such conditions and discover generalized properties of desirable network topologies. The rest of the paper is organized as follows. In Section 2, we present the framework of PSO and the role of network topologies in driving its performance. Section 3 describes the agent-based simulation setup including the objective functions, parameters chosen for the PSO algorithm, the set of systematically generated network topologies primarily used in this work, and the probabilistic framework used for simulating hostile environments for the agents. In Section 4, we discuss graph-theoretic properties for efficiency and robustness, and introduce a set of measures used for quantifying performance of various network topologies in the PSO framework. The results corresponding to various levels of hostility on different objective functions are shown in Section 5. A discussion on the major findings of this study is presented in Section 6. Finally, in Section 7 we summarize our contributions along with a few concluding remarks.

2. Problem formulation and objectives

Let us consider a function $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ that we wish to optimize. For the sake of simplicity, we consider the range of $f(\cdot)$ to be $\mathbb{R}$ and the function to exhibit only one global optimum with several local optima. The objective of PSO is to optimize the given function, i.e., to find the value $\mathbf{x}^*$, referred to as the optimal solution, that results in the best value (global optimum) of $f(\cdot)$. We intend to study the impact of network topology on the performance of a swarm intelligence-based algorithm such as PSO in a hostile environment. In the following, we briefly introduce particle swarm optimization and highlight the importance of network configuration on the performance of the algorithm.

Particle Swarm Optimization is a nature-inspired algorithm that begins with a set of *agents*, typically initialized at random locations in the search space. The positions of the agents are updated at each iteration according to a set of heuristics-based rules that account for the best known position of the agent and that of the swarm. The update rule in the standard PSO results in the agents moving in a direction that is the resultant of the direction of the best position of the swarm ($\vec{g}$) and the best position of the i^{th} agent ($\vec{p}_i$), weighted by acceleration coefficients and a stochastic component in each direction. The update rule for the position of the i^{th} agent in a swarm at iteration t can be expressed as follows:

$$\vec{v}_i(t) = \chi[\vec{v}_i(t-1) + \phi_1.\text{rand}(0,1)(\vec{p}_i(t-1) - \vec{x}_i(t-1)) + \phi_2.\text{rand}(0,1)(\vec{n}_i(t-1) - \vec{x}_i(t-1))] \quad (1)$$

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (2)$$

In the above equations, $\vec{x}_i(t)$ and $\vec{v}_i(t)$ represent the position and velocity, respectively , of the i^{th} agent at iteration t; $\vec{p}_i(t-1)$ and $\vec{n}_i(t-1)$ represent the best known positions of the i^{th} agent and its neighborhood at iteration $(t-1)$, respectively; ϕ_1 and ϕ_2 are the acceleration coefficients, and χ represents the constriction coefficient. In a swarm where all agents communicate with each other (in standard PSO), $\vec{n}_i(t-1)$ represents the best position of the entire swarm ($\vec{g}$) and is the same for all agents, while for other configurations, $\vec{n}_i(t-1)$ represents the best position of the neighbors of the i^{th} agent and can be different for different agents. These update rules result in exploration of the search space as well as exploitation of knowledge of the landscape over multiple iterations. Table 1 presents the step-by-step procedure for the standard PSO algorithm.

It can be clearly observed from Eqs. (1) and (2) that the updated positions of the agents depend on the manner in which the best-known position among the neighbors($\vec{n}_i(t-1)$) is obtained at each iteration. This is in turn determined by the topology of the communication network of the swarm. The network configuration thus influences the information received by each agent, determining the updated positions for the agents, and plays a crucial role in convergence to the global optimum. The impact of network topology on PSO performance is well-studied in the literature [43–45] and it has been established that network topologies play a significant role in determining the convergence of the algorithm. However, general topological factors that result in superior performance have not been identified. Furthermore, the performance of the algorithm is also seen to have a strong dependence on the type of function being optimized. These studies are also limited to the case where agents are not lost to the environment as a result of which the trade-off between efficiency of convergence and robustness to the environment is not studied. We conduct several computational experiments that simulate PSO in a hostile environment which are discussed in detail in the next section.

3. Agent-basedsimulation setup

The computational experiments considered in this work have four crucial components, the choice of which determines the performance of the algorithms, and hence the generalizability of results. These factors include:

(i) the objective function
(ii) the hyperparameters of the PSO algorithm
(iii) the communication network configuration of agents in PSO
(iv) the severity and nature of the hostile environment

The following sections present short discussions on the choice of the above factors and the rationale behind them.

3.1. Objective function

We consider five standard benchmark objective functions – Shekel, Ackley, Griewank, Schwefel and Rastrigin [46,47] – to evaluate and compare the performance of PSO for different network topologies. We perform extensive computational experiments with the Shekel function to identify common patterns in PSO performance and then choose a subset of topologies that exhibit a desirable trade-off between efficiency of convergence and robustness to hostile environments, before comparing their performance on the remaining four functions. The first objective

Table 1
Pseudo code for the standard PSO algorithm.

Algorithm 1: Standard PSO
Inputs: `nAgents`: the number of agents in the swarm f: the function to be optimized `maxIters`: the maximum number of iterations

Initialize the agent positions and velocities randomly
while $t <$`numIters`:
for each agent i in `nAgents`:
Compute $\vec{p}_i$, position of best solution agent i has found so far
Compute $\vec{g}$, position of best solution found by all the agents in the swarm so far
Update $\vec{v}_i$, velocity of agent i updated using Eq. (1)
Update $\vec{x}_i$, position of agent i updated using Eq. (2)

Outputs: $\vec{x}^*$, the position of the global optimum $f(\vec{x}^*)$, the function values at the global optimum

Table 2
Description of objective functions.

Function	Expression	n	Range	Local Optima	Location of Global Optimum
Shekel	$f(\mathbf{x}) = \sum_{i=1}^{m}\left(c_i + \sum_{j=1}^{n}(x_j - a_{ji})^2\right)^{-1}$	4	$[0, 10]$	10	$[4, 4, 4, 4]^T$
Ackley	$f(\mathbf{x}) = -a\exp\left(-b\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)\right) + a + \exp(1)$	2	$[-15, 30]$	> 10	$[0, 0]^T$
Griewank	$f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	2	$[-600, 600]$	> 10	$[0, 0]^T$
Schwefel	$f(\mathbf{x}) = 418.9829n - \sum_{i=1}^{n}\left[-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right]$	2	$[-500, 500]$	> 10	$[420.9687, 420.9687]^T$
Rastrigin	$f(\mathbf{x}) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	2	$[-5.12, 5.12]$	> 10	$[0, 0]^T$

function considered in this work is the Shekel function which can be expressed as:

$$f(\mathbf{x}) = \sum_{i=1}^{m}\left(c_i + \sum_{j=1}^{n}(x_j - a_{ji})^2\right)^{-1} \tag{3}$$

where, m is the number of local maxima, n is the dimension of the input space ($\mathbf{x} \in \mathbb{R}^n$), c_i determines the magnitude of the i^{th} local maximum and a_{ji} is the j^{th} coordinate of location of i^{th} maximum. Fig. 1(a) depicts the landscape of a two-dimensional Shekel function with 9 local maxima and one global maximum. The remaining four objective functions are shown in Fig. 1(b)-(e) for the two dimensional case. The functional form, number of dimensions, number of local optima and location of global optimum for the five functions consider in our work are listed in Table 2.

3.2. Hyperparameters for particle swarm optimization

The second component includes the set of hyperparameters used in the PSO algorithm. Throughout all the simulations, we have chosen the following values of the hyperparameters:

1. $\chi = 0.7298438$, a constriction coefficient that prevents the velocities from exploding
2. $\phi_1 = 0$, $\phi_2 = 2.05$, acceleration coefficients for the agents' cognizance and neighborhood cognizance
3. $V_{min} = -10, V_{max} = 10$, lower and upper bounds on the agents' velocities
4. $N = 100$, number of agents in the swarm
5. $T = 1000$, maximum number of iterations to allow convergence of all network configurations

Remark 1: In this study, we set $\phi_1 = 0$ so that the best known position of an agent is not incorporated in updating its position at any iteration. As a result, agents move in a direction that is solely driven by the information made available to them by their neighborhood.

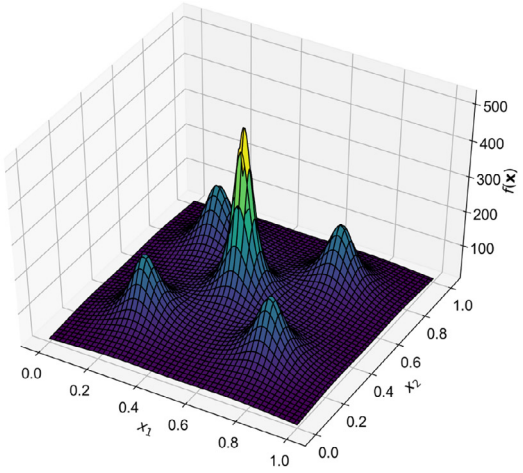
Remark 2: It is important to note here that while the hyperparameters of the algorithm could be tuned to achieve desired efficiency and robustness for each scenario, such an exercise can be practically infeasible and thus of little significance in several applications. Furthermore, owing to the uncertainty in the form of random loss of agents in our study and absence of complete information about the objective function in practical scenarios, a perfect tuning of hyperparameters can be extremely time-consuming, difficult, and computationally expensive. Therefore, we fix the values of the hyperparameters for all the experiments and focus the study towards discovering generalized features of the algorithm that are of interest in practical applications. With the objective function and tuning of the algorithm fixed, we next describe the different network configurations and the simulation of the hostile environment used in our study.

3.3. Network topology of communicating swarm

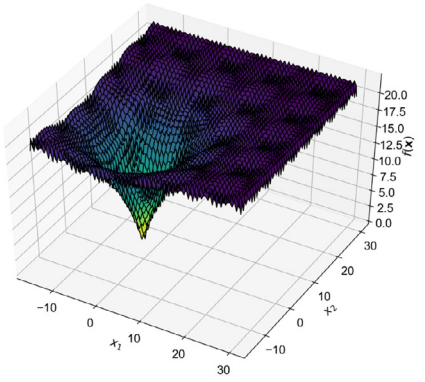
Network topologies can broadly be classified as deterministic and random topologies. Since the objective of this study is to identify network configurations that result in a desired trade-off between efficiency and robustness of the algorithm, we focus our attention on deterministic graphs and present only limited results on random graphs. We consider three classes of graphs – complete (all agents connected to each other), star (hub-and-spoke configuration) and ring (all agents connected to two neighbors, forming a closed ring) – that represent different degrees and manners of connectivity of the graph. We create a spectrum of intermediate graphs by performing deterministic operations on the above graph classes as follows:

I. Complete–to–Star:
   (a) The initial graph is a complete graph where all the agents constitute the fully connected core with each node connected to every other node
   (b) The core size of the graph is shrunk in steps of 1 by removing an agent from the core and attaching it to one of the core nodes
   (c) In reducing the core size, it is ensured that the resultant core at each step is full connected, as shown in Fig. 2(a).

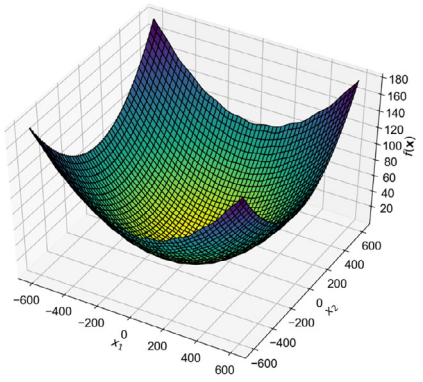
II. Star–to–Ring:



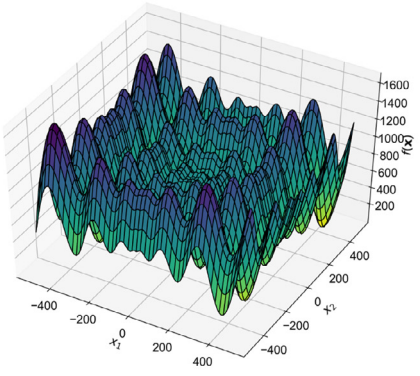
(a) Shekel Function



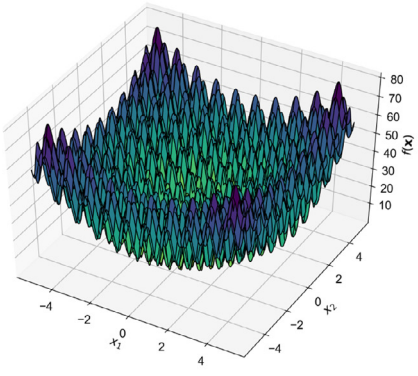
(b) Ackley Function



(c) Griewank Function



(d) Schwefel Function



(e) Rastrigin Function

Fig. 1. Benchmark objective functions for $n = 2$.

(a) The initial graph is a star graph which consists of only one core node and $N - 1$ non-core nodes.
(b) The core is expanded in steps of 1 to connect one non-core node to the core at each step.
(c) In expanding the core, it is ensured that the core is not fully connected but connected in a ring topology as shown in Fig. 2(a).

III. Ring–to–Complete:
(a) The initial graph is a ring-graph where each node is connected to its two immediate neighbors in a circular topology

Table 3
Pseudo code for randomized death of agents.

Algorithm 2: `Randomized Death` of agents
Inputs: p : the deactivation probability for each agent `active agents`: the list of agents that are active

for each `agent` in `active agents`:
  Generate a random number r in the range $[0, 1]$
  if $r < p$:
  deactivate `agent`
  Remove `agent` from the list `active agents`

Outputs: `active agents`: list of active agents after randomized death

(b) In each step, an additional ring structure is introduced to the graph by symmetrically connecting agents to the non-immediate neighbors.
(c) This results in a multi-ring graph with increasing connectivity as one moves from ring to complete graph as shown in Fig. 2(a).

The above rules are used to generate 80 configurations between each pair of basic graphs, resulting in a total of 240 configurations. The colors shown in Fig. 2(a) are used to demarcate different segments of the spectrum and are used for clarity in presentation of the results. In addition, we also use the von Neumann grid, scale-free graph, random graph and small-world graph as additional cases of network configurations as shown in Fig. 2(b)–(e). These networks are generated using the NetworkX 2.4 package in Python. Specifically, random graphs are generated in a manner that the probability of an edge between any two nodes is 0.1, while small-world graphs are generated with a degree of 10 for each node and a rewiring probability of 0.1 for each edge.

3.4. Loss of agents in the hostile environment

We consider a hostile environment where each agent has a probability of getting killed/deactivated at a given iteration. Let us consider a scenario where all agents have the same probability p of being lost to the environment at any iteration of the algorithm. Then, the expected number N_a of the alive/active agents at the end of t iterations can be obtained as:

$$N_a = N(1-p)^t \tag{4}$$

The expected fraction of active and deactivated agents, F_a and F_d, respectively are given by,

$$F_a = (1-p)^t \tag{5}$$

$$F_d = 1-(1-p)^t \tag{6}$$

We study the performance of PSO in an environment that causes loss (death fractions) of 15% and 30% of the total number of agents at the end of 500 iterations. The corresponding probabilities of deactivation for each agent for these cases are $p = 0.00033$ and $p = 0.0007$, respectively. The pseudo-code for the randomized death of agents is presented in Table 3.

The combined pseudo-code for the resultant algorithm that combines network topology with loss of agents for the PSO algorithm is presented in Table 4.

In the next section, we present different graph-theoretic metrics considered in this study to quantify the robustness and efficiency of different network topologies followed by various measures that capture performance in the PSO framework.

4. Graph theoretic properties and PSO performance metrics

The convergence of PSO with different network topologies in a hostile environment depends on – (1) the ability of the network to *efficiently* transmit information between the nodes (agents), and (2) the ability of the network to sustain loss of agents with minimal deterioration in performance. We first present graph-theoretic metrics that quantify the efficiency and robustness of a network topology in itself (independent of PSO) followed by a set of four metrics that we use to quantify the performance of any topology in the PSO framework.

We wish to highlight here that the networks that maximize the graph-theoretic measures of efficiency and robustness do not necessarily result in the most efficient and robust performance on the PSO algorithm as we discuss in detail later in Section 5.

4.1. Graph-theoretic properties

The efficiency of a graph is its ability to quickly communicate information between different nodes in the graph. The average geodesic distance (also called average path length) is a very commonly used metric to quantify efficiency of a graph. The average geodesic distance L of a graph G with N nodes can be expressed as:

$$L = \frac{1}{N(N-1)} \sum d_{ij} \tag{7}$$

where d_{ij} is the length of the shortest path between i^{th} and j^{th} nodes of G. The above quantity is equal to 1 for a complete graph, signifying highly efficient transfer of information in the graph. On the other hand, the star graph exhibits an average geodesic distance of approximately 2 while the ring graph has an average geodesic distance $\sim N/4$.

The robustness of a graph can be quantified using the natural connectivity which has been shown to be sufficient, superior, and is known to be a physically meaningful measure for quantifying the robustness of complex graphs [48]. Physically, it is proportional to the number of closed loops for each node of the graph and is derived from the eigenvalues of the adjacency matrix of the graph as:

$$\bar{\lambda} = \ln\left(\frac{1}{N}\sum_{i=1}^{N} e^{\lambda_i}\right) \tag{8}$$

where λ_i is the i^{th} eigenvalue of the adjacency matrix of the graph. The adjacency matrices of a complete, star and ring graphs (A, B, and C, respectively) with 5 nodes can be represented as:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The set of eigenvalues for the adjacency matrices (graph spectrum) for the three graphs above is given by,

$$A_\lambda = \{4, -1, -1, -1, -1\}$$
$$B_\lambda = \{2, 0, 0, 0, -2\}$$
$$C_\lambda = \{2, 0.62, 0.62, -1.62, -1.62\}$$

Therefore, using Eq. (8), the corresponding natural connectivities obtained for the given graphs are,

Table 4
Pseudo code for PSO algorithm with network topology and dying agents.

Algorithm 3: PSO with Network Topology and Dying Agents
Inputs: `nAgents`: the number of agents in the swarm f: the function to be optimized `maxIters`: the maximum number of iterations p : the death probability for each agent `adjMatrix`: the `nAgents` $\times$ `nAgents` matrix describing the network connectivity for each agent Initialize the agent positions and velocities randomly

while $i <$ `numIters` and `numActiveAgents` $\neq 0$:
  for each agent i in `active agents`:
  Compute $\vec{p}_i$, position of best solution agent i has found so far
  Compute $\vec{n}_i$, position of best solution found by agent i's neighborhood so far defined by the `adjMatrix`
  Update $\vec{v}_i$, velocity of agent i updated using Eq. (1)
  Update $\vec{x}_i$, position of agent i updated using Eq. (2)
  Invoke the `Randomized Death` algorithm and update `active agents`, `adjMatrix`

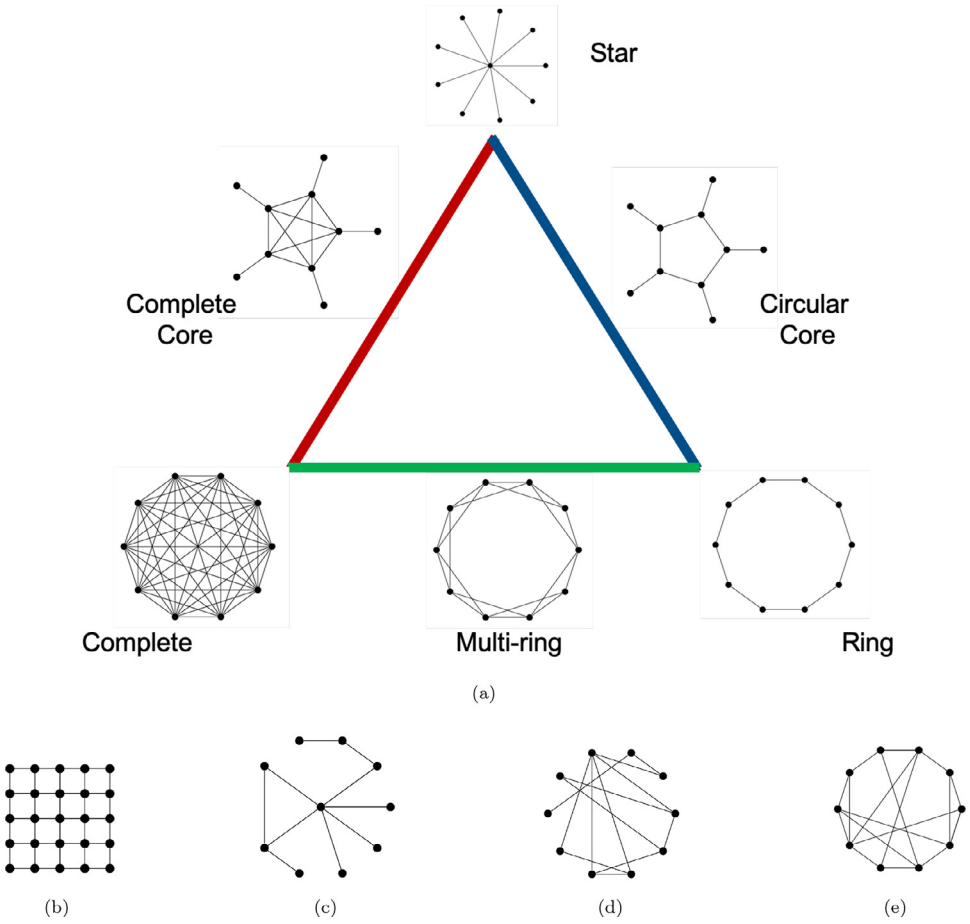


Fig. 2. Network topologies used for swarm connectivity: (a) 240 network topologies are generated by traversing along the spectrum from complete-to-star (red), star-to-ring (blue), and ring-to-complete (green) (b) a von Neumann grid: each node is connected to 4 neighboring nodes, (c) Scale-free graph: the probability of finding a node with degree k is proportional to $k^{-\gamma}$, (d) Random graph: edges between nodes are assigned at random with a predetermined probability p_e, (e) Watts-Strogatz (small-world) graph: edges of a multi-ring graph are rewired with a rewiring probability p_w.

$\bar{\lambda}_{complete} = 2.42 \quad \bar{\lambda}_{star} = 0.74 \quad \bar{\lambda}_{ring} = 0.83$

It can be clearly seen that the natural connectivity values for the three graphs are in agreement with the expected behavior. For example, the complete graph is the most robust network owing to the presence of multiple loops for a given node, while star and ring graphs – with zero and one loops respectively – are the least robust since the network can be fragmented with removal of only a few nodes. Fig. 3 shows the computed measures of efficiency and robustness for the triangular spectrum of topologies described in Fig. 2(a). It can be seen that as we traverse the spectrum, the following patterns in the efficiency and robustness are observed:

- Complete–to–Star: The average path length increases and natural connectivity decreases. This is equivalent to decreasing efficiency and robustness, respectively. This is a result of removal of edges from the graph in this section of the spectrum.
- Star–to–Ring: The average path length increases while natural connectivity remains same, suggesting decreasing efficiency and near-constant robustness. This is a result of increase in the core size for approximately the same number of edges along this section of the spectrum.
- Ring–to–Complete: The average path length decreases and natural connectivity increases, resulting in an increase in network efficiency and robustness. This is a result of increased number of edges due to addition of multi-rings.

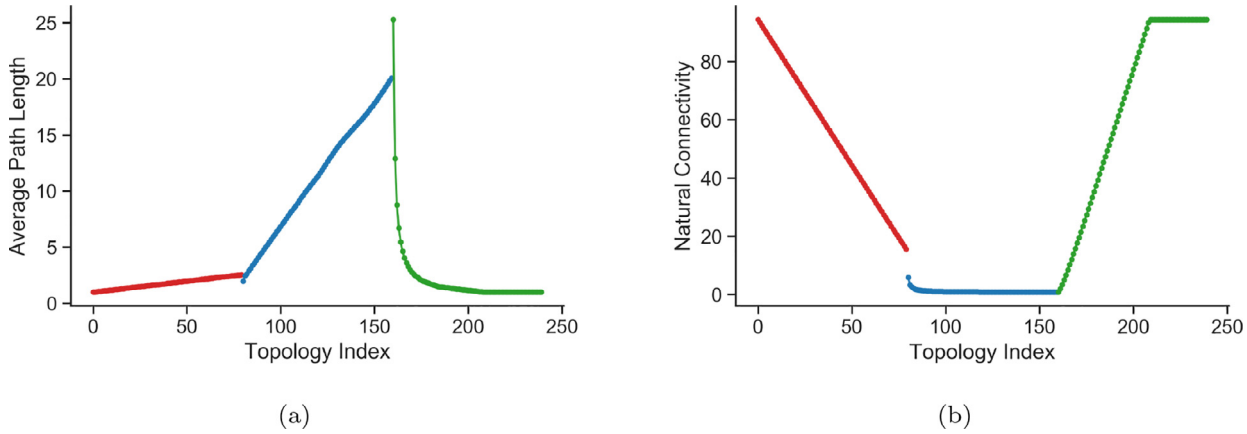


Fig. 3. Network topology metrics (a) Average Path Length and (b) Natural Connectivity as a function of Topology Index corresponding to the spectrum of networks shown in Fig. 2. Average path length is a measure of efficiency of the network while natural connectivity is a measure of robustness. The color codes correspond to the traversal from complete-to-star (red), star-to-ring (blue) and ring-to-complete (green) topologies.

4.2. PSO performance metrics

In order to compare the performance of PSO for different communication network configurations and different severity of the hostile environment, we note the following metrics:

1. Global Success Ratio (GSR): The fraction of times the *entire swarm* reaches the global optimum
2. Global Success Time to Convergence (GS Time): Average number of iterations required for the *entire swarm* to reach the global optimum. This metric is calculated only for the cases when *global convergence* is achieved.
3. Number of Winners: Average number of agents that reach the global optimum
4. Trade-off metric: In addition to the performance measures defined above, we define the following performance metric that captures the trade-off between the number of winners and the time for convergence:

$$\text{Trade-off metric} = \alpha \frac{\text{Winners}}{\text{Winners}_{\max}} - (1-\alpha)\frac{\text{GS Time}}{\text{GS Time}_{\max}} \tag{9}$$

where α determines the relative importance of the two measures towards the metric. In our simulations, we choose $\alpha = 0.7$ to assign a relatively higher importance to success ratio, emphasizing on convergence of all agents to the global optimum. This implies that we desire all the agents to reach the global optimum while allowing for a larger time to reach convergence.

In the next section, we present the results of the study based on the above performance metrics for the benchmark objective functions, for different death fractions with respect to the graph-theoretic metrics.

5. Results

In order to balance out the effect of initial state of agents on the convergence of the algorithm and improve generalizability of results, 50 different initializations of agents for each network topology and deactivation were considered. In each scenario, the performance metrics listed in the previous section were noted and the average metric for each configuration was computed. We first present the performance of PSO for different communication network topologies and death fractions, followed by an analysis of the performance in relation to graph-theoretic metrics of efficiency and robustness. As discussed earlier, we first perform exhaustive experiments with the 4D-Shekel function to identify common trends and then present results on additional benchmark functions with a subset of important topologies.

5.1. Performance of PSO for different network topologies

The performance of PSO in terms of the four metrics listed above as a function of the network topology is presented in Fig. 4. For the sake of clarity, all metrics corresponding to networks in complete-to-star, star-to-ring and ring-to-complete regimes of the spectrum are shown in red, blue and green colors, respectively.

It can be observed from Fig. 4 that in the case of no deactivation of agents, the global success ratio (GSR) increases as we move from complete graph to star graph, becomes nearly flat at the maximum value of 1 while going from star graph to ring graph, and then decreases again while moving from the ring graph to complete graph. However, with deactivation of agents, GSR decreases in all the three regimes, with a significant increase in GSR after the initial few topologies in the ring-to-complete regime.

The global success time to convergence (GS Time) is lowest for the complete graph with only a marginal increase as we move towards the star topology. However, it rapidly increases while going from the star graph to the ring graph, before falling off again in the ring-to-complete regime. These trends are observed irrespective of the degree of hostility of the environment.

The number of winners for all the three regimes exhibits a similar trend as GSR. However, it is interesting to note that in the case of deactivation of agents, the number of winners in the complete-to-star regime follows an inverse trend with GSR suggesting a higher number of winners for the star-like topologies than completely connected topologies.

Finally, the trade-off metric that captures the weighted combination of number of winners and convergence time is maximized for an intermediate region in the star-to-ring and ring-to-complete regimes across all death fractions. It is also fascinating to note that the networks that maximize this metric under non-hostile conditions also result in the best performance under hostile conditions. This suggests an invariance in the performance of a network topology to hostility of the environment that can be exploited in designing robust network topologies. This is explored later in the article.

This section highlighted the dependence of PSO performance on various network architectures. While the complete graph is known to be robust and efficient graph in a purely graph-theoretic sense, such a topology exhibits consistently poor performance (GSR of $\sim 50\%$) in terms of optimizing the objective function. In the following sections, we adopt a systematic approach to explore underlying patterns relating the properties of a graph with its performance in the PSO framework.

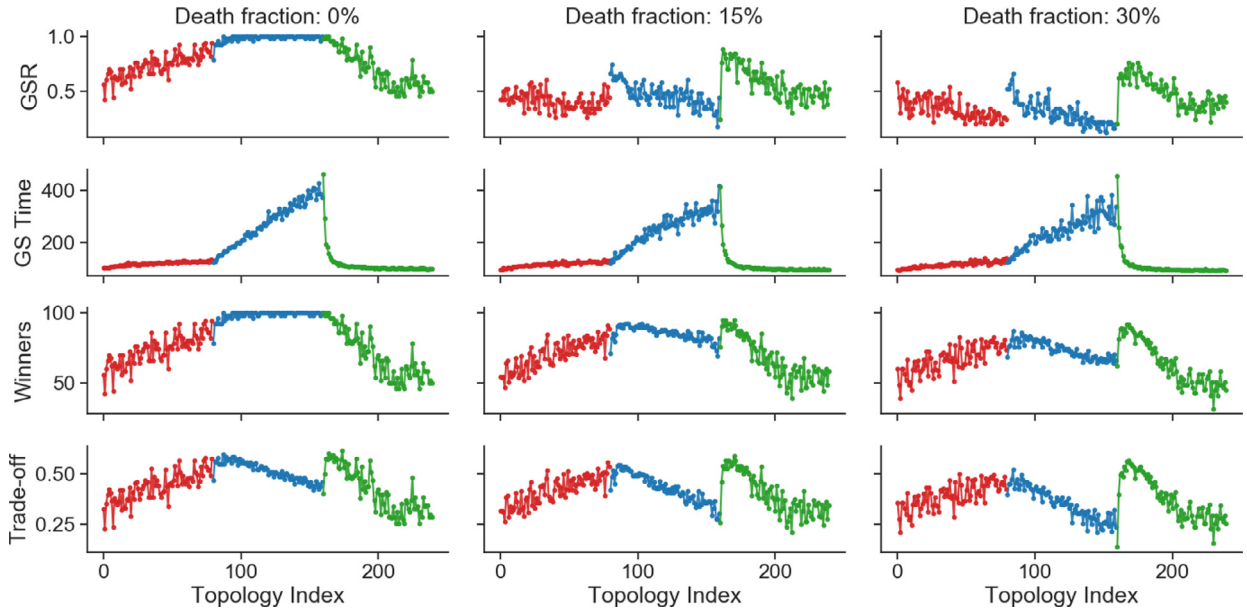


Fig. 4. Global Success Rate (GSR), Global Success Time to Convergence (GS Time), Number of Winners, and Trade-off metric as a function of Topology Index. For 0% death fraction, the GSR increases in the complete to star graph regime, remains almost constant in the star to ring graph regime, and decreases in the ring to complete graph regime. This trend, however, changes significantly as the death fraction is increased to 15% and then further to 30%. Similar behavior towards death fractions could be seen for the other metrics such as GS Time, Number of Winners, and the Trade-off metric.

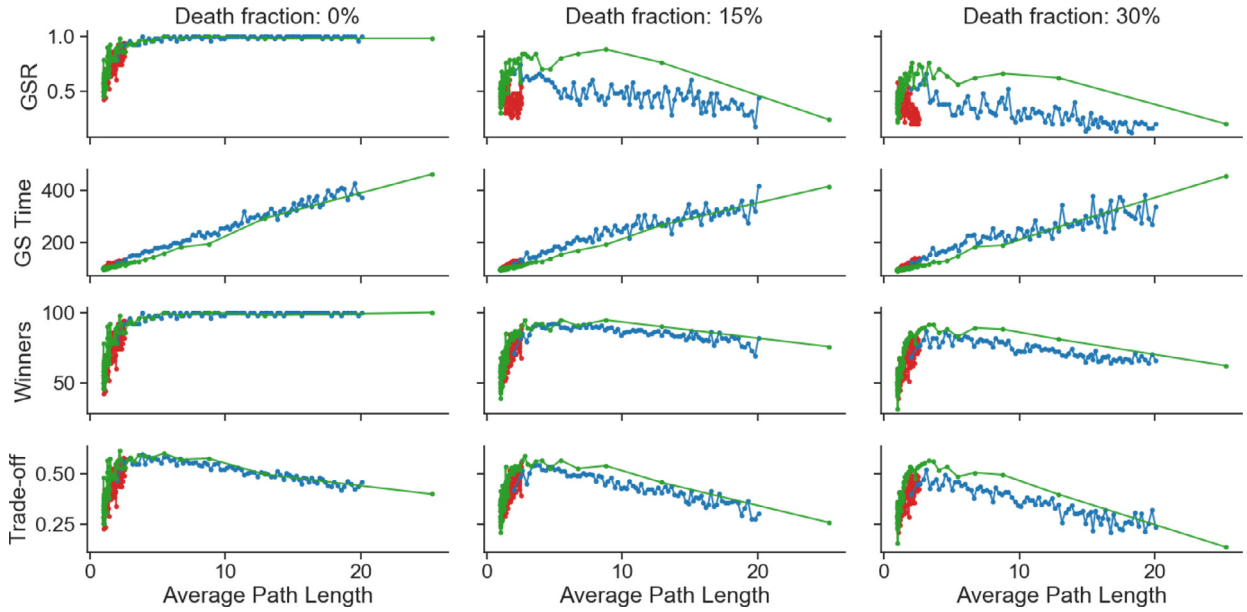


Fig. 5. Global Success Rate (GSR), Global Success Time to Convergence (GS Time), Number of Winners, and Trade-off metric as a function of Average Path Length. Across all the death fractions, the GSR, Number of Winners, and the Trade-off metric reach a peak around an average path length of ~ 3. This indicates towards the existence of a class of networks that are optimally connected and maximize the PSO performance under hostile environments.

5.2. *Performance of PSO andgraph efficiency*

We now study the variation of network performance as a function of its graph-theoretic efficiency. The four performance metrics of PSO defined in Section 4.2 as a function of the average geodesic distance (L) of the network are shown in Fig. 5. In the case of no deactivation of agents, the GSR initially increases with increase in L after which it saturates close to the maximum value of 1 for larger values of L. When agents are deactivated, the absolute values of GSR across all values of L reduce with the magnitude of deterioration being higher for large values of L. Irrespective of the rate of deactivation, GSR reaches a maximum for $L \sim 3$, indicating toward the performance invariance of such graphs performance towards hostile environments.

The GS Time increases linearly with increase in average geodesic distance of the graph. This is expected since larger values of L imply lower information transfer through the network resulting in longer times to convergence. The number of winners exhibits a trend that is similar to that observed for GSR but the sensitivity to loss of agents is relatively lower. The trade-off measure capturing the number of winners and the GS Time is also maximized for an average path length ~ 3 irrespective of the network topology, or the death fraction. This is suggestive of the existence of a class of networks that are inherently

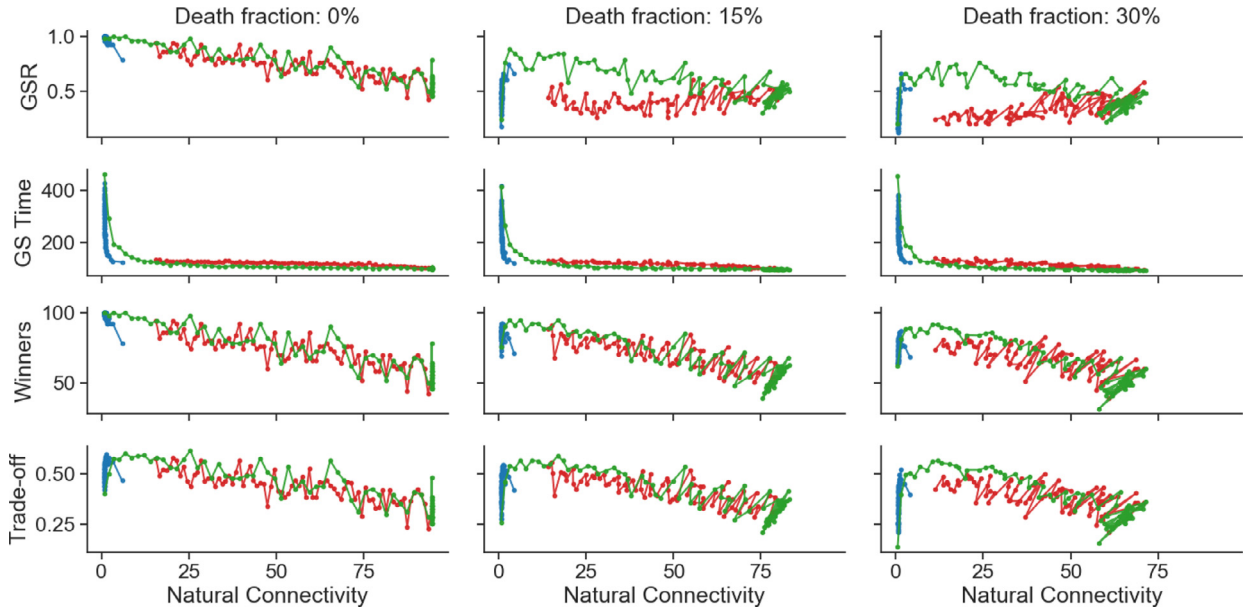


Fig. 6. Global Success Rate (GSR), Global Success Time to Convergence (GS Time), Number of Winners, and Trade-off metric as a function of Natural Connectivity. The performance of the topologies with respect to the graph-theoretic measure of robustness (natural connectivity) indicates that there exists several network topologies that have the same value of natural connectivity but significantly different performance in the PSO framework. It is observed that networks with decentralized architectures perform relatively better than those with centralized structures.

suited for maximizing performance under scenarios involving loss of agents.

It can be inferred from the above observations that there is a trade-off between the overall performance of a network in PSO and its average path length. Specifically, while highly efficient graphs take less time to converge, the GSR is much lower because of premature convergence to local optima. On the other hand, while graphs with low efficiency promise near-certain convergence, they take longer to reach the global optimum. The trade-off metric captures this effect well, and reveals that the performance peaks at an average path length of ~ 3.

5.3. Performance of PSO and graph robustness

After studying the performance of PSO with respect to graph efficiency quantified by the average geodesic distance, we study the same with respect to the graph-theoretic robustness of networks. The performance of PSO as a function of the natural connectivity (robustness) of the network topology is shown in Fig. 6.

It is observed that in the absence of a hostile environment, the GSR decreases with an increase in robustness of the graph. This is potentially due to the presence of larger number of closed-loops for highly robust graphs that result in a sub-optimal exploration of the search space and consequently results either in premature convergence, or no convergence at all. An interesting observation is that for the same value of natural connectivity, there exist multiple architectures that result in significantly different performances. For instance, networks with decentralized architectures (such as multi-ring) with nearly the same connectivity for all nodes perform better than networks with centralized structures (such as a star-graph) where a few central-nodes are highly connected to a large fraction of other nodes. This is because in centralized networks, removal of the central-nodes would significantly limit the information transfer across the network.

Therefore, the foregoing sections have established that in order to maximize performance of a network topology in the PSO framework, the network should be sufficiently connected with an average geodesic distance of ~ 3 to ensure optimal information transfer and should have decentralized architectures in order to ensure robustness towards hostile conditions where agents are deactivated at random.

5.4. Performance of PSO with standard topologies

In addition to the 240 topologies from the triangular spectrum described in Fig. 2(a), we study the performance of PSO with a few standard network topologies that do not fall under any regime in the above spectrum. Specifically, we consider the von Neumann grid, scale-free graphs, random graphs, and small-world graphs for our study. The results are presented in Table 5.

It is observed that the von Neumann grid has the highest number of winners and the largest GSR but the convergence time of this graph is the largest among all the four topologies considered. This behaviour is suggestive of lower information transfer (and hence, lower efficiency), which is also reflected in its largest average geodesic distance across all other networks. It also exhibits a highly sensitive GSR with respect to loss of agents, which is explained by the lowest value of natural connectivity ($\bar{\lambda}$) for this network.

The scale-free network exhibits a low average geodesic distance and hence higher efficiency, which results in lower convergence times, but a slightly lower GSR. However, the GSR of this network is extremely sensitive to loss of agents even with a relatively high natural connectivity because of the centralized architecture similar to the networks in the star-to-ring regime. This results in a performance that is highly sensitive to the loss of agents in the network.

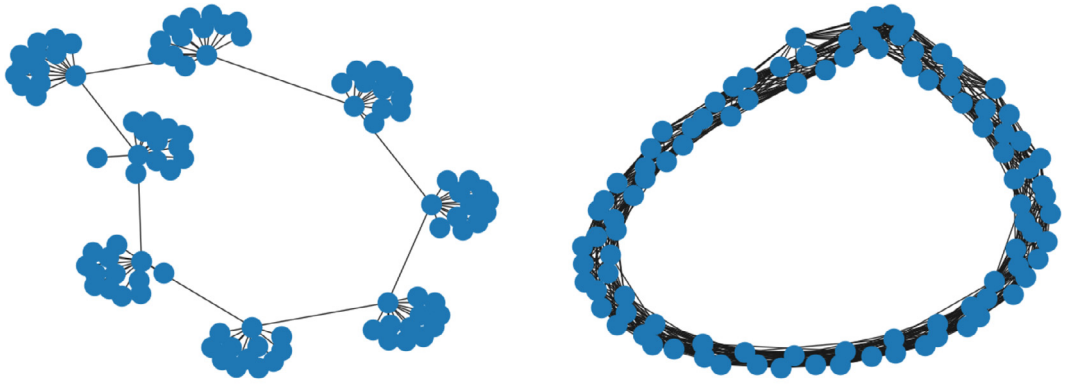
Random graphs exhibit the highest efficiency and hence the lowest convergence times. However, they also result in a lower GSR, potentially because of a relatively higher information overload in the network. Furthermore, due to lack of central-nodes in random graphs, the performance is seen to be robust to loss of agents.

Finally, small-world graphs are seen to result in a relatively higher GSR and lower convergence times as a result of higher efficiency. These graphs can also be observed to be relatively robust to loss of agents. The performance of small-world graphs is marginally better than random graphs in terms of the GSR and the number of winners in the PSO framework. The convergence times are, however, comparable with the random-graphs converging marginally faster than the small-world graphs.

In the final section, we present the results of PSO on standard benchmark optimization functions for selected networks, followed by a de-

Table 5
Comparison of mean performance metrics for standard topologies on the Shekel function. The small-world topology with an average geodesic distance of $L \sim 2.72$, and natural connectivity of $\lambda \sim 6.31$ outperforms every other standard topology by retaining its relatively better PSO performance across all death fractions.

Network	Death Fraction (%)	Number of Winners	GSR	GS Time	L	$\bar{\lambda}$
von Neumann	0	100	1.00	189.1	6.67	1.51
	15	95	0.70	168.5		
	30	87	0.68	174.1		
Scale-free	0	96	0.96	138.7	2.75	20.01
	15	94	0.56	135.8		
	30	75	0.34	139.4		
Random	0	94	0.94	106.5	1.70	25.69
	15	91	0.78	105.0		
	30	85	0.72	101.9		
Small-world	0	98	0.98	124.7	2.72	6.31
	15	94	0.78	117.5		
	30	88	0.78	121.5		



(a) Star-to-ring regime with 8 hubs (b) Ring-to-complete regime with 9 multi-rings

Fig. 7. The top–2 topologies from the triangular spectrum in Fig. 2 in terms of robustness and efficiency on the Shekel function. Both the topologies are characterized by an average geodesic distance $L \sim 3$ along with a decentralized architecture that ensures robustness towards node deactivation.

tailed discussion of the different inferences on desired network characteristics for maximizing PSO performance.

5.5. Effect of objective function

In order to study the impact of objective function on network performance in PSO, we identified key network configurations that exhibit a desirable trade-off between efficiency, robustness, and performance to analyze their performance on four standard objective functions.

For this study, we choose the complete graph, star graph, and the ring graph topologies (corners of the triangular spectrum in Fig. 2(a)) along with the standard topologies discussed in the previous section. In addition, we consider two graphs that result in the best performance among the 240 topologies in the triangular spectrum – one in the star-to-ring regime with 8 central-nodes and the other in the ring-to-complete regime with 9 multi-rings as shown in Fig. 7.

We test the performance of these graphs on the Ackley, Griewank, Schwefel, and Rastrigin functions with function parameters and range as described in Table 2. The results for the respective functions are presented in Tables 6, 7, 8, and 9.

It is observed that for each objective function, there exists a network topology that results in best performance in terms of high GSR and number of winners with low convergence times. For instance, across all death fractions, the von Neumann topology results in best performance for Ackley function, small-world topology for Griewank function, small-world for Schwefel function, and complete graph for Rastrigin function. These topologies maximize the number of winners and GSR while simultaneously minimizing the time to convergence (GS Time).

However, small-world graphs are observed to consistently outperform almost all the other topologies irrespective of the choice of objective function or the death fraction. Even for the Ackley and the Rastrigin functions where the von Neumann graph and complete graph perform the best, small-world graphs result in a performance very close to the best performance achieved by both of these networks. We attribute such a superior performance to two factors – first, the average geodesic distance of 2.72 for small-world graphs as reported in Table 5 is close to our observation of optimal information transfer occurring at $L \sim 3$; second, the small-world networks have a decentralized architecture with almost equal connectivity for all the nodes in the network resulting in a robust performance in the presence of higher deactivation of agents as discussed in the foregoing sections.

Similar to small-world networks, the 9 multi-ring architecture of Fig. 7(b) also exhibits consistently better performance across all the objective functions and is robust to various levels of death fraction. The 9 multi-ring network architecture is characterized by an average geodesic distance of 3.27 and has a decentralized architecture – validating the underlying network structures identified in the previous sections that are attributed to better and robust PSO performance. This is further reinforced by analyzing the performance of the 8-hub network architecture in Fig. 7(a) with a average geodesic distance of 3.86 and with 8 central-nodes. Although this architecture results in faster convergence and better performance under non-hostile conditions, the performance degrades significantly when agents are deactivated at random due to the fragility of such an architecture towards random deactivation of agents.

Table 6
Comparison of mean performance metrics on Ackley function.

Network	Death Fraction (%)	Number of Winners	GSR	GS Time
Complete	0	100	1.00	146.2
	15	96	0.72	137.4
	30	91	0.72	136.4
Star	0	100	1.00	156.0
	15	93	0.64	145.9
	30	84	0.68	144.4
Ring	0	100	1.00	156.8
	15	95	0.76	149.9
	30	90	0.80	147.5
8-hub Graph	0	100	1.00	153.6
	15	92	0.60	151.5
	30	87	0.54	148.7
9-ring Graph	0	100	1.00	148.1
	15	95	0.78	141.3
	30	91	0.72	134.4
von Neumann	0	100	1.00	153.3
	15	95	0.80	144.6
	30	90	0.78	138.3
Scale-free	0	100	1.00	154.5
	15	94	0.60	148.6
	30	87	0.50	148.3
Random	0	100	1.00	147.0
	15	96	0.70	141.2
	30	91	0.64	137.8
Small-world	0	100	1.00	150.4
	15	96	0.84	142.4
	30	91	0.74	138.2

Table 7
Comparison of mean performance metrics on Griewank function.

Network	Death Fraction (%)	Number of Winners	GSR	GS Time
Complete	0	58	0.58	154.6
	15	53	0.52	147.7
	30	36	0.32	144.2
Star	0	66	0.66	183.3
	15	68	0.62	172.5
	30	52	0.42	174.5
Ring	0	75	0.38	829.7
	15	34	0	–
	30	17	0	–
8-hub Graph	0	82	0.82	257.3
	15	70	0.42	247.7
	30	60	0.32	260.1
9-ring Graph	0	70	0.70	223.2
	15	66	0.54	197.2
	30	59	0.52	227.0
von Neumann	0	90	0.90	354.7
	15	82	0.66	333.2
	30	72	0.58	347.3
Scale-free	0	80	0.80	218.5
	15	70	0.44	203.5
	30	64	0.28	196.6
Random	0	58	0.58	161.9
	15	43	0.38	160.8
	30	43	0.32	155.3
Small-world	0	86	0.86	201.6
	15	77	0.56	193.5
	30	73	0.52	187.8

Table 8
Comparison of mean performance metrics on Schwefel function.

Network	Death Fraction (%)	Number of Winners	GSR	GS Time
Complete	0	56	0.56	102.4
	15	64	0.54	100.1
	30	58	0.46	93.8
Star	0	86	0.86	117.9
	15	75	0.66	111.0
	30	83	0.66	108.5
Ring	0	100	1.00	289.5
	15	80	0.42	234.4
	30	64	0.20	422.0
8-hub Graph	0	92	0.92	129.8
	15	88	0.64	123.7
	30	81	0.46	120.6
9-ring Graph	0	86	0.86	115.7
	15	83	0.74	113.6
	30	78	0.6	111.4
von Neumann	0	98	0.98	148.9
	15	94	0.80	137.3
	30	86	0.74	134.2
Scale-free	0	92	0.92	126.7
	15	90	0.58	117.4
	30	81	0.50	116.0
Random	0	76	0.76	107.8
	15	76	0.66	103.4
	30	74	0.58	100.5
Small-world	0	84	0.84	118.0
	15	92	0.72	113.4
	30	87	0.74	119.9

Table 9
Comparison of mean performance metrics on Rastrigin function.

Network	Death Fraction (%)	Number of Winners	GSR	GS Time
Complete	0	100	1.00	110.8
	15	97	0.86	107.5
	30	93	0.80	101.4
Star	0	100	1.00	128.5
	15	94	0.76	122.1
	30	90	0.56	119.9
Ring	0	100	1.00	249.1
	15	88	0.44	240.8
	30	78	0.30	220.3
8-hub Graph	0	100	1.00	144.3
	15	94	0.70	135.5
	30	89	0.42	132.0
9-ring Graph	0	100	1.00	114.6
	15	96	0.84	114.7
	30	93	0.78	112.9
von Neumann	0	100	1.00	149.2
	15	96	0.80	140.6
	30	92	0.66	140.6
Scale-free	0	100	1.00	139.8
	15	94	0.66	129.3
	30	88	0.44	124.7
Random	0	100	1.00	115.1
	15	97	0.78	110.9
	30	93	0.72	105.6
Small-world	0	100	1.00	123.5
	15	96	0.86	120.5
	30	92	0.72	116.1

6. Discussion

In the previous section, we presented the results of PSO-based optimization of different objective functions using various network topologies. A key insight derived from these results is that *graph-theoretic metrics of efficiency and robustness do not extend directly towards quantifying the efficiency and robustness of a network topology in the PSO framework*. For instance, agents communicating in highly-connected and hence efficient topologies exhibit poor global convergence rates. This can be attributed to the fact that with every node connected to every other node, each agent in the swarm is presented with an excessive amount of information which results in the swarm converging prematurely to one of the local optima. In general, such disagreements can be attributed to the differences in the definitions of graph-theoretic properties (efficiency and robustness) and the performance of PSO. The graph-theoretic efficiency and robustness are typically defined with respect to the speed of information transfer and the size of the largest connected component in a graph. On the other hand, the efficiency and robustness of a topology in the PSO setting also depends on additional crucial factors such as rate of convergence of the entire swarm to the global optimum, performance tolerance to various levels of hostility, and the characteristics of the objective function such as the number of local and/ global optima, and the function landscape around the global optimum.

In terms of performance, it is required that the networks exhibit sufficiently high levels of connectivity between the nodes to facilitate faster information transfer among agents. However, as seen above, very high levels of connectivity also results in poor performance. It is evident from Fig. 5 that network topologies with an average path length $L \sim 3$ maximize the performance trade-off between the number of winners and the convergence time defined in Section 4.2. This trade-off measure is very small for lower values of average path lengthdue to agents getting stuck in local optima. On the other hand, the trade-off measure decreases gradually as the average path length increases beyond 3 due to reduced information transfer that results in higher convergence times. Hence, *there appears to be a negative utility of increasing the connectivity beyond a certain threshold and networks that exhibit a high-enough connectivity are desirable from a PSO performance standpoint.*

In terms of robustness, it is required that the networks retain their information transfer rates to exhibit robust performance under hostile conditions. Networks in the star-to-ring regime are very likely to fragment under hostile conditions due to disruption in information transfer. The ring graph is the least robust in this regime since removal of even a few nodes fragments the entire topology. On the other hand, centralized structures are relatively more robust with the robustness decreasing with increasing number of central-nodes. This is because the loss of even a single central-node fragments the graph and higher the number of central-nodes in the graph, higher is the probability of graph-fragmentation. Mathematically, the probability of a centralized structure with k central-nodes getting fragmented at the i^{th} iteration can be expressed as $1-(1-p)^{ki}$ with p being the probability of loss of an agent. However, the most robust graphs are characterized by high connectivity in addition to having a decentralized architecture as observed in the ring-to-complete and complete-to-star regimes. For example, complete graphs exhibit the maximum robustness among all the graphs across the spectrum. Therefore, *networks with* decentralized *architectures and high connectivity are desirable from a robustness standpoint.*

In addition to the topologies across the triangular spectrum in Fig. 2(a), our study on several standard topologies involving multiple objective functions revealed that small-world graphs achieve a high global success ratio while retaining the performance with loss of agents. Specifically, while the best performing graph for different objective functions varies, the success ratio of small-world graphs is consistently among the best two topologies identified. Moreover, the convergence time for small-world graphs is consistently close to the lowest convergence time achieved for all objective functions. Small-world network architectures, therefore, appear to be optimizing information transfer and retain their performance under hostile conditions. In order to investigate this effect, we discuss a commonly used metric of *small-world-ness* of networks [49]:

$$\omega = \frac{L_{random}}{L} - \frac{C}{C_{lattice}} \tag{10}$$

Here L is the average geodesic length (Eq. (7)) and C is the average clustering coefficient of the network given as:

$$C = \frac{1}{N}\sum_{i=1}^{N}\frac{2e_i}{k_i(k_i-1)}$$

with e_i being the number of edges between the neighbors of a given node i and $k_i(k_i-1)/2$ is the total number of edges possible between the neighbors. The quantities L_{random} and $C_{lattice}$ represent the average path length of an equivalent random graph and clustering coefficient of an equivalent lattice graph, respectively. The *small-world-ness* is close to zero for small-world networks. The above quantity suggests that for a network to exhibit small-world behavior, it must have an average path length close to that of an equivalent random graph and clustering coefficient close to that of an equivalent lattice graph. Therefore, it can be said that small-world graphs *simultaneously* maximize the efficiency (low average path length) and robustness (high clustering coefficient) of a network. *Therefore, small-world graphs exhibit the desired properties of sufficient connectivity and decentralized architecture, and hence simultaneously maximize efficiency and robustness in the PSO framework.*

7. Conclusions

In this work, we report a study on the performance of particle swarm optimization under hostile environments as a function of the network topologies characterizing communication between agents. Such problems bear significance in applications such as targeted drug-delivery and high-value-target localization. Based on our study, we first conclude that maximizing the graph-theoretic measures of efficiency and robustness of a network do not necessarily result in efficient and robust performance in the PSO framework. This is attributed to factors such as local optima and nature of the function landscape around global optimum that are not incorporated in graph theoretic metrics. Second, we observe that in order to maximize the performance of PSO, a network topology has to be sufficiently connected to ensure optimal information transfer – a higher information transfer results in overexploitation of the information limiting exploration, whereas a lower information transfer results in overexploration of the search space that results in higher convergence times. An average path length of $L \sim 3$ was observed to give the best performance in terms of higher global success ratio with lower convergence times – consistent with our *sufficient connectivity* hypothesis. Third, networks with decentralized architectures with high connectivity between nodes were observed to be highly robust to random deactivation of agents and retain their information transfer rates for various levels of severity of the hostile environment. We showed that scale-free and centralized networks are therefore less robust compared to multi-ring and von Neumann graphs.

Our study on the performance of the best topologies in the triangular spectrum indicated that two networks with 8-hub and 9 multi-ring structures exhibit the best performance among all the other architectures. Further analysis of the performance of these topologies and the four standard topologies (including small-world graphs) on different objective functions revealed that small-world graphs consistently perform well in terms of a higher global success ratio and lower convergence time. Small-world graphs are in agreement with our observation of decentralized architectures with high connectivity and an average path length of ~ 3 performing well in the PSO framework. An analysis of a metric quantifying small-world-ness revealed that such networks inherently achieve a trade-off between efficient flow of information between agents and tolerance to hostile conditions. We therefore conclude that small-world like networks are well-suited for optimizing objective

functions when the function landscape is unknown and the environment could be hostile towards the agents.

Although the findings of this work are based on performance of the PSO algorithm, the results are fairly generalizable and should extend to algorithms that rely on efficient communication between agents while searching for an optimal solution. Future work in this area involves extending the current ideas to incorporate adaptive death rates as a function of time and space, and swarm reorganization to minimize the impact of agent deactivation on network topologies.

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.

Acknowledgements

The authors gratefully acknowledge the contributions of the following former students in our group for their earlier work on this topic - Arun V. Giridhar, Balachandra B. Krishnamurthy, Chunhua Zhao, Priyan R. Patkar, and Santhoji Katare. We also wish to thank the following research interns at the Complex Resilient Intelligent Systems (CRIS) laboratory at Columbia University - Xijao Li, Liyi Zhang, and Jia Wan. This work was supported in part by the Center for the Management of Systemic Risk (CMSR) at Columbia University.

References

[1] S. Patnaik, X.-S. Yang, K. Nakamatsu, Nature-Inspired Computing and Optimization, 10, Springer, 2017.
[2] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization, arXiv:1307.4186(2013).
[3] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the ICNN'95-International Conference on Neural Networks, 4, IEEE, 1995, pp. 1942–1948.
[4] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.
[5] X.-S. Yang, et al., Firefly algorithm, in: Nature-Inspired Metaheuristic Algorithms, 20, 2008, pp. 79–90.
[6] I. Fister Jr, M. Perc, S.M. Kamal, I. Fister, A review of chaos-based firefly algorithms: perspectives and research challenges, Appl. Math. Comput. 252 (2015) 155–165.
[7] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009, pp. 210–214.
[8] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010, pp. 65–74.
[9] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: squirrel search algorithm, Swarm Evol. Comput. 44 (2019) 148–175.
[10] M. Mavrovouniotis, C. Li, S. Yang, A survey of swarm intelligence for dynamic optimization: algorithms and applications, Swarm Evol. Comput. 33 (2017) 1–17.
[11] S. Sundar, A. Singh, A swarm intelligence approach to the early/tardy scheduling problem, Swarm Evol. Comput. 4 (2012) 25–32.
[12] O. Ertenlice, C.B. Kalayci, A survey of swarm intelligence for portfolio optimization: algorithms and applications, Swarm Evol. Comput. 39 (2018) 36–52.
[13] S. Nebti, A. Boukerram, Swarm intelligence inspired classifiers for facial recognition, Swarm Evol. Comput. 32 (2017) 150–166.
[14] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[15] T. Bäck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, CRC Press, 1997.
[16] M. Schwaab, E.C. Biscaia, J.L. Monteiro, J.C. Pinto, Nonlinear parameter estimation through particle swarm optimization, Chem. Eng. Sci. 63 (6) (2008) 1542–1552, doi:10.1016/j.ces.2007.11.024.
[17] Y. Zhou, C. Zhao, X. Liu, An iteratively adaptive particle swarm optimization approach for solving chemical dynamic optimization problems, Huagong Xuebao/CIESC J. 65 (4) (2014) 1296–1302, doi:10.3969/j.issn.0438-1157.2014.04.020.
[18] C.O. Ourique, E.C. Biscaia, J.C. Pinto, The use of particle swarm optimization for dynamical analysis in chemical processes, Comput. Chem. Eng. 26 (12) (2002) 1783–1793, doi:10.1016/S0098-1354(02)00153-9.
[19] Y. Wang, Y. Ni, N. Li, S. Lu, S. Zhang, Z. Feng, J. Wang, A method based on improved ant lion optimization and support vector regression for remaining useful life estimation of lithium-ion batteries, Energy Sci. Eng. (2019), doi:10.1002/ese3.460.
[20] S. Alam, G. Dobbie, Y.S. Koh, P. Riddle, S.U. Rehman, Research on particle swarm optimization based clustering: a systematic review of literature and techniques, Swarm Evol. Comput. 17 (2014) 1–13.
[21] J.R. Zhang, J. Zhang, T.M. Lok, M.R. Lyu, A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training, Applied Mathematics and Computation 185 (2) (2007) 1026–1037, doi:10.1016/j.amc.2006.07.025.
[22] E. Camci, D.R. Kripalani, L. Ma, E. Kayacan, M.A. Khanesar, An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm, Swarm Evol. Comput. 41 (2018) 1–8.
[23] A. El-Zonkoly, Optimal placement of multi-distributed generation units including different load models using particle swarm optimization, Swarm Evol. Comput. 1 (1) (2011) 50–59.
[24] N. Jin, Y. Rahmat-Samii, Analysis and particle swarm optimization of correlator antenna arrays for radio astronomy applications, IEEE Trans. Antennas Propag. 56 (5) (2008) 1269–1279.
[25] T. Navalertporn, N.V. Afzulpurkar, Optimization of tile manufacturing process using particle swarm optimization, Swarm Evol. Comput. 1 (2) (2011) 97–109.
[26] J. Zhang, C. Zhang, T. Chu, M. Perc, Resolution of the stochastic strategy spatial Prisoner's dilemma by means of particle swarm optimization, PloS One 6 (7) (2011) e21787.
[27] M. Pluhacek, R. Senkerik, A. Viktorin, T. Kadavy, I. Zelinka, A review of real-world applications of particle swarm optimization algorithm, Lect. Notes Electr. Eng. 465 (2018) 115–122, doi:10.1007/978-3-319-69814-4_11.
[28] M.R. Bonyadi, A theoretical guideline for designing an effective adaptive particle swarm, IEEE Trans. Evol. Comput. (2019).
[29] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the Congress on Evolutionary Computation, CEC 1999, 3, IEEE Computer Society, 1999, pp. 1945–1950, doi:10.1109/CEC.1999.785511.
[30] Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, in: V.W. Porto, N. Saravanan, D. Waagen, A.E. Eiben (Eds.), Evolutionary Programming VII, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 591–600.
[31] K.M. Ang, W.H. Lim, N.A.M. Isa, S.S. Tiang, C.H. Wong, A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems, Expert Syst. Appl. 140 (2020) 112882.
[32] M.Z. bin Mohd Zain, J. Kanesan, J.H. Chuah, S. Dhanapal, G. Kendall, A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization, Appl. Soft Comput. 70 (2018) 680–700.
[33] D. Fatemeh, C. Loo, G. Kanagaraj, Shuffled complex evolution based quantum particle swarm optimization algorithm for mechanical design optimization problems, J. Modern Manuf. Syst. Technol. 2 (2019) 23–32.
[34] A. Shukla, V.A. Lalit, V. Venkatasubramanian, Optimizing efficiency-robustness trade-offs in supply chain design under uncertainty due to disruptions, Int. J. Phys. Distrib. Logist. Manag. (2011).
[35] A. Shukla, V. Agarwal, L.M. Hailemariam, V. Venkatasubramanian, Supply chain optimization for efficiency and robustness objectives, in: Proceedings of the AIChE Annual Meeting, 2007.
[36] A. Shukla, V. Agarwal, V. Venkatasubramanian, A functional approach to optimization of supply chain networks with both efficiency and robustness objectives, in: Proceedings of the AIChE Annual Meeting, 2008.
[37] Y. Meepetchdee, N. Shah, Logistical network design with robustness and complexity considerations, Int. J. Phys. Distribut. Logist. Manag. (2007).
[38] V. Venkatasubramanian, S. Katare, P.R. Patkar, F.-p. Mu, Spontaneous emergence of complex optimal networks through evolutionary adaptation, Comput. Chem. Eng. 28 (9) (2004) 1789–1798.
[39] W. Ellens, R.E. Kooij, Graph measures and network robustness, arXiv:1311.5064(2013).
[40] W. Du, M. Zhang, W. Ying, M. Perc, K. Tang, X. Cao, D. Wu, The networked evolutionary algorithm: a network science perspective, Appl. Math. Comput. 338 (2018) 33–43.
[41] A. Giridhar, B.B. Krishnamurthy, P. Patkar, C. Zhao, V. Venkatasubramanian, Effect of network topologies on particle swarm optimization, in: Proceedings of the AIChE Annual Meeting, 2004.
[42] B.B. Krishnamurthy, A. Giridhar, P. Patkar, C. Zhao, V. Venkatasubramanian, Robustness of networks in particle swarm optimization, in: Proceedings of the AIChE Annual Meeting, 2005.
[43] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), 2, 2002, pp. 1671–1676vol.2, doi:10.1109/CEC.2002.1004493.
[44] R. Mendes, J. Kennedy, J. Neves, Watch thy neighbor or how the swarm can learn from its environment, in: Proceedings of the IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), 2003, pp. 88–94, doi:10.1109/SIS.2003.1202252.
[45] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Proceedings of the Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 3, 1999, pp. 1931–1938Vol. 3, doi:10.1109/CEC.1999.785509.
[46] S. Surjanovic, D. Bingham, Virtual library of simulation experiments: test functions and datasets, 2013, (Retrieved March 13, 2020, from http://www.sfu.ca/~ssurjano).
[47] M. Molga, C. Smutnicki, Test functions for optimization needs 101 (2005). http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf, https://scholar.google.com/scholar_lookup?title=Test%20functions%20for%20optimization%20needs&publication_year=2005&author=M.%20Molga&author=C.%20Smutnicki.
[48] W. Jun, M. Barahona, T. Yue-Jin, D. Hong-Zhong, Natural connectivity of complex networks, Chin. Phys. Lett. 27 (7) (2010) 078902.
[49] Q.K. Telesford, K.E. Joyce, S. Hayasaka, J.H. Burdette, P.J. Laurienti, The ubiquity of small-world networks, Brain Connect. 1 (5) (2011) 367–375.